

D4.2: Final report on zero-touch data-driven network and service orchestration in the 3D ETHER architecture

Revision: v.1.0

Work package	WP 4
Task	Tasks 4.1, 4.2, and 4.3
Due date	30/08/2025
Submission date	02/09/2025
Deliverable lead	I2CAT
Version	1.0
Editors	Hossein Rouzegar (i2CAT), Margherita Facca (MAR)
Reviewers	Agapi Mesodiakaki (AUTH), Konstantinos Ntontin (UL), Luís Santos (UBW)
Abstract	This deliverable includes the final works performed in WP4 on developing zero-touch data-driven network and service orchestration in the 3D ETHER architecture. This document provides a comprehensive overview of three key developments in WP4: 1) Zero-touch management, orchestration and monitoring of the virtualised network and cloud continuum resources, 2) Unified RAN and transport management leveraging distributed network control across the ETHER layers and 3) AI-enabled E2E network optimisation.
Keywords	6G, NTN, MANO, NFV, AI, Unified RAN, SDN, Network Optimisation

Document Revision History

Version	Date	Description of change
V0.1	30/07/2025	1st version ready for internal review
V0.2	30/08/2025	2nd version after internal review
V1.0	02/09/2025	Final version for submission

Author List

Organisations	Author
UL	Konstantinos Ntontin, Ilora Maity, Mario Camana, Duc-Dung Tran
AUTH	Charalampos Bratsoudis, Agapi Mesodiakaki, George Kalfas, Marios Gatzianas, Amalia Miliou, Dimitrios Palianopoulos
CA	Hamada Alshaer and Fadhil Firyaguna
AVA	Eleftheria Hadjioannou
UBW	Luís Santos, João Viegas
I2CAT	Hossein Rouzegar, Reza Mosahebfard, Joan A. Ruiz-de-Azua, José Ávila
NBC	Godfrey Kibalya, Angelos Antonopoulos
OPL	Robert Kołakowski, Lechosław Tomaszewski
NETAI	Alexis Duque

DISCLAIMER



ETHER (*sElf-evolving terrestrial/non-Terrestrial Hybrid nEtwoRks*) project has received funding from the [Smart Networks and Services Joint Undertaking \(SNS JU\)](#) under the European Union's [Horizon Europe research and innovation programme](#) under Grant Agreement No 101096526. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

This work has received funding from the [Swiss State Secretariat for Education, Research and Innovation \(SERI\)](#).

COPYRIGHT NOTICE

© 2023 - 2025 ETHER Consortium

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	Report (R)	
Dissemination Level		
PU	Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)	✓
SEN	Sensitive, limited under the conditions of the Grant Agreement	
Classified R-UE/ EU-R	EU RESTRICTED under the Commission Decision <u>No2015/ 444</u>	
Classified C-UE/ EU-C	EU CONFIDENTIAL under the Commission Decision <u>No2015/ 444</u>	
Classified S-UE/ EU-S	EU SECRET under the Commission Decision <u>No2015/ 444</u>	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

EXECUTIVE SUMMARY

This document serves as the WP4 final deliverable (D4.2), outlining the completed activities in developing the zero-touch, data-driven network and service orchestration within the 3D ETHER architecture. This report builds upon the previous deliverable (D4.1), which was introduced as the initial report. The framework aims to facilitate efficient service management and to establish the essential virtual networks for interconnectivity, driven by AI-optimised resource allocation. The developments are aligned with all three ETHER UCs: UC1 on flexible payload-enabled service provisioning to semantics-aware and delay-tolerant IoT; UC2 as unified RAN for direct handheld device access at the Ka band and UC3 for air-space safety critical operations.

This report offers a detailed description of three significant developments in WP4:

1. Zero-touch management, orchestration, and monitoring of virtualised network and cloud continuum resources: This development is described in **Chapter 2** of this document encompassing two main activities. **The 1st activity** corresponds to the AI-driven zero-touch closed-loop orchestration to effectively harness the cloud and edge computing capabilities provided by the ETHER infrastructure, which pertains to UC3. **The 2nd activity** is focused on the localisation and mobility management towards continuous services addressing the challenges posed by the dynamic and resource-constrained nature of the infrastructure. The activity provides strategies for orchestration, mobility management, and geo-localisation, which relate to UC1 and may also be relevant to the other two UCs. Both activities are a follow-up of the two activities of chapter 2 in deliverable D4.1.
2. Integrated RAN and transport management utilising distributed network control across the ETHER layers: This development is mapped to **Chapter 3** of this document that comprises **six key activities**. **The 1st activity** pertains to the deployment of a Software-Defined Networking (SDN)-based Wide Area Network Infrastructure Manager (WIM), which facilitates the integration of both terrestrial and non-terrestrial networks. This activity aims at enhancing real-time adaptability, minimising control plane overhead, and dynamically responding to changes in topology to maintain uninterrupted connectivity. The functionalities of the SDN-based WIM will be demonstrated through a standalone simulation focused on UC1, while also being applicable to the other two UCs. **The 2nd activity** is focused on the integration of Quality of Service (QoS)-aware, multi-domain 3D routing utilising Deep Reinforcement Learning. This methodology ensures End-to-End (E2E) differentiated and guaranteed services for users, while simultaneously enhancing network performance through effective resource allocation. The activity is specifically addressing UC3. Both the first and second activities are continuations of the A2-1 and A2-2 activities in D4.1, respectively. **The 3rd activity** corresponds to Reinforcement learning-based data compression and routing design for multi-satellite systems which concerns UC3. **The 4th activity** corresponds to Model-based 3D network management and SDN integration. The activity will focus on further work on 3D network handover, message routing, and SDN-enabled HAP and terrestrial networks integration, addressing UC3. **The 5th activity** is dedicated to the distributed SDN controller placement, addressing UC3 as well as the ground segment of the UC1. Finally, **the 6th activity** corresponds to a technical specification for an SDN controller that is focused on integrating NTN and TN across the 3D ETHER network, in the scope of UC3.
3. AI-driven E2E network optimisation: Chapter 4 of this document elaborates on this development, which includes four activities. **The 1st activity** involves the implementation of an AI-driven 3D network performance and topology forecasting system. This system employs a specialised machine learning pipeline to ensure scalable and precise predictions, even in the presence of Key Performance Indicator (KPI) noise (addressing

UC3). The **2nd activity** focuses on a real-time, energy-efficient resource allocation solution that effectively manages joint user association, traffic routing, and xNF placement by proposing both an optimal though complex solution and a heuristic for reduced complexity, also addressing UC2 and UC3. The **3rd activity** leverages reinforcement learning to offer an energy-efficient offline user association strategy addressing both UC2 and UC3. Lastly, the **4th activity** pertains to the design of AI-Driven Beamforming for Integrated Sensing and Communication in Terrestrial-Satellite Networks, addressing UC2.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	4
TABLE OF CONTENTS	6
LIST OF FIGURES.....	9
LIST OF TABLES	13
ABBREVIATIONS	14
1 INTRODUCTION	19
1.1 DOCUMENT STRUCTURE AND CORRESPONDENCE TO D4.1	19
1.2 RELATIONSHIP WITH OTHER ETHER DELIVERABLES	23
2 ZERO-TOUCH MANAGEMENT, ORCHESTRATION AND MONITORING OF THE VIRTUALISED ETHER NETWORK AND CLOUD CONTINUUM RESOURCES.....	25
2.1 INTRODUCTION.....	25
2.2 AI-DRIVEN ZERO-TOUCH CLOSED-LOOP ORCHESTRATION.....	26
2.2.1 <i>Background</i>	26
2.2.2 <i>Motivation and Contribution</i>	27
2.2.3 <i>General architecture of closed-loop framework</i>	27
2.2.4 <i>Northbound Interface enhancements</i>	28
2.2.5 <i>Observability Stack architecture</i>	31
2.3 GEO-LOCALISATION AND MOBILITY MANAGEMENT TOWARDS CONTINUOUS SERVICE	32
2.3.1 <i>Background</i>	32
2.3.2 <i>Motivation and Contribution</i>	33
2.3.3 <i>Implementation</i>	34
2.3.4 <i>Experimental Validation and Key Results:</i>	35
3 UNIFIED RAN AND TRANSPORT MANAGEMENT LEVERAGING DISTRIBUTED NETWORK CONTROL ACROSS THE ETHER LAYERS	38
3.1 INTRODUCTION.....	38
3.2 SDN-BASED WIM FOR INTEGRATED TN-NTN.....	40
3.2.1 <i>Background</i>	40
3.2.2 <i>Motivation and Contribution</i>	40
3.2.3 <i>System model for the WIM developments</i>	41
3.2.4 <i>Final Implementation</i>	42
3.2.4.1 <i>Automation Scripts</i>	44
3.2.4.2 <i>WIM Control Plane</i>	45
3.2.5 <i>Performance Evaluation</i>	45
3.2.5.1 <i>Evaluation Scenario</i>	46
3.2.5.2 <i>Results and Discussion</i>	47
3.3 QOS-AWARE MULTI-DOMAIN 3D ROUTING (SDN-APP).....	50
3.3.1 <i>Background</i>	50
3.3.2 <i>Concept description</i>	51
3.3.3 <i>E2E Routing and Path Allocation Approach</i>	52
3.3.4 <i>3DQR Algorithm</i>	53
3.3.4.1 <i>System Model</i>	53
3.3.4.2 <i>DRL problem setup</i>	55
3.3.4.3 <i>E2E routing</i>	56
3.3.5 <i>Evaluation approach & results</i>	58
3.3.5.1 <i>Performance</i>	59

3.3.5.2	Transfer capabilities	62
3.4	REINFORCEMENT LEARNING- BASED DATA COMPRESSION AND ROUTING DESIGN FOR MULTI-SATELLITE SYSTEMS	63
3.4.1	<i>Background</i>	63
3.4.2	<i>Motivation and Contribution</i>	63
3.4.3	<i>System Model</i>	63
3.4.4	<i>Proposed Solution</i>	67
3.4.5	<i>Simulation Results</i>	69
3.5	MODEL-BASED 3D NETWORK MANAGEMENT AND SDN INTEGRATION	72
3.5.1	<i>Background</i>	73
3.5.2	<i>Motivation and Contribution</i>	73
3.5.3	<i>System Model</i>	73
3.5.4	<i>Problem Description</i>	74
3.5.5	<i>Proposed Solution</i>	75
3.5.5.1	Aircraft Message Routing Policies	75
3.5.5.2	Cost-Driven Handover and Base Station Association	76
3.5.6	<i>Simulation Results</i>	80
3.5.6.1	Simulation Setup	81
3.5.6.2	Performance Evaluation	82
3.6	DISTRIBUTED SDN CONTROLLER PLACEMENT	86
3.6.1	<i>Background</i>	86
3.6.2	<i>Motivation and Contribution</i>	87
3.6.3	<i>System Model</i>	88
3.6.3.1	Link Budget Model	89
3.6.3.2	Traffic Model	90
3.6.3.3	Cost Model	90
3.6.4	<i>Problem description and formulation</i>	92
3.6.5	<i>The proposed scheme</i>	93
3.6.5.1	GA for Domain and Master Controller Placement	93
3.6.5.2	Incremental Update of Domains and Controller Placement	94
3.6.6	<i>Performance evaluation</i>	94
3.6.6.1	Simulation Settings	94
3.6.6.2	Benchmark Schemes	95
3.6.6.3	Performance Metrics	95
3.6.6.4	Result and Discussion	95
3.7	SOFTWARE DEFINED NETWORK CONTROLLER FOR THE 3D ETHER ARCHITECTURE	97
3.7.1	<i>Background</i>	97
3.7.2	<i>Motivation and Contribution</i>	98
3.7.3	<i>System Model and Components</i>	99
3.7.4	<i>Communication Flows</i>	100
3.7.5	<i>Implementation details and testing</i>	102
4	AI-ENABLED E2E NETWORK OPTIMISATION	104
4.1	INTRODUCTION	104
4.2	AI-ENABLED 3D NETWORK PERFORMANCE AND TOPOLOGY FORECAST	105
4.2.1	<i>Background</i>	105
4.2.2	<i>Motivation and Contribution</i>	106
4.2.3	<i>Methodology and approach</i>	106
4.2.4	<i>KPM forecasting on the cellular backhaul (CBH) sites</i>	107
4.2.5	<i>KPM Forecasting of flight link quality</i>	107

4.2.5.1	Dataset Description.....	107
4.2.5.2	Forecasting models.....	108
4.2.5.3	Prediction Task	111
4.2.5.4	Model Results and Predictions	113
4.2.5.5	Inference Pass Speed.....	117
4.2.5.6	Discussion.....	118
4.3	REAL-TIME ENERGY-EFFICIENT RESOURCE ALLOCATION AND ROUTING.....	118
4.3.1	<i>Background</i>	118
4.3.2	<i>Motivation and Contribution</i>	119
4.3.3	<i>Proposed algorithm</i>	120
4.3.4	<i>Performance evaluation</i>	122
4.3.5	<i>Results</i>	124
4.4	AI-BASED OFFLINE USER ASSOCIATION FOR THE 3D SUSTAINABLE ETHER NETWORK	127
4.4.1	<i>Background</i>	127
4.4.2	<i>Motivation and Contribution</i>	128
4.4.3	<i>System Model and Problem Formulation</i>	128
4.4.4	<i>Proposed RL-algorithm</i>	130
4.4.5	<i>Performance Evaluation</i>	132
4.4.6	<i>Simulation Results</i>	137
4.5	AI-DRIVEN BEAMFORMING FOR INTEGRATED SENSING AND COMMUNICATION IN TERRESTRIAL-SATELLITE NETWORKS	139
4.5.1	<i>Background</i>	139
4.5.2	<i>Motivation and Contribution</i>	140
4.5.3	<i>System Model</i>	141
4.5.4	<i>Problem Formulation</i>	143
4.5.5	<i>Proposed Solution</i>	143
4.5.6	<i>Results</i>	145
5	CONCLUSION	150
6	REFERENCES	154

LIST OF FIGURES

FIGURE 1-1: WP4 POSITIONING WITH ETHER 3D NETWORK.	19
FIGURE 2-1: CORRESPONDENCE OF THE A1-1 AND A1-2 ACTIVITIES WITH THE MANO ARCHITECTURE.	26
FIGURE 2-2: ZERO-TOUCH CLOSED-LOOP FRAMEWORK.	28
FIGURE 2-3: INFRASTRUCTURE RELATED ENDPOINTS FOR THE NORTHBOUND INTERFACE.	29
FIGURE 2-4: OBSERVABILITY STACK ARCHITECTURE.	32
FIGURE 2-5: GIS-ENHANCED MANO SYSTEM FOR NETWORK SERVICES DEPLOYMENT.	35
FIGURE 2-6: TESTBED DEPLOYMENT OF GIS-ENHANCED MANO FRAMEWORK.	35
FIGURE 2-7: CPU & RAM USAGE OF A SATELLITE DURING A CONTACT OVER A TARGET AREA.	36
FIGURE 2-8: TRANSFER RATE AND LATENCY BETWEEN TWO USERS INSIDE THE TARGET AREA.	36
FIGURE 2-9: FRAMEWORK ANALYSIS OSM WITH SMM&GIS VS OSM WITHOUT SMM&GIS.	37
FIGURE 3-1. CORRESPONDENCE OF THE A2-1, A2-2, A2-3, A2-4, A2-5 AND A2-6 ACTIVITIES WITH THE MANO ARCHITECTURE.	39
FIGURE 3-2: HIGH-LEVEL SYSTEM ARCHITECTURE FOR SDN-BASED WIM DEVELOPMENTS.	42
FIGURE 3-3: IMPLEMENTED TESTBED FOR TESTING THE FUNCTIONALITY OF THE DEVELOPED SDN-BASED WIM.	43
FIGURE 3-4: THE OUTPUT OF THE TESTBED CREATION SCRIPT, FOLLOWED BY THE LIST OF CREATED DOCKER CONTAINERS.	44
FIGURE 3-5: THE OUTPUT OF THE TESTBED DESTRUCTION SCRIPT.	45
FIGURE 3-6: RTT DURING A CONTINUOUS HPING3 EXPERIMENT FOR FOUR HOURS, SHOWING RAW ICMP PROBE MEASUREMENTS, A 120-POINT ROLLING MEDIAN, AND THE MEAN RTT.	47
FIGURE 3-7: SERVER-SIDE RAW AND MEAN VALUES FOR UDP THROUGHPUT AS WELL AS THE PACKET LOSS PERCENTAGES DURING A CONTINUOUS 10 MBPS IPERF3 EXPERIMENT SESSION.	48
FIGURE 3-8: DISTRIBUTION OF SERVER-SIDE UDP THROUGHPUT OVER THE 4-HOUR EXPERIMENT, ILLUSTRATING THE MEDIAN, IQR, AND MEAN.	49
FIGURE 3-9: ONE-WAY UDP JITTER MEASURED AT THE SERVER OVER THE 4-HOUR EXPERIMENT, SHOWING RAW JITTER VALUES, A 120-POINT ROLLING MEDIAN, AND THE MEAN VALUE FOR JITTER.	49
FIGURE 3-10: HIGH-LEVEL VIEW OF THE 3DQR CONCEPT AND COMPONENTS.	51
FIGURE 3-11: DATA FLOW WITHIN M-QR/DQR.	52
FIGURE 3-12: E2EP ALLOCATION PROCESS IN 3DQR.	53
FIGURE 3-13: DQR/M-QR AND DGA INTERNALS (CURRENT, TARGET Q-NETWORKS AND LOSS FUNCTION).	57
FIGURE 3-14: EPISODIC REWARD OBTAINED BY 3DQR MODEL IN LOW-TRAFFIC ENVIRONMENT: FOR DOMAIN AGENTS FOR NETWORK TOPOLOGY T1 (UP) AND T2 (DOWN).	60
FIGURE 3-15: FLOW REJECTION RATE PER SCENARIO.	61
FIGURE 3-16: FLOW REJECTION RATE IMPROVEMENT PER QOS CLASS COMPARED TO BASELINE H-SP ROUTING.	61

FIGURE 3-17: STANDARD DEVIATION OF UTILISATION OF LINKS (MEAN MARKED WITH RED DOT).....	62
FIGURE 3-18: PERFORMANCE COMPARISON OF H-SP-ROUTING AND 3DQR MODEL IN TERMS OF FLOW REJECTION RATE.....	62
FIGURE 3-19: ILLUSTRATION OF A MULTI-ORBIT LEO SATELLITE NETWORK ARCHITECTURE.....	64
FIGURE 3-20: ELEVATION ANGLE DEFINITION.	64
FIGURE 3-21: CONVERGENCE ANALYSIS.	70
FIGURE 3-22: DELAY ANALYSIS.....	70
FIGURE 3-23: EFFECT OF CR LEVELS.	71
FIGURE 3-24: EFFECT OF DATA SIZE.....	71
FIGURE 3-25: SDN-ENABLED INTEGRATED 3D NETWORK ARCHITECTURE.	74
FIGURE 3-26: PREDICTED SERVICE TIMES OF VISIBLE GNBS TO THE AIRCRAFT.	77
FIGURE 3-27: DIRECTED GRAPH GENERATED FROM PREDICTED SERVICE TIME, WITH THE MAIN HANDOVER DECISION (HIGHLIGHTED IN GREEN) IS CALCULATED AS THE SHORTEST PATH OF THE GRAPH.	77
FIGURE 3-28: DIRECTED GRAPH GENERATED BY REMOVING THE MAIN HANDOVER DECISION. THE ALTERNATIVE HANDOVER DECISION (HIGHLIGHTED IN YELLOW) IS CALCULATED AS THE NEW SHORTEST PATH OF THE NEW GRAPH.	79
FIGURE 3-29: REGIONAL FLIGHT BAW827 FROM DUBLIN TO LONDON (GREEN LINE). THE PINK CIRCLES ILLUSTRATE THE HAPS COVERAGE, THE BLUE CIRCLES ILLUSTRATE THE ATG COVERAGE, AND THE YELLOW-BLUE DOTS ARE THE LEO SATELLITES.	81
FIGURE 3-30: AVERAGE NETWORK LOAD DISTRIBUTION AMONG THE DIFFERENT SUBNETWORKS FOR EACH HANDOVER METHOD.....	83
FIGURE 3-31: COST OF NETWORK USAGE OVER THE FLIGHT FOR EACH HANDOVER METHOD.....	83
FIGURE 3-32: 99TH PERCENTILE OF DELAY OVER THE FLIGHT FOR EACH HANDOVER METHOD.....	85
FIGURE 3-33: [CLOSE UP BETWEEN 0 AND 0.2 SECONDS] 99TH PERCENTILE OF DELAY OVER THE FLIGHT FOR EACH HANDOVER METHOD.....	85
FIGURE 3-34: AVERAGE DATA RATE OVER THE FLIGHT FOR EACH HANDOVER METHOD.	86
FIGURE 3-35: COMOSAT - SYSTEM ARCHITECTURE.....	88
FIGURE 3-36: COMOSAT - NUMBER OF CONNECTIONS TO DOMAIN NODES.	88
FIGURE 3-37: COMOSAT - CONTROLLER COUNT.....	95
FIGURE 3-38: COMOSAT - FLOW SETUP DELAY.....	96
FIGURE 3-39: COMOSAT - AVERAGE NUMBER OF REMAPS WITH VARYING LEO COUNT.	97
FIGURE 3-40: COMOSAT - AVERAGE NUMBER OF REMAPS ACROSS TIME SLOTS.	97
FIGURE 3-41: SDN CONTROLLER MAPPING TO THE 3D ETHER NETWORK.	100
FIGURE 3-42: DEVICE ONBOARDING FLOW DIAGRAM.	101
FIGURE 3-43: COMMUNICATION HANDOVER FLOW DIAGRAM.....	102
FIGURE 3-44: ONOS IMPLEMENTATION SETUP.	103
FIGURE 4-1: CORRESPONDENCE OF THE ACTIVITIES OF THE A3.1, A3.2, A3.3 AND A3.4 DEVELOPMENTS WITH THE MANO ARCHITECTURE.....	105
FIGURE 4-2 GEOGRAPHICAL DISTRIBUTION OF THE FLIGHT PATHS.....	108
FIGURE 4-3: SEQ2SEQ LSTM ARCHITECTURE.	109
FIGURE 4-4: AUTO-CORRELATION VS. SELF-ATTENTION.	110
FIGURE 4-5: ARCHITECTURE DIAGRAM.....	110

FIGURE 4-6: ARCHITECTURE OF THE N-BEATS MODEL.....	111
FIGURE 4-7: SNR DISTRIBUTION ACROSS STATION TYPES.....	112
FIGURE 4-8: DELAY DISTRIBUTION ACROSS STATION TYPES.....	112
FIGURE 4-9: EFFECT OF LOG NORMALISATION ON PREDICTION ERRORS FOR THE SNR KPM.....	113
FIGURE 4-10: PREDICTION ERRORS FOR THE SNR KPM ACROSS STATION TYPES AND MODELS.....	113
FIGURE 4-11: NBEATS LEO 119 PREDICTIONS.....	114
FIGURE 4-12: TSMIXER LEO 119 PREDICTIONS.....	114
FIGURE 4-13: DELAY PREDICTION ERRORS VS. MODEL ACROSS STATION TYPES.....	115
FIGURE 4-14: TSMIXER LEO 10 PREDICTIONS.....	115
FIGURE 4-15: NBEATS LEO 10 PREDICTIONS.....	116
FIGURE 4-16: LEO 10 TRAIN VS. TEST SET DYNAMICS (ALL FLIGHTS PLOTTED WITH START TIMESTEP ALIGNED).....	116
FIGURE 4-17: TSMIXER HAP 1 PREDICTIONS.....	117
FIGURE 4-18: NBEATS HAP 1 PREDICTIONS.....	117
FIGURE 4-19: OPERATION FLOWCHART OF PETA.....	121
FIGURE 4-20: SIMULATION SETUP EXAMPLE DEPICTING BOTH THE TERRESTRIAL AND NON-TERRESTRIAL BASE STATION POSITIONS.....	122
FIGURE 4-21: ENERGY EFFICIENCY (BITS/JOULE) OF ALL ALGORITHMS FOR DIFFERENT TRAFFIC LOAD CONDITIONS AND SR ARRIVAL RATES ($1/\Lambda$).....	124
FIGURE 4-22: EXECUTION TIME IN SEC OF ALL ALGORITHMS FOR DIFFERENT USER TRAFFIC LOAD CONDITIONS AND SR ARRIVAL RATES ($1/\Lambda$).....	125
FIGURE 4-23: ACCEPTANCE RATIO OF ALL ALGORITHMS FOR DIFFERENT USER TRAFFIC LOAD CONDITIONS AND SR ARRIVAL RATES ($1/\Lambda$).....	126
FIGURE 4-24: POWER BREAK-DOWN IN W OF ALL ALGORITHMS FOR LOW AND HIGH TRAFFIC ($N=10$ AND $N=40$, RESPECTIVELY) AS WELL AS HIGH ARRIVAL RATE ($1/\Lambda=1$).....	126
FIGURE 4-25: POWER BREAK-DOWN IN W OF ALL ALGORITHMS FOR LOW AND HIGH TRAFFIC ($N=10$ AND $N=40$, RESPECTIVELY) AS WELL AS LOW ARRIVAL RATE ($1/\Lambda=10$).....	127
FIGURE 4-26: SYSTEM MODEL OF THE INTEGRATED TERRESTRIAL, AERIAL, AND SATELLITE NETWORK.....	129
FIGURE 4-27: ACTOR NETWORK LAYER VISUALIZATION.....	131
FIGURE 4-28: FIRST DNN TRAINING OVER 4000 SAME ITERATION SCENARIOS.....	134
FIGURE 4-29: SECOND DNN TRAINING OVER 10 DIFFERENT SCENARIOS. EACH SCENARIO HAS DIFFERENT UE SERVICES AND TOPOLOGIES. THE SCENARIOS ARE MULTIPLIED BY 1000 AND SHUFFLED TO AVOID OVERFITTING.....	135
FIGURE 4-30: POWER CONSUMPTION REDUCTION ACROSS 2ND TRAINING'S EPISODES.....	136
FIGURE 4-31: THIRD DNN TRAINING OVER 9900 DIFFERENT SCENARIOS. EACH SCENARIO HAS DIFFERENT UE SERVICES AND TOPOLOGIES. THE SCENARIOS ARE MULTIPLIED BY 3 AND SHUFFLED.....	137
FIGURE 4-32: SIMULATION OVER 100 DIFFERENT SCENARIOS. THE 40 UES HAVE DIFFERENT TOPOLOGIES AND SERVICES, WHILE THE CELL NETWORK STAYS THE SAME. THE AGENT ISN'T TRAINED ON THESE SPECIFIC 100 SCENARIOS.....	138
FIGURE 4-33: POWER CONSUMPTION COMPARISON AMONG RL-PROPOSED, RSRP AND TERA ALGORITHMS.....	139
FIGURE 4-34: ISAC-ENABLED INTEGRATED SATELLITE AND TERRESTRIAL DISTRIBUTED MIMO SYSTEM.....	141

FIGURE 4-35: ILLUSTRATION OF THE RSMA FRAMEWORK.	141
FIGURE 4-36: PROPOSED CNN-BASED MODEL FOR POWER ALLOCATION.	144
FIGURE 4-37: CONVERGENCE BEHAVIOUR OF THE CNN-BASED MODEL.	147
FIGURE 4-38: SUM-RATE VERSUS THE TRANSMISSION POWER.	147
FIGURE 4-39: CDF OF THE SUM-RATE.	148
FIGURE 4-40: NORMALIZED TRANSMIT BEAMPATTERN FOR A TARGET WITH $\theta^* = 0^\circ$.	148
FIGURE 4-41: CDF OF THE COMPUTATIONAL TIME.	149

LIST OF TABLES

TABLE 3-1: EVALUATION SCENARIOS.....	59
TABLE 3-2: CONSIDERED SERVICES (BASED ON [30]).	59
TABLE 3-3: SIMULATION PARAMETERS.....	69
TABLE 3-4: EFFECT OF QUEUE LENGTH (QMAX).	72
TABLE 3-5: TWO METHODS FOR MESSAGE ROUTING AND RESOURCE OPTIMISATION PROCESS.....	75
TABLE 3-6: COST OF OPERATION FOR DIFFERENT GNBS.....	78
TABLE 3-7: SYSTEM PARAMETERS	82
TABLE 3-8: COMOSAT - SIMULATION PARAMETERS.....	94
TABLE 4-1: INFERENCE TIMES FOR A FULL INFERENCE PASS.....	118
TABLE 4-2 SIMULATION PARAMETERS [81], [78], [82], [83].....	123
TABLE 4-3 PARAMETER VALUES USED IN ALL TRAINING PHASES.....	133
TABLE 4-4 SIMULATION PARAMETERS FOR DIFFERENT ACCESS NETWORK PARTS...	133

ABBREVIATIONS

3D	3-Dimensional
3DQR	3D QoS Routing
ACARS	Aircraft Communications, Addressing and Reporting System
ADS-B	Automatic Dependent Surveillance-Broadcast
AoI	Age of Information
AI	Artificial Intelligence
AlaaS	AI as a Service
AN	Access Network
API	Application Programming Interface
ATG	Air-To-Ground
BER	Bit Error Rate
BH	BackHaul
BN	Border Node
BS	Base Station
CBH	Cellular BackHaul
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
CoMOSat	Controller Placement in SDN-enabled Multi-Orbit Satellite Networks
CP	Control Plane
CPDLC	Controller-Pilot Data Link Communications
CPP	Controller Placement Problem
CPU	Central Processing Unit
DC	Domain Controller or Data Compression (depending on the context)
DD	Data Decompression
DI	Domain Infrastructure
DMMF	Domain Mobility Management Function
DMOC	Domain M&O Component
DP	Data Plane
DQR	Domain QoS Router
DQRN	Dueling double deep Q network
DRL	Deep Reinforcement Learning
DTMC	Discrete-Time Markov Chain
E2E	End-to-End
E2EAO	E2E Application Orchestrator
E2ENO	E2E Network Orchestrator

EE	Energy Efficiency
eMBB	enhanced Mobile Broadband
EMOC	E2E M&O Component
ETSI	European Telecommunication Standards Institute
FCAPS	Fault, Configuration, Accounting, Performance, Security
FCFS	First Come First Served
FIFO	First In First Out
FL	Feeder Link
FP	Fixed Placement
FSPL	Free Space Path Loss
GA	Genetic Algorithm
GBH	Graph-Based Handover
GBR	Guaranteed Bit Rate
GEO	Geostationary Earth Orbit
GFB	Guaranteed Flow Bit Rate
GIS	Geographical Information System
GMMF	Global Mobility Management Function
gNB	next-generation Node B
GNN	Graph Neural Network
GQR	Global QoS Router
GS	Ground Station
GW	Gateway
HAPS	High Altitude Platform System
HPA	Horizontal Pod Autoscaler
ISL	Inter-Satellite Link
ISAC	Integrated Sensing and Communication
ITNTN	Integrated Terrestrial and Non-Terrestrial Network
ITSN	Integrated Terrestrial and Satellite Network
JCR	Joint Data Compression and Routing
JPL	Joint Probabilistic Latency
KEDA	Kubernetes Event-Driven Autoscaler
KPI	Key Performance Indicator
KPM	Key Performance Metric
LAPS	Low Altitude Platform System
LC	Latency Cost
LCM	Life Cycle Management
LEO	Low Earth Orbit

LEONet	LEO satellite networks
LEOSat	LEO satellite
LoS	Line of Sight
LMMF	Local Mobility Management Function
LSTM	Long Short-Term Memory
M&O	Management and Orchestration (activities, processes)
MADDQN-JCR	Multi-Agent Double Deep Q-Network Algorithm JCR
MADRL	Multi-Agent Deep Reinforcement Learning
MAE	Mean Absolute Error
MANO	MANagement and Orchestration (architectural framework, software suite)
MC	Master Controller
MEAO	MEC Application Orchestrator
MEC	Multi-Access Edge Computing
MEO	Medium Earth Orbit
MEPM	MEC Platform Manager
MFBR	Maximum Flow Bit Rate
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLP	Multi-Layer Perceptron
MPLR	Maximum Packet Loss Rate
MPNN	Message Passing Neural Network
MRP	Message Routing Plan
MRT	Maximum Ratio Transmission
MSDNO	Master SDNO
MSE	Mean Squared Error
NBI	NorthBound Interface
NB-IoT	Narrow-Band IoT
NFV	Network Function Virtualisation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NGSO	Non-Geostationary Orbit
NI	Network Intelligence
NO	Network Orchestrator
NOS	Network Orchestration Strategy
NTN	Non-Terrestrial Network
OF	OpenFlow
ONF	Open Networking Forum

OPEX	Operational EXpenditures
O-RAN	Open RAN
OVS	Open Virtual Switch
QL	Q-learning
QoE	Quality of Experience
QoS	Quality of Service
PER	Packet Error Rate
PLR	Packet Loss Ratio
PRB	Physical Resource Block
RAN	Radio Access Network
RIC	Radio Intelligent Controller
RNN	Recurrent Neural Network
RPUD	Routing Plan Update Decision
RSMA	Rate-Splitting Multiple Access
RT	Real-Time
SBI	SouthBound Interface
SC	Small Cell
SCA	Successive Convex Approximation
SDN	Software-Defined Network
SDNC	SDN Controller
SDNO	SDN Orchestrator
SDR	Semidefinite Relaxation
SD-WAN	Software-Defined Wide Area Network
SF	Supplementary Function
SFC	Service Function Chain
SIC	Successive Interference Cancellation
SINR	Signal-to-Interference-plus-Noise Ratio
SMO	Service M&O
SNR	Signal-to-Noise Ratio
SotA	State-of-the-Art
SPDA	Static Placement with Dynamic Assignment
SR	Service Request
T-API	Transport API
TDM	Time Division Multiplexing
TE	Traffic Engineering
TLE	Two-Line Element
TN	Terrestrial Network

TS	Time Slot
UAV	Unmanned Aerial Vehicle
UC	Use Case
UE	User Equipment
UP	User Plane
URLLC	Ultra-Reliable Low Latency Communications
VIM	Virtualised Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFM	VNF Manager
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WIM	WAN Infrastructure Manager
WoC	MADRL-Based Routing Method without DC
WP	Work Package
xNF	any type of Network Function
ZF	Zero Forcing
ZSM	Zero-touch network and Service Management

1 INTRODUCTION

This chapter serves as an introduction of the ETHER WP4 final report, with a focus on zero-touch and data-driven orchestration of networks and services within the 3D ETHER architecture. It begins by describing the progress made in WP4, illustrating how these developments align with each ETHER Use Case (UC). This chapter subsequently presents the structure of the document, emphasising the various chapters, each focused on a particular aspect of the developments. Following this, it highlights the principal developments outlined in deliverable D4.1 [1] and notes the ongoing efforts documented in this report to bring WP4 to completion.

1.1 DOCUMENT STRUCTURE AND CORRESPONDENCE TO D4.1

The ETHER framework combines terrestrial and non-terrestrial network architectures, requiring an automated, data-centric orchestration of networks and services. This is made possible through the application of diverse innovative technologies aimed at meeting the expected UCs. ETHER presents three significant UCs including: Flexible payload-enabled service provisioning to semantics-aware and delay-tolerant IoT applications (UC1); Unified RAN for direct handheld device access at the Ka band (UC2); and Air-space safety critical operations (UC3).

For a better understanding, an overview of WP4 developments positioning within ETHER 3D network is depicted in Figure 1-1.

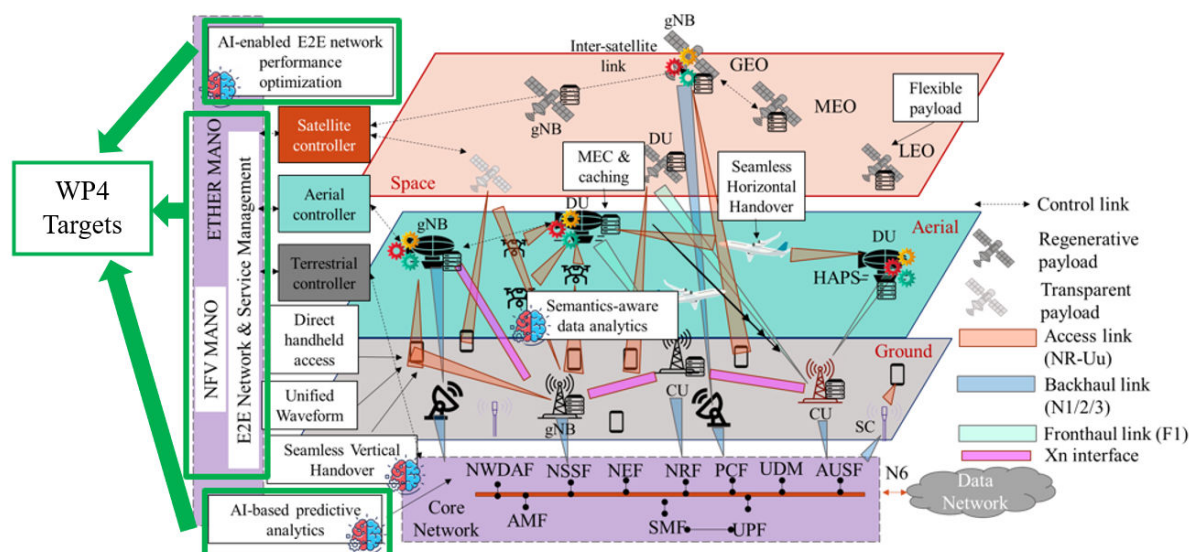


Figure 1-1: WP4 positioning with ETHER 3D network.

The initiatives conducted in WP4 are categorised into three main developments, each aligned with a separate chapter within this document. Furthermore, each development is broken down into several principal activities.

Chapter 2 is focused on the 1st development of WP4, which is dedicated to designing and implementing a framework that facilitates zero-touch management and orchestration of edge and cloud resources within the context of the integrated 3D architecture. This development comprises two main activities: AI-driven zero-touch closed-loop orchestration (A1-1) and Mobility and Geo-localisation management (A1-2).

- A1-1 is concerned with the concept of AI-driven zero-touch closed-loop orchestration which encompasses a fully automated pipeline for service orchestration across the integrated Terrestrial Network (TN) and Non-Terrestrial Network (NTN). The pipeline supports automated monitoring and management of both service and network infrastructure, thus mitigating the challenges and delays associated with traditional human-centric service orchestration, which is often fraught with errors. By employing an AI-driven closed-loop mechanism, it fosters a self-managing system that learns from both historical and real-time data, allowing for proactive, data-centric orchestration decisions. Such adaptability is vital for the effective and continuous lifecycle management of applications and infrastructure across the 3D strata, enabling real-time orchestration adjustments in response to the changing network conditions. In the previous WP4 deliverable (D4.1), a zero-touch closed-loop framework powered by artificial intelligence was introduced for E2E service orchestration, aimed at integrating TN and NTN while building upon the advanced NearbyOne [2] orchestration platform. The document detailed the various components of this closed-loop framework, which included the E2E edge orchestration platform, the observability stack, the AI module, the MLflow [3] framework, the essential infrastructure, and enabling interfaces like the Northbound interface. This document, serving as the final WP4 deliverable (D4.2), on the other hand, emphasises the progress made to this closed-loop orchestration framework, specifically focusing on the enhancements to the Northbound interface in terms of infrastructure related endpoints and the modular architecture of the observability stack, both aligned to suit the TN and NTN integration scenario. A1-1 is associated with UC3, which capitalises on contextual awareness achieved through continuous monitoring and predictive analytics. It operates within a framework for autonomous proactive orchestration, where the priorities of safety, reliability, and rapid response are crucial for airspace safety-critical operations, such as air traffic management, emergency response, and other significant operations involving aerial and space resources.
- A1-2 is aimed at addressing the challenges associated with dynamic topology and resource constraints in the unified TN-NTN infrastructure, with an emphasis on orchestration, mobility management, and geo-localisation strategies. In the context of ETHER architecture, Geographic Information Systems (GIS) are a pivotal element for enabling spatially informed Network Function Virtualisation (NFV). This system allows operators to visualise coverage patterns and set geographical boundaries for services offered by satellites that are continuously moving. The spatial intelligence provided by GIS is crucial for effectively managing the NTN dynamics. The architecture includes a GIS extension plugin for ETSI MANO, which is responsible for capturing, analysing, and visualising spatial data. This plugin enhances resource allocation by considering the positions and predict the movements of satellites, integrating effortlessly with other network management systems to improve decision-making processes. Moreover, GIS promotes zero-touch management by allowing real-time adjustments to network resources in response to satellite mobility and spatial data, thereby facilitating automated decision-making. The plugin promotes the traditional MANO by enabling dynamic resource allocation scaling based on satellite movement and continuous service migration. 1. A1-2 is discussed in the second section of Chapter 2 to address the developments achieved in the ongoing efforts stemming from D4.1 and D3.2, aimed at completing the management of mobility and geolocation within WP4. Although A1-2 has the potential to apply to all three UCs, its primary intention is to address UC1.

Chapter 3 is focused on the 2nd development in WP4, which involves the management of a unified RAN and transport systems through distributed network control across multiple layers. This development is divided into six main activities: Software-Defined Networking (SDN)-based Wide Area Network Infrastructure Manager (WIM) for integrated TN-NTN (A2-1), QoS-aware multi-domain 3D routing (SDN-app) (A2-2), Reinforcement learning- based data compression and routing design for multi-satellite systems (A2-3), Model-based 3D network

management and SDN integration (A2-4), Distributed SDN controller placement (A2-5) and an SDN controller implementation for the 3D ETHER Network Architecture (A2-6).

- A2-1 is focused on the SDN-based Wide Area Network Infrastructure Manager (WIM), designed to address the operational complexities of integrated TN-NTN environments, such as dynamic topologies and heterogeneous network segments. The core challenge addressed is the inefficiency of traditional network management and static routing protocols in handling rapid topological changes. The developed WIM provides a unified and programmable Control Plane that enables real-time adaptability, thereby minimizing control plane overhead and ensuring seamless connectivity. The key goals of this activity are to achieve a robust, scalable, and automated system capable of dynamically adapting transport pathways, orchestrating data flows across OVS-equipped nodes, and maintaining service continuity. The functionalities of the WIM are pivotal for the 3D networks envisioned by ETHER, enhancing multi-segment performance and resilience. This work represents a significant advancement of the foundational concepts initially presented in D4.1 [1].
- A2-2 is concentrated on the implementation of QoS-aware routing in multi-domain 3D SDN environment (in the form of cooperating SDN applications). The integration of NTN into terrestrial mobile network infrastructure poses the challenge of a time-varying network topology. In the case of LEO satellites, their high-speed mobility raises several issues such as frequent link availability interruptions including both cross-strata links and Inter-Satellite Links (ISLs), which can lead to the isolation of network nodes. From the standpoint of communication services, these challenges may lead to disruptions in user data sessions and violations of Service Level Agreements (SLAs), or necessitate path reconfiguration, which in turn increases control plane signalling and the operational burden of orchestration processes. Consequently, routing algorithms must utilise external contextual information, including node trajectories, meteorological data, and topology forecasts, to identify the most efficient paths. Thus, it is crucial for mobile network operators to implement intelligent algorithms for network path allocation to minimise operational expenses. A particularly promising approach in this regard involves the use of intelligent Traffic Engineering (TE) agents that employ Deep Reinforcement Learning (DRL) algorithms. Research has demonstrated that, after training, the routing strategies developed by DRL agents surpass traditional TE algorithms in performance. A2-2 proposes a path allocation technique termed 3D QoS Routing (3DQR), which can be employed for selecting single or multiple paths. This approach is tailored for a hierarchical distributed environment based on SDN featuring multiple controllers. It details the methodology and outcomes that continue the investigations presented in the prior WP4 deliverable D4.1 [1]. This activity facilitates the real-time dynamic prioritisation and management of essential data streams, enabling the adjustment of network resources to meet the specific needs of safety critical applications in air-space domains, which is compliant ETHER UC3 targets.
- A2-3 is focused on the Reinforcement learning- based data compression and routing design for multi-satellite systems. A novel joint data compression and routing framework specifically designed for LEO satellite networks with dynamic traffic demands was presented. Leveraging multi-agent deep reinforcement learning (MADRL), the proposed approach coordinates data compression and routing decisions across a distributed network of LEO satellites, with considerations for maximum hop count and buffer constraints. This innovation well aligns with the dynamic topology of UC3.
- A2-4 is aimed at a novel model-based 3D network management and SDN integration. It supports software-defined traffic flows engineering joint with cost-effective handover and route switching in a 3D integrated network, comprised of terrestrial, HAPS and satellite network layers. Aircraft messages are routed through the different 3D network links based on various multimetric routing policies that consider network links' resources in terms of spectrum, power, and available capacity to meet the required communication performance

and expected cost. The conducted work is part of capabilities CA develops, in alignment with UC3 objectives, to enable cost effective message routing and resilient communications in the integrated 6G aeronautical networks. The proposed 3D network routing approaches have been evaluated by using MATLAB simulation environment, considering the performance communication metrics data collected from different flights from Dublin to Heathrow. A2-4 provides a scalable and adaptive control architecture that maintains efficient network operation in highly dynamic non-terrestrial networks (NTNs), particularly in safety- and latency-critical applications

- A2-5 is focused on the distributed SDN controller placement in multi-orbit satellite networks. Integrating LEO and MEO satellite constellations into SDN environments introduces challenges such as dynamic topologies, varying link stability, and constrained satellite resources. In traditional SDN setups, static controller placement schemes result in suboptimal performance due to frequent handovers and limited visibility in satellite-ground communications. These limitations lead to increased flow setup delays and reduced consistency in control plane operations. This activity addresses this by proposing a hierarchical control plane composed of domain controllers placed at LEO or ground nodes and master controllers placed at MEO nodes. A Genetic Algorithm (GA) optimises controller placement based on traffic demand, network topology, and link stability. Furthermore, an incremental update mechanism ensures minimal reconfiguration overhead during topology changes. This approach significantly reduces the number of required controllers and improves flow setup latency. Such an innovation is related to the needs of UC1 and UC3 regarding distributed SDN controller placement for a better management of the integrated ETHER architecture
- A2-6 is concerned with a technical specification for an SDN controller that is focused on integrating NTN and TN network components. This controller orchestrates traffic and communications from the ground domain (such as base stations, RAN and IoT devices), the aerial domain (such as HAPS and UAVs) and the satellite domain (such as LEO and MEO satellites), enabling interoperability and efficient data routing between devices in the ETHER 3D Network. This SDN controller aims to provide better orchestration and communications for ETHER devices, enhancing UC3's need for continuous and resilient data links, as well as service continuity for air-space safety critical operations.

Chapter 4 is centred on the 3rd development, which pertains to AI-enabled E2E network optimisation. This development is segmented into four key activities: AI-enabled 3D network performance and topology forecast (A3-1), Real-time energy-efficient resource allocation and routing (A3-2), AI-based offline energy-efficient user allocation (A3-3) and AI-Driven Beamforming for Integrated Sensing and Communication (ISAC) in Terrestrial-Satellite Networks (A3-4).

- A3-1 is concentrated on the AI-enabled 3D networks performance and topology forecast which employs artificial intelligence to actively monitor and analyse network operations across terrestrial, aerial, and space domains. Through the application of AI, this system can predict and improve network behaviour in real time, effectively addressing the specific challenges presented by high-mobility, multi-layered networks. NETAI has performed a comprehensive univariate Key Performance Metric (KPM) forecasting analysis on the cellular backhaul (CBH) site satellite dataset. The deep learning forecasting techniques utilised can effectively leverage large datasets to achieve superior results compared to conventional statistical methods, while also adapting to variations in KPI dynamics. The A3-1 contributions in this document serve as a continuation and expansion of the same activity presented in D4.1 [1]. The methodologies discussed can be applied to both univariate and multivariate contexts, allowing for the utilisation of intra-KPI dynamics to improve forecasting performance. The activity is planned for implementation in UC3, which facilitates the application of advanced AI algorithms to foresee changes in network performance and

topology in real time, ensuring that the connectivity demands of safety-critical operations are consistently fulfilled with minimal latency and maximum reliability.

- A3-2 is concerned with real-time energy-efficient resource allocation and traffic routing in a unified TN-NTN. The rapid increase in data traffic has necessitated network densification in 6G networks, compelling them to enhance their energy efficiency. This enhancement is primarily aimed at lowering the Operational Expenditures (OPEX) associated with the network and minimising energy consumption to foster sustainable networks. Consequently, as these modifications and additions to future mobile networks become essential, it is crucial to develop strategies for online resource allocation. This strategy encompasses the integrated challenges of online user association, traffic routing, and the placement of xNF within a complex 3D heterogeneous network including terrestrial components such as gNBs and Small Cells (SCs), as well as aerial (HAPS) and space-based (LEO, MEO, and GEO) Base Stations (BSs). These contributions build upon the advancements made in the previous deliverable D4.1. This activity is planned to correspond mainly to UC3 to fulfil the challenging needs of air-space safety critical operations, which in turn boosts both operational reliability and sustainability. However, the main algorithmic logic will be leveraged also for the UC2 demo activity.
- A3-3 is focused on leveraging RL to associate the UEs in a 3D ETHER network in an energy efficient way. In particular, a low-complexity Deep Neural Network (DNN) trained with Proximal Policy Optimization is proposed, capable of associating UEs with the access network based on their service requirements, while minimizing network power consumption as well as avoiding violating the capacity constraints. A three-phase curriculum learning strategy was employed for the DNN to effectively learn the intricate relationships between Physical Resource Block (PRB) allocation and energy consumption. The proposed approach demonstrates significant improvements over current SotA methods, achieving significant improvements in energy saving (by reducing power consumption by up to 47.7%) while meeting user service requirements.
- A3-4 is aimed at the design of precoding and power allocation strategies within a cognitive satellite–terrestrial network architecture that simultaneously enables target sensing and communication services for satellite users. This is achieved through the application of Rate-Splitting Multiple Access (RSMA). A deep learning-based framework is developed to maximise the achievable weighted sum rate of satellite users, while ensuring a minimum beampattern gain for effective target sensing and maintaining an interference threshold with terrestrial operations supported distributed multiple-input multiple-output (MIMO) networks. The formulated problem jointly addresses precoding design, power allocation, and common rate optimisation. A hybrid solution approach, combining deep convolutional neural networks (CNNs) with a semidefinite relaxation (SDR) technique, is proposed to efficiently maximise system throughput while satisfying interference and sensing constraints. The proposed RSMA-based method aims to achieve a superior performance compared to conventional non-RSMA techniques. This activity is primarily aligned with UC2.

Chapter 5 presents the conclusion, highlighting the significant final outcomes of WP4 by mentioning each development and the corresponding activities.

1.2 RELATIONSHIP WITH OTHER ETHER DELIVERABLES

The main objectives of WP4 and the connection between WP4 and the other ETHER work packages have been outlined in the earlier deliverable D4.1 [1]. To illustrate the alignment of the present document with other ETHER deliverables, a brief overview is provided here. This document integrates contributions from various ETHER deliverables, including D4.1, which presents the initial report on zero-touch data-driven network and service orchestration within the 3D ETHER architecture; D2.1 [4], detailing the initial report on ETHER network

architecture, interfaces, and evaluation; D2.2 [5], which covers UCs and KPIs/KVIs; and D2.4 [6], the final report on ETHER network architecture, interfaces, and evaluation. Furthermore, the work described herein utilises the enabling technologies outlined in D3.1 [7] and D3.2, the initial and final report on key technological enablers for seamless and energy-efficient ETHER network operation. Additionally, D5.1 [8] identified several improvements required for the technologies developed in WP4, which have been considered and are documented in this final deliverable.

Furthermore, regarding WP5 this deliverable will contribute to D5.2 by integrating the upgraded WP4 technologies into the demonstration activities that will be reported in D5.3. Regarding this, the innovations that are reported in D4.2 concern both innovations that will be included in the final demonstration to be reported in D5.3 and also innovations that will not be integrated in the final demonstration, but are important for the overall ETHER architecture that was reported in D2.4. More specifically, among the presented innovations of D4.2 the ones that will be integrated in the final demonstration activities are related to **UC1: Flexible payload-enabled service provisioning to semantics-aware and delay-tolerant IoT applications** and **UC3: Air-space safety critical operations**. These innovations per use case that will be included in the final demonstration activities are the following:

UC1: Flexible payload-enabled service provisioning to semantics-aware and delay-tolerant IoT applications

- Geo-localisation management (A1-2)

UC3: Air-space safety critical operations

- AI-driven zero-touch closed-loop orchestration (A1-1)
- AI-enabled 3D network performance and topology forecast (A3-1)
- Real-time energy-efficient resource allocation and routing (A3-2)

Furthermore, the dissemination, exploitation, and standardisation initiatives of WP4 and this deliverable will be highlighted in the WP6 deliverables.

2 ZERO-TOUCH MANAGEMENT, ORCHESTRATION AND MONITORING OF THE VIRTUALISED ETHER NETWORK AND CLOUD CONTINUUM RESOURCES

2.1 INTRODUCTION

This chapter discusses the first development in WP4 concerning zero-touch management, orchestration, and monitoring of the virtualised ETHER network and cloud continuum resources. The main objective is to create an orchestration framework that facilitates seamless service delivery within the 3D ETHER network architecture. This development encompasses two activities: 1) AI-driven zero-touch closed-loop orchestration (A1-1) and 2) Mobility and geo-localisation management (A1-2). Both activities build upon the progress established in the D4.1 deliverable.

- A1-1 is concerned with AI-driven zero-touch closed-loop orchestration. This activity builds on and extends the capabilities of a cutting-edge E2E orchestration platform known as NearbyOne [2] to realise zero-touch orchestration in an integrated TN and NTN. Although some of the extensions have been discussed in previous deliverable D4.1, the rest of the progress to finalise the development are the main outcomes of this activity in D4.2. The main components of this closed-loop orchestration framework include the observability stack, the AI module, the MLflow framework, the requisite infrastructure and the enabling interfaces such as the Northbound interface. These components are integrated with the NearbyOne orchestration platform to facilitate zero-touch closed-loop service orchestration.
- A1-2 is focused on Mobility and geo-localisation management. This activity aims to embed Geographic Information Systems (GIS) within satellite network management to enhance spatial awareness, which is crucial for the effective deployment of location-based services and the anticipation of future network topologies, ultimately ensuring uninterrupted service provision. It adheres to the standardised ETSI MANO framework with kubernetes acting as single node and multi node cluster. A1-2 intends to address the network's dynamic topology by factoring in the spatio-temporal changes in coverage areas, thus promoting dynamic scaling, service continuity, and efficient resource management. The architecture and principal components were outlined in D3.2 and D4.1, while the key outcomes and analyses are presented in D4.2.

The correspondence of these activities and the MANO architecture are shown in Figure 2-1.

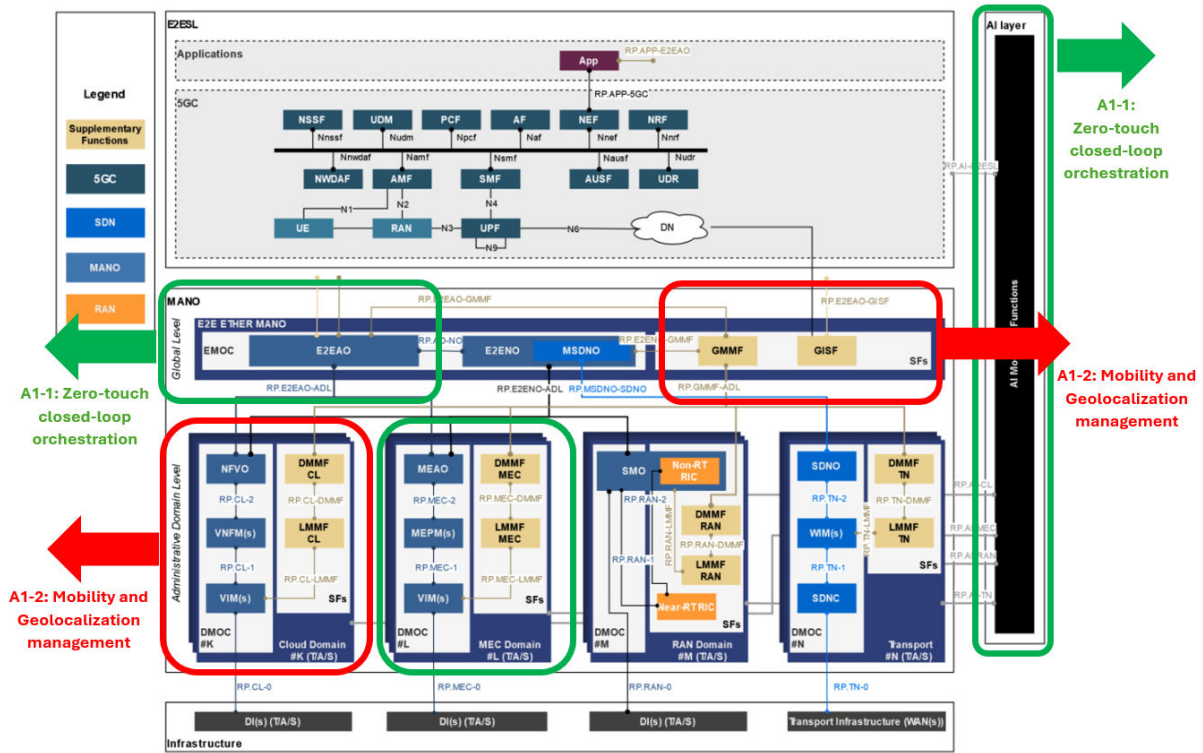


Figure 2-1: Correspondence of the A1-1 and A1-2 activities with the MANO architecture.

2.2 AI-DRIVEN ZERO-TOUCH CLOSED-LOOP ORCHESTRATION

2.2.1 Background

6G aims to achieve ubiquitous connectivity through a 3D architecture integrating terrestrial and non-terrestrial networks, including HAPS and LEO satellites, to overcome the coverage limitations of 5G networks. In addition, in order to support latency sensitive services, edge computing concept will be leveraged to extend computational resources closer to the data sources. However, service orchestration in such an integrated network faces significant challenges due to the complex interplay of hardware, software, and networking components across the different layers, as well as the dynamic conditions imposed by node mobility and fluctuating user traffic. This scenario renders the error-prone human-centric approaches for service orchestration unfeasible, especially in cloud-native scenarios, inherently characterised by massive amounts of monitoring data related to service and distributed infrastructure metrics.

In deliverable D4.1 [1], an AI-driven closed-loop framework for zero-touch E2E service orchestration for an integrated TN and NTN leveraging and extending NearbyOne [2], a cutting-edge E2E orchestration platform was introduced. The different components of the closed-loop framework including the E2E edge orchestration platform, the observability stack, the AI module, the MLflow framework, the requisite infrastructure and the enabling interfaces such as the Northbound interface were discussed, including how these components interact to realise the closed-loop service orchestration automation. This deliverable, D4.2, highlights the extensions made to the closed-loop orchestration framework, targeting to align it with the unique requirements of the integrated terrestrial and non-terrestrial network. Specifically, the deliverable highlights the extensions of the NBI in terms of infrastructure related endpoints, and the modular architecture of the observability stack.

2.2.2 Motivation and Contribution

Given the inherent challenges posed by dynamic network topologies, diverse node characteristics, and spatio-temporal varying traffic volumes in integrated terrestrial and non-terrestrial networks, traditional human-driven orchestration approaches originally designed for terrestrial system become impractical. To ensure seamless E2E service orchestration across such a multi-layered 3D network, a shift toward autonomous, intelligent orchestration is essential. This approach leverages AI and data analytics to automate and refine orchestration decisions throughout the edge-to-cloud landscape. Additionally, in densely distributed and cloud-native infrastructures, robust mechanisms for retrieving service and infrastructure metrics are crucial to support real-time monitoring and proactive orchestration. Towards this goal, activity A1-1 develops an AI-driven zero-touch closed-loop framework for zero-touch E2E service orchestration for integrated terrestrial and non-terrestrial network leveraging and extending a state-of-art NearbyOne, a cutting-edge E2E orchestration platform. This closed-loop architectural framework and the different constituent components were introduced and discussed in deliverable D4.1 [1]. This deliverable enhances the closed-loop orchestration framework in two major aspects:

- Development of infrastructure related endpoints for the northbound interface: In contrast to traditional terrestrial networks, where nodes are typically fixed in number and location, the Integrated Terrestrial and Non-Terrestrial Network (ITNTN) involves mobile nodes that can dynamically change position or temporarily exit the network during service lifetime or deployment window. This dynamic topology behaviour necessitates real-time visibility into node availability, the services currently hosted on them, and their geographical locations. Such data is crucial for enabling intelligent and timely orchestration decisions within the closed-loop orchestration framework. To support this, a set of infrastructure-related endpoints has been developed as part of this deliverable to facilitate extraction of the aforementioned infrastructure related information. These endpoints provide the AI-driven orchestration platform with continuous, up-to-date infrastructure related information, complementing the service-related endpoints previously introduced in deliverable D4.1.
- Hybrid architecture of the observability stack: The dense and distributed nature of the ITNTN network infrastructure particularly in cloud-native environments demands efficient mechanisms for retrieving both service and infrastructure related metrics to support real-time monitoring and proactive orchestration. The mobility of nodes and their diverse storage capabilities further require a metrics collection and aggregation framework that accounts for this heterogeneity. This deliverable proposes a hybrid observability architecture that combines lightweight, decentralised data collection at the edge or resource constrained air and space nodes with centralised, long-term storage and unified metrics querying capabilities at resource rich nodes.

2.2.3 General architecture of closed-loop framework

The general architecture of the closed-loop framework was introduced in D4.1. However, for continuity of this document, this is shown in Figure 2-2. The core components of this architecture are the AI module, the E2E orchestration platform, the cloud-native infrastructure and the observability stack. The Observability stack serves as a complete suite for infrastructure and service metrics monitoring, aggregation, storage, and retrieval. These aggregated metrics are consumed by the AI module to perform analytics and generate orchestration decisions based on application requirements and real-time network conditions. These generated orchestration decisions are communicated to the E2E orchestration platform through the northbound interface, which then enforces these to the underlying infrastructure through the southbound interface. The different components of the closed-loop framework were detailed in section 2.2 of deliverable D4.1. In the subsequent sections, we discuss the

progress made towards enhancing the capabilities of the framework to align with the integrated terrestrial and non-terrestrial network scenario targeted by ETHER.

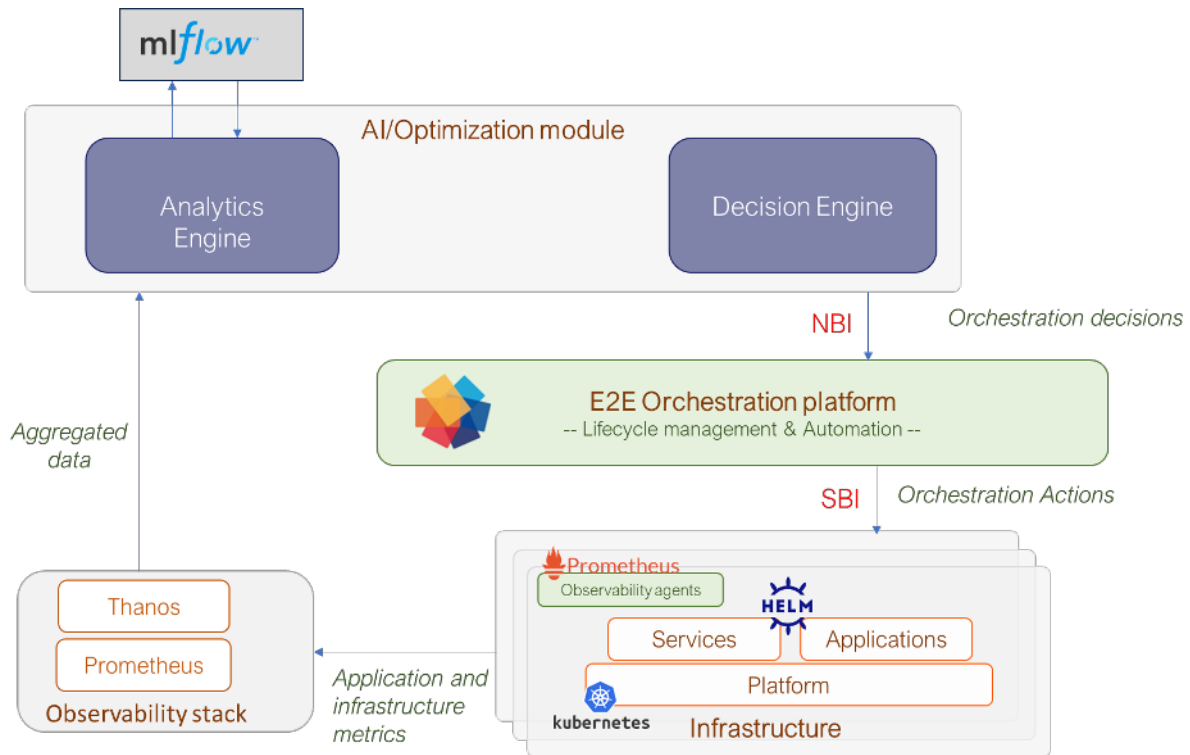


Figure 2-2: Zero-touch closed-loop framework.

2.2.4 Northbound Interface enhancements

The AI module of the ETHER closed-loop framework is designed to maintain the desired state of deployed services, applications, and network infrastructure throughout their lifecycle. It achieves this by triggering network and service reconfiguration decisions such as migration, scaling, and termination to the NearbyOne orchestration platform, which then executes these to the underlying infrastructure. The orchestration decisions from the AI module are communicated to the Orchestration platform through the Northbound Interface (NBI). This interface, which was developed as a RESTful API was introduced in D4.1 including the associated service endpoints to achieve the aforementioned functionality. However, for efficient orchestration in an integrated network, infrastructure related endpoints are crucial. In this way, the capability of the NBI has been enhanced by incorporating 5 infrastructure-related endpoints in addition to the four service-related endpoints introduced in D4.1. A snapshot of these endpoints is shown in Figure 2-3.

Organizations Obtain organizations details. ^		
GET	/organizations	Retrieve details for all organizations accessible to the user
Infrastructure Manage infrastructure interactions including listing of devices and their conceptual organization within sites. ^		
GET	/infrastructure/sites/{siteId}	Retrieve details for a specific site within an organization
GET	/infrastructure/devices/{deviceId}	Retrieve details for a specific device within an organization
Marketplace Manage marketplace interactions including listing of marketplace offerings. ^		
GET	/marketplace/blockcharts	List all available block charts in the marketplace
GET	/marketplace/blockcharts/{blockName}/{blockVersion}	Retrieve specific block chart details from the marketplace

Figure 2-3: Infrastructure related endpoints for the Northbound Interface.

1. GET /organizations

Purpose:

Retrieve details for all organisations accessible to the orchestration platform user. In the context of the NearbyOne orchestration platform, an organisation refers to an entity defined by a group of resources in the orchestration platform and the associated rights to use, manage and modify these resources. This could be a service provider or an infrastructure provider.

Output Parameters:

Organisations: Array of organisation details, each containing the following parameters:

- id: Unique identifier of the organisation.
- displayName: Display name of the organisation.
- description: Description of the organisation.
- sites: Array of site IDs associated with the organisation.
- devices: Array of device/cloudlets IDs associated with the organisation.

GET /infrastructure/sites/{siteId}

Purpose:

Retrieve details for a specific site within an organisation. In the NearbyOne orchestration platform concept, a site is a logical grouping of a set of cloudlets or clusters. A single site may consist of multiple cloudlets but each cloudlet in the underlying infrastructure belongs to a single site. This logical grouping facilitates infrastructure management, for instance by enabling grouping of cloudlets based on given attributes such as location, resource capacity or domain. For instance, in an ITNTN targeted by ETHER, the cloudlets at the different layers (i.e. terrestrial, space and air) can be assigned different site IDs. Such an approach reduces

infrastructure search space during service deployment by only querying sites that meet a given predefined specification.

Input Parameters:

- **siteId**: The unique identifier of the site to fetch. (Required)

Output Parameters:

- **site**: Object containing site details including:
 - **id**: Unique identifier of the site.
 - **displayName**: Display name of the site.
 - **description**: Description of the site.
 - **ancestors**: Array of ancestor site IDs.
 - **sites**: Array of child site IDs.
 - **devices**: Array of device IDs associated with the site.
 - **org**: Organisation ID to which the site belongs

GET /infrastructure/devices/{deviceId}

Purpose:

Retrieve details for a specific device within an organisation, where a device denotes a cloudlet. Given the dynamic nature of the nodes in ITNTN in terms of mobility and resource availability, this endpoint is crucial for validating the status of the cloudlet before the orchestrator makes a placement decision on such a cloudlet.

Input Parameters:

- **deviceId**: The unique identifier of the device to fetch details for. (Required)

Output Parameters:

- **device**: Object containing device details:
 - **id**: Unique identifier of the device.
 - **tags**: Array of key-value pairs associated with the device.
 - **displayName**: Display name of the device.
 - **position**: Object containing latitude and longitude of the device.
 - **specs**: Specifications of the device.
 - **status**: Status of the device.
 - **progress**: Object containing progress details (step and goal).
 - **site**: Site ID to which the device belongs.
 - **cloudResourceChart**: Object containing chart key and latest configuration details.
 - **org**: Organisation ID to which the device belongs.

GET /marketplace/blockcharts

Purpose

List all available block charts in the marketplace. Block

Output Parameters:

- charts: Array of block chart details, each containing:
 - id: Unique identifier of the chart.
 - name: Name of the chart.
 - display_name: Display name of the chart.
 - description: Description of the chart.
 - vendor: Vendor of the chart.
 - categories: Array of categories associated with the chart.
 - all_versions: Array of all available versions of the chart.

GET /marketplace/blockcharts/{blockName}/{blockVersion}

Purpose:

Retrieve specific block chart details from the marketplace.

Input Parameters:

- blockName: The unique name of the BlockChart to retrieve. (Required)
- blockVersion: The version of the BlockChart to retrieve. (Required)

Output Parameters:

- displayName: Display name of the BlockChart.
- description: Description of the BlockChart.
- version: Version of the BlockChart.
- chartYaml: Contents of Chart.yaml which define the chart's metadata.
- overridesYaml: Contents of overrides.yaml which specify configuration overrides.

2.2.5 Observability Stack architecture

The ETHER project targets a 3D architecture realised through integrating terrestrial and non-terrestrial nodes. In this context, non-terrestrial nodes such as HAPs, UAVs and satellites complementing the terrestrial network are mobile and potentially resource constrained. These nodes may have intermittent connectivity (e.g., due to power failures) or physically move out of range before their collected telemetry data can be offloaded to the interested entities such as ML models. Coupled with limited storage and processing capabilities, such mobile nodes are not well-suited for retaining long-term monitoring data requisite for intelligence driven network operation. This renders fully distributed architectures for metrics aggregation unsuited for the envisioned network architecture. To overcome these challenges, a distributed observability architecture is crucial to support lightweight, decentralised data collection at the edge, while enabling centralised long-term storage and unified querying. Considering a multi-layer and multi-node cloud-native deployment scenario, the observability stack architecture proposed in ETHER divides the environment into a centralised Observer cluster and multiple decentralised observer clusters, as illustrated in Figure 2-4.

The Observer cluster functions as the control center of the observability stack. Each Observer cluster, typically located at or near edge or mobile nodes, hosts a minimal Observer Monitoring Block composed of Prometheus for local metrics scraping, paired with a Thanos sidecar for

data forwarding. Due to the mobility and resource limitations of some Observee nodes/clusters, they do not store long-term data locally. Instead, they stream the collected metrics to the centralised Observer cluster, where the full Observer Monitoring Block comprising Prometheus, Thanos, MinIO, and Grafana resides. This centralised setup allows the Observer to aggregate short-term metrics from each Observee cluster and persist long-term observability data using MinIO object storage. Thanos Query within the Observer cluster provides a unified API for querying both real-time and historical data across all clusters.

This design ensures observability in environments where nodes may be disconnected, mobile, or storage-constrained, while still maintaining a coherent, queryable view of system health and performance across the entire network.

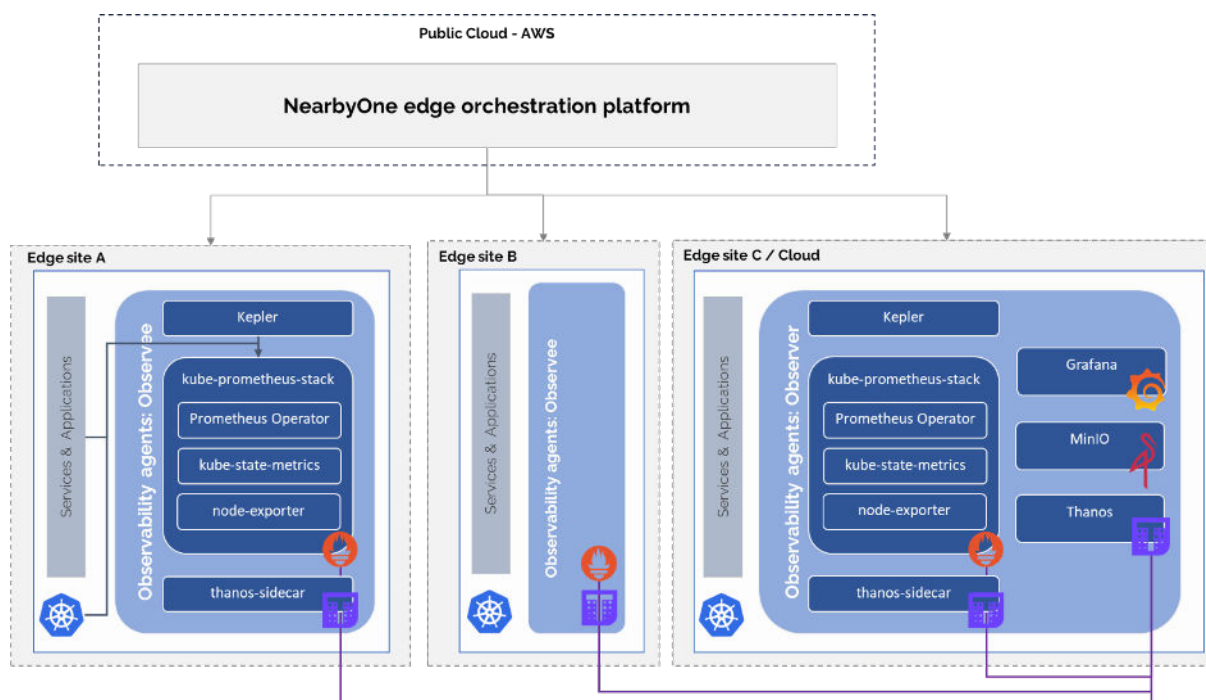


Figure 2-4: Observability stack architecture.

2.3 GEO-LOCALISATION AND MOBILITY MANAGEMENT TOWARDS CONTINUOUS SERVICE

The increasing integration of diverse network infrastructures, particularly the inclusion of dynamic NTN components like satellite constellations, necessitates sophisticated mechanisms for managing service continuity and resource allocation. Central to achieving this is the effective use of geo-localisation and mobility management strategies. Geo-localisation provides the spatial awareness required to understand the real-time positioning and coverage capabilities of network nodes, especially mobile ones such as satellites or aerial platforms.

2.3.1 Background

The European Telecommunications Standards Institute (ETSI) NFV architectural framework [9] is widely adopted for virtualised networks, with Management and Orchestration (MANO) systems like Open Source MANO (OSM) at its core. These systems effectively manage VNFs according to ETSI standards. However, traditional MANO systems face significant challenges when applied to NTNs, particularly those involving Low Earth Orbit (LEO) satellite constellations. The primary limitations stem from their design for static terrestrial

infrastructures. Consequently, they lack the inherent spatial awareness and adaptability required to manage the high mobility of satellite nodes, predict coverage, dynamically allocate resources based on geographical target areas, and ensure service continuity during frequent handovers. Traditional MANO approaches, while effective for terrestrial networks, do not possess the necessary spatial intelligence for dynamic satellite environments, where factors such as limited on-board computational resources and intermittent connectivity further complicate service management. This gap in geographic intelligence restricts the ability to dynamically adjust VNFs based on real-time satellite or user locations, which is crucial for efficient operation in NTN where orbital paths dictate periodic coverage windows.

2.3.2 Motivation and Contribution

The integration of geo-localisation and robust mobility management is critical for ensuring continuous service delivery in dynamic NTN environments, such as those envisioned in the ETHER architecture. The primary motivation is to address the limitations of traditional MANO systems by imbuing them with spatial intelligence. This allows for the dynamic adaptation of network services and resources in response to the constantly changing topology of satellite constellations and their coverage over specific geographical areas.

In this context, GIS become a pivotal element for enabling spatially informed NFV. By leveraging GIS, operators can visualise coverage patterns, define precise geographical boundaries for services offered by moving satellites, and predict future network topologies. This spatial intelligence is crucial for effectively managing NTN dynamics, optimising resource distribution, and automating service provisioning based on satellite visibility and location.

The progression from D4.1 to D4.2 in the ETHER project reflects a decisive transition from architectural theory to laboratory validation in the use of geo-localization for orchestration in 3D integrated networks. In D4.1, GIS-based plugin was proposed as an essential enabler for dynamic resource allocation and seamless handover management across satellites, aircraft, and ground stations, delineating motivations and system requirements. D4.2 builds on this foundation by moving geo-localization into the practical domain: real-time spatial data is directly embedded into orchestration algorithms, and the impact is rigorously tested using simulated laboratory environments and performance comparisons. Notably, D4.2 provides concrete evidence that integrating GIS into orchestration logic not only minimizes unnecessary handovers but also balances network loads more efficiently compared to non-GIS approaches, based on results from simulations using real flight data and quantitative performance metrics.

The main contribution is the development and validation of a novel spatial-aware orchestration framework. This framework seamlessly integrates Kubernetes-based satellite deployments with a GIS-enhanced MANO system. Key contributions include:

- A spatial-aware orchestration framework for satellite networks: This extends traditional MANO (specifically OSM) by embedding geospatial intelligence into its operations, allowing for adaptive configuration based on geographical factors such as satellite positioning and coverage.
- A Satellite Mobility Manager (SMM): This component, in conjunction with GIS, enables dynamic resource allocation based on real-time satellite positions, predicts coverage patterns, and ensures seamless service transitions. It leverages Kubernetes Event-Driven Autoscaling (KEDA) for event-driven scaling based on satellite positioning.
- Spatial-aware algorithms: These algorithms are designed for dynamic resource distribution and efficient mobility management, ensuring services are activated or deactivated based on satellite visibility over predefined geofenced regions during optimal coverage windows.

These developments were fine-tuned for three different scenarios of Kubernetes deployment:

- Single-node deployment: Each satellite operates its own independent Kubernetes cluster.
- Multi-node deployment: All satellites are integrated into a single, unified Kubernetes cluster.
- High-availability deployment: The master node is distributed across both ground and space segments to enhance reliability and fault tolerance.

This approach ensures the algorithms are adaptable and optimized for various satellite network architectures. This approach promotes zero-touch management by facilitating real-time adjustments to network resources based on satellite mobility and spatial data, enabling automated decision-making and efficient, continuous service migration.

2.3.3 Implementation

The implementation of the GIS-enhanced MANO framework involves several key architectural components and processes, as shown in Figure 2-5.

Architectural Components:

- **Orchestration Layer:** OSM manages the lifecycle of VNFs using Helm charts. It coordinates with Kubernetes clusters deployed on individual satellites.
- **Container Management:** Lightweight Kubernetes (K3s) manages containerised applications on resource-constrained satellites, enabling autonomous operation and scaling.
- **Geographic Information System (GIS):** Integrated with OSM, the GIS component uses data like NORAD's TLE datasets for satellite trajectory projection and contact window calculation. It employs PostgreSQL with PostGIS extensions and GeoServer for spatial data analysis, storage, and visualisation via APIs. A custom web browser plugin provides a user interface within OSM.
- **Satellite Mobility Manager (SMM):** This component leverages KEDA for event-driven scaling. It processes contact windows from the GIS and translates them into service activation parameters within Helm charts. Service activation follows a time-window-based approach using TLE projections to determine entry and exit timestamps for service activation. The scaling decision function is defined to activate services with a desired replica count only when the satellite is within the contact window.

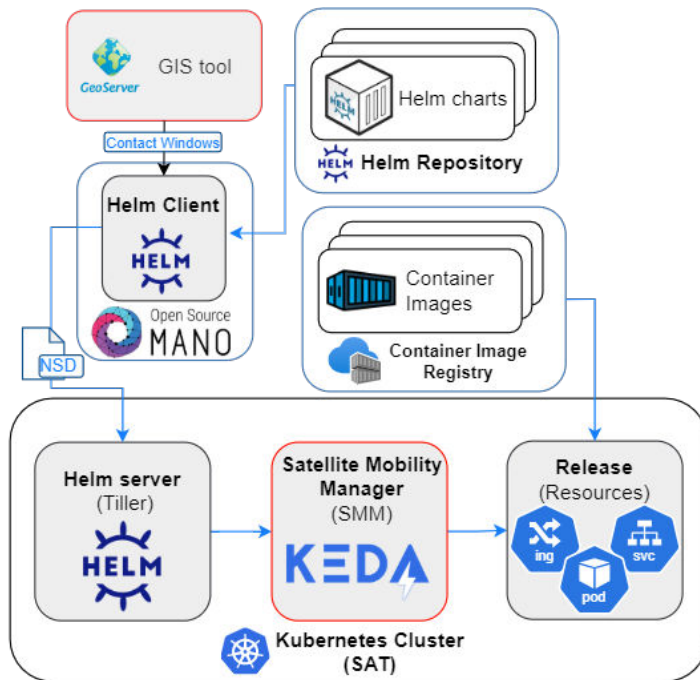


Figure 2-5: GIS-enhanced MANO system for Network Services deployment.

2.3.4 Experimental Validation and Key Results:

The testbed, shown in Figure 2-6, comprised a ground segment (PROXMOX-based private cloud hosting OSM, GeoServer, PostgreSQL) and a satellite segment (ARM-based development platforms like Raspberry Pi running K3s and KEDA). Network conditions were emulated using VSNes [10].

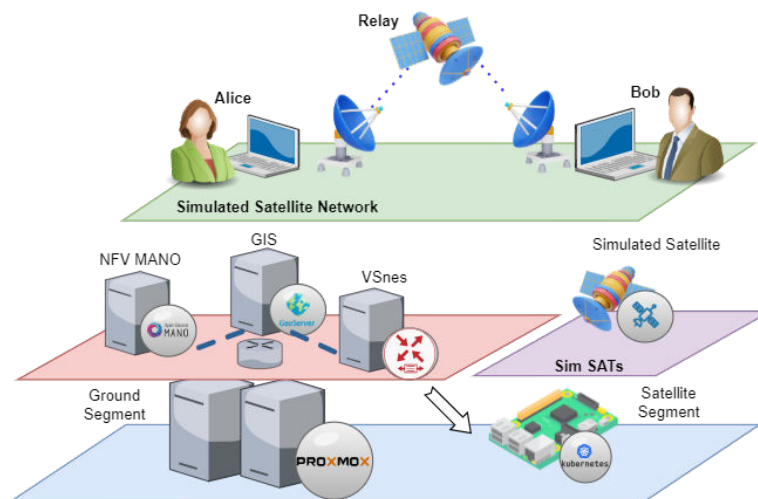


Figure 2-6: Testbed deployment of GIS-enhanced MANO framework.

Regarding resource utilisation, the results of the experiment measurement are presented through Figure 2-7 that illustrate the usage of CPU and RAM on the satellite node. The CPU usage on the satellite node remained low, averaging around 20% during service operation, with predictable spikes observed during the resource-intensive phases of service launch and

deactivation. The consistency of this behaviour across repetitions underscored the predictability of resource consumption. RAM usage, while showing some variability, displayed a clear pattern: median usage was approximately 17.3% during stable service operation, indicating the expected resource allocation for the VNF container.

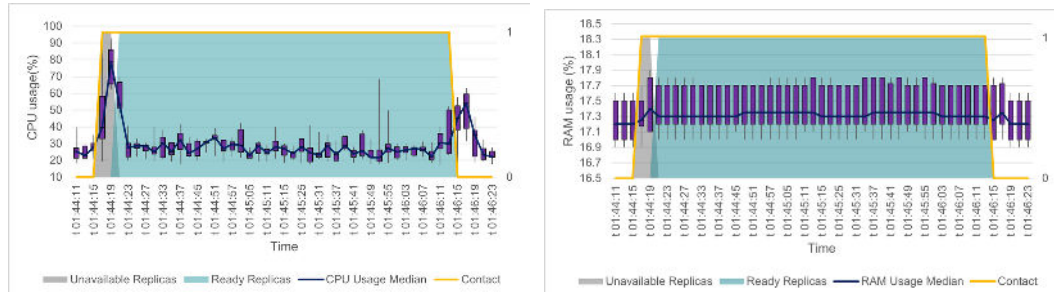


Figure 2-7: CPU & RAM Usage of a satellite during a contact over a target area.

Network performance analysis revealed that data transfer rates followed a bell-shaped curve, logically correlating with the satellite's orbital dynamics as is shown in Figure 2-8. The transfer rate peaked at approximately 130 MB/s when the satellite achieved optimal positioning over the target area. Data transmission commenced once the VNF was activated and ceased upon deactivation as the satellite exited the target area. Correspondingly, latency was inversely correlated with transfer rates, reaching its minimum of 12-15 ms at the point of optimal satellite positioning, which coincided with peak throughput.

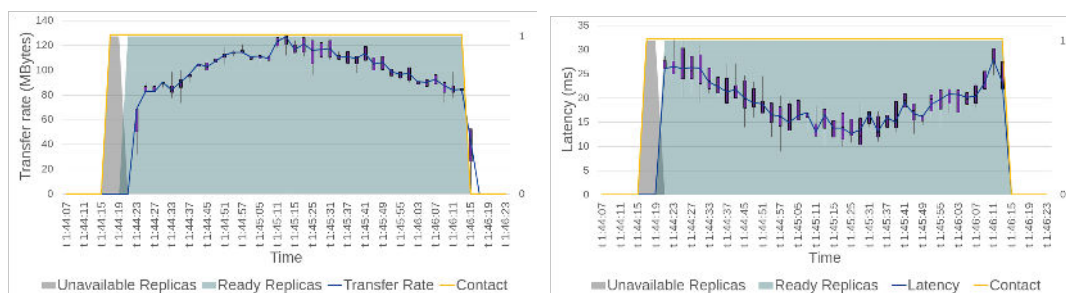


Figure 2-8: Transfer Rate and Latency between two users inside the target area.

In a direct performance comparison shown in Figure 2-9, the GIS-enhanced OSM framework demonstrated superior resource efficiency over traditional MANO implementations. Specifically, the GIS-enhanced system dynamically activated services only when satellites were positioned over designated target areas. This resulted in CPU usage spiking only during these contact periods and dropping to near-zero levels otherwise. In stark contrast, traditional OSM setups kept services continuously active, leading to significant resource wastage when satellites were not servicing a target area, as illustrated by comparative performance graphs.

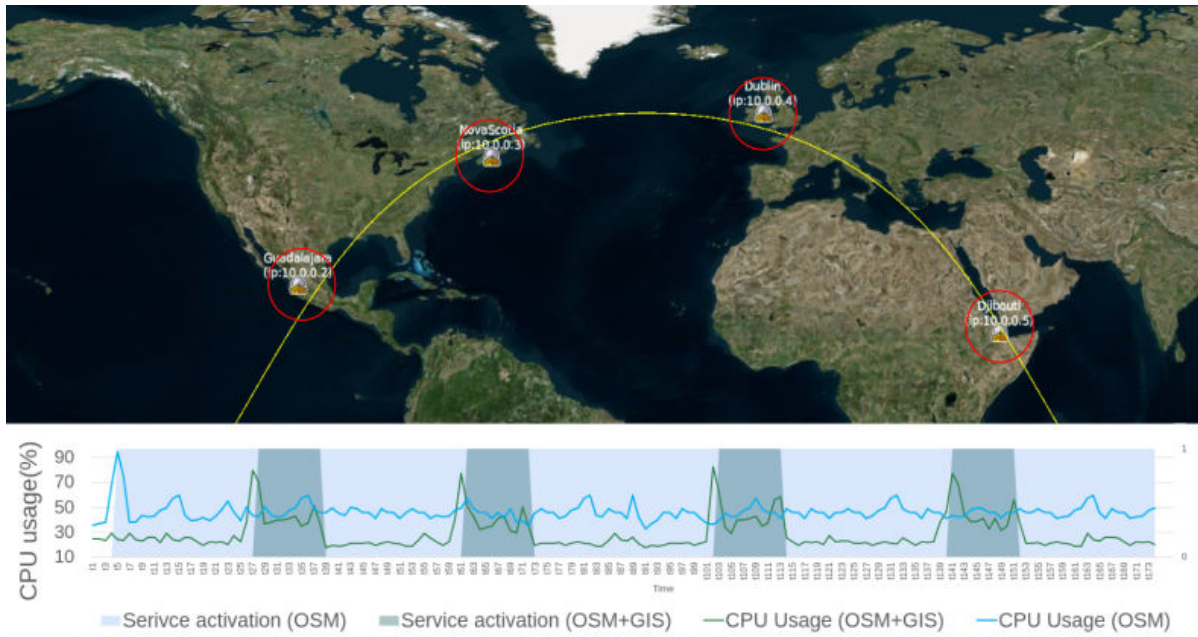


Figure 2-9: Framework analysis OSM with SMM&GIS vs OSM without SMM&GIS.

These results confirm the feasibility and benefits of integrating GIS with MANO systems for efficient mobility management and continuous service provisioning in satellite-based NTN, laying a strong foundation for future 6G applications that require precise spatial awareness and agile resource management.

3 UNIFIED RAN AND TRANSPORT MANAGEMENT LEVERAGING DISTRIBUTED NETWORK CONTROL ACROSS THE ETHER LAYERS

3.1 INTRODUCTION

This chapter delves into the second pivotal development within WP4, which focuses on the unified RAN and transport management via distributed network control across multiple layers. This development encompasses six main activities: SDN-based WIM for integrated terrestrial and non-terrestrial networks (A2-1), QoS-aware, multi-domain 3D network orchestration (A2-2), Reinforcement learning- based data compression and routing design for multi-satellite systems (A2-3), Model-based 3D network management and SDN integration (A2-4), Distributed SDN controller placement (A2-5), and a technical specification for a Software Defined IoT (SD-IoT) controller (A2-6).

- A2-1 focuses on the SDN-based WIM for integrated TN-NTN environments. The WIM aims to resolve operational complexities arising from the dynamic topologies and heterogeneous network segments inherent in such integrated networks. The core challenge addressed is inefficient network management due to rapid topological changes and the limitations of static routing protocols. The developed WIM provides a unified and programmable Control Plane (CP) that enables real-time adaptability to these changes, thereby minimising CP overhead and ensuring seamless connectivity. Key goal of this activity is to achieve a robust, scalable, and automated system capable of dynamically adapting transport network pathways, orchestrating data flows across OVS-equipped nodes, and maintaining service continuity. The WIM's functionalities are pivotal for the forthcoming 3D networks proposed by ETHER, improving multi-segment performance and durability. This activity is a direct continuation and significant advancement of the foundational work and initial WIM concepts presented in D4.1 [1].
- A2-2 deals with QoS-aware, multi-domain 3D network orchestration exploiting the 3D QoS Routing (3DQR) method, based on DRL. This approach is tailored for a hierarchical distributed SDN framework, which involves multiple controllers. The key goals of this strategy are to allocate QoS-meeting paths in 3D network, decrease the chances of path failures, limit the number of path reconfigurations to save computational resources, prolong the duration of network paths to mitigate potential session interruptions due to rerouting and reconfigurations, and to allocate resources efficiently, particularly in the NTN layers where inter-satellite link capacity is limited. Furthermore, the approach incorporates contextual data from external systems to refine path allocation. During the allocation phase, we emphasise QoS differentiation to provide optimal paths for the highest-priority traffic. This activity represents a continuation of the ongoing efforts documented in the prior deliverable D4.1 [1].
- A2-3 is focused on Reinforcement learning- based data compression and routing design for multi-satellite systems. Existing multi-hop routing frameworks focus on optimising path selection and latency but are not adapted to handle high-variability traffic rates. To address these challenges, this contribution presents a novel joint data compression and routing framework specifically designed for LEO satellite networks with dynamic traffic demands. Leveraging multi-agent deep reinforcement learning, the proposed approach coordinates data compression and routing decisions across a distributed network of LEO satellites, with considerations for maximum hop count and buffer constraints. The proposed method optimises the trade-off between data compression and routing efficiency, significantly enhancing end-to-end latency. Simulation results demonstrate the effectiveness of this approach in minimising transmission delays for latency-sensitive applications in LEO satellite networks.

- A2-4 introduces a novel model-based 3D network management and SDN integration. It supports software-defined traffic flows engineering joint with cost-effective handover and route switching in a 3D integrated network, comprised of terrestrial, HAP and satellite network layers. Aircraft messages are routed through the different 3D network links based on various multimetric routing policies that consider network links' resources in terms of spectrum, power, and available capacity to meet the required communication performance and expected cost. The conducted work is part of capabilities CA develops, in alignment with UC3 objectives, to enable cost effective message routing and resilient communications in the integrated 6G aeronautical networks. The proposed 3D network routing approaches have been evaluated by using MATLAB simulation environment, considering the performance communication metrics data collected from different flights from Dublin to Heathrow.
- A2-5 corresponds to Distributed SDN controller placement. The network performance is optimized by strategically positioning SDN controllers in multi-orbit satellite networks using a genetic algorithm based on network topology, traffic distribution, and link stability constraints considering low-earth and medium-earth satellites. The proposed scheme is evaluated through simulations using operational constellation parameters, demonstrating a notable reduction in controller count compared to the benchmark static placement with dynamic assignment
- A2-6 discusses a technical specification for an SDN controller that is focused on integrating NTN and TN network components. This controller orchestrates traffic and communications from the ground domain (such as base stations, RAN and IoT devices), the aerial domain (such as HAPS and UAVs) and the satellite domain (such as LEO and MEO satellites), enabling interoperability and efficient data routing between devices in the ETHER 3D Network. This SDN controller aims to provide better orchestration and communications for ETHER devices, enhancing UC3's need for continuous and resilient data links, as well as service continuity for air-space safety critical operations.

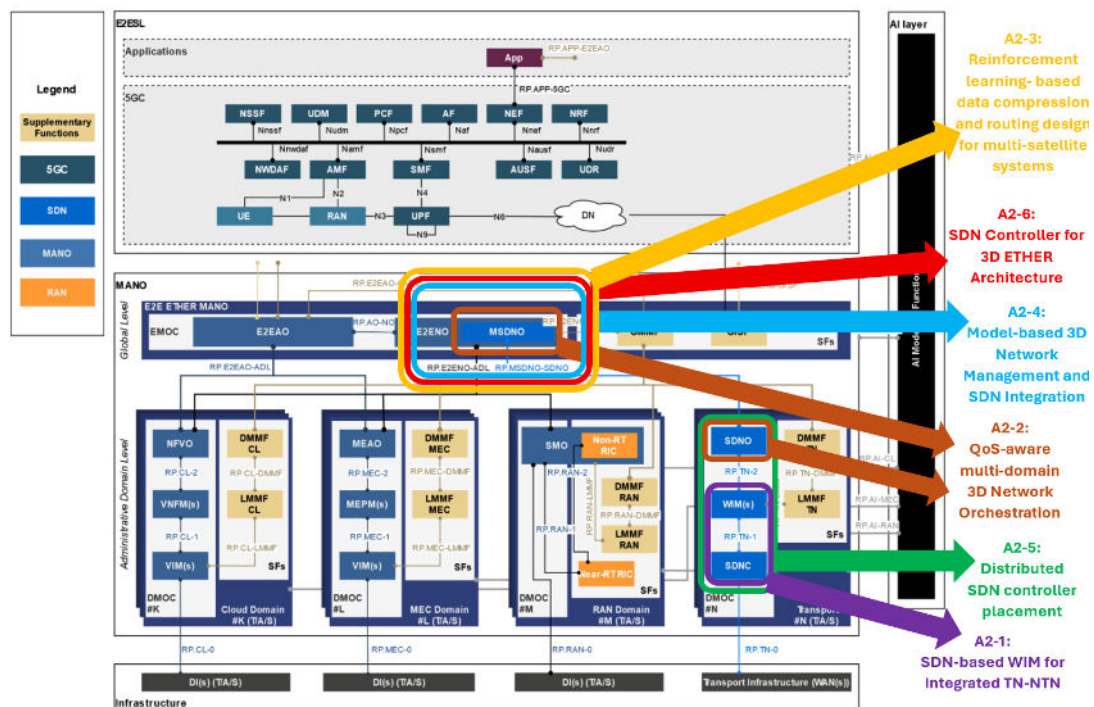


Figure 3-1. Correspondence of the A2-1, A2-2, A2-3, A2-4, A2-5 and A2-6 activities with the MANO architecture.

The correspondence of these activities and the MANO architecture are shown in Figure 3-1.

3.2 SDN-BASED WIM FOR INTEGRATED TN-NTN

The initial report, D4.1 [1], laid the groundwork by discussing the fundamental challenges of TN-NTN integration for transport network management and introduced the concept of an SDN-based WIM as a key enabler within the ETHER architecture. Building upon those initial insights and the preliminary system model presented therein, this section details the subsequent evolution and comprehensive development of the WIM.

3.2.1 Background

Building upon the insights presented in D4.1 [1], the integration of TN and NTN continues to present unique challenges for transport network management, largely due to the non-static nature of NTN nodes such as LEO satellites or aerial platforms. These dynamically changing topologies create conditions under which traditional, predominantly static routing protocols struggle to maintain efficient and seamless services, often resulting in increased overhead and potential disruptions.

SDN has emerged as a fundamental paradigm to address these challenges. By decoupling the CP and data plane, SDN enables centralised, programmable network control and facilitates proactive or reactive reconfiguration when topological changes arise. Within the ETHER architecture, this concept is embodied in the SDN-based WAN Infrastructure Manager (WIM). The WIM coordinates transport resources by interfacing with one or multiple SDN controllers—effectively acting as an SDN application—and orchestrates data flows on Open Virtual Switches (OVSs) to adapt in real time to network state variations. This flexible and adaptive approach aligns with industry trends emphasising automation and resilience for future 6G networks, where TN-NTN convergence is a driving factor in design considerations.

3.2.2 Motivation and Contribution

The SDN-based WIM developed within the ETHER project aims to address the operational complexities of integrated TN-NTN networks, where continuously shifting topologies and heterogeneous network segments necessitate advanced orchestration capabilities. These integrated environments introduce several critical challenges [11, 12, 13]:

- **Rapid topological changes:** With moving NTN elements (e.g., LEO satellites), the frequency of path recalculations can be significantly higher than in purely terrestrial environments, placing a substantial burden on network control mechanisms.
- **Seamless integration of TN and NTN segments:** Maintaining consistent connectivity and unified management across drastically different network segments (e.g., terrestrial backbones vs. satellite constellations) requires a flexible, domain-aware control framework.
- **Increased scalability demands:** As new network nodes (e.g., satellites, drones) join the infrastructure, the complexity of resource allocation and path management grows, which necessitates scalable orchestration tools to avoid performance bottlenecks.
- **Automation and reduced manual intervention:** High levels of automation are critical for improving operational efficiency and reducing both errors and the time required for network reconfigurations.

By leveraging SDN's centralised control and programmability, the WIM component focuses on dynamically adapting transport pathways, thus minimising disruptions and ensuring more robust service continuity. Through this approach, the WIM underpins an efficient, scalable, and

automated framework that handles the diverse requirements of TN and NTN segments alike. Ultimately, the WIM's capabilities will be central to supporting the next generation of 3D networks envisioned by ETHER, enhancing both the performance and resilience of multi-segment connectivity.

3.2.3 System model for the WIM developments

The system under consideration comprises a unified NTN–TN architecture designed to provide resilient connectivity to remote or underserved regions. A high-level system architecture for the SDN-based WIM is depicted in Figure 3-2, where the overall framework is separated into two main layers, including a space layer and a ground layer, with a specialised WIM overseeing the SDN-based transport domain.

In the space layer, LEO satellites, equipped with gNB payloads, form a dynamic mesh. Each satellite maintains vertical Ground-Satellite Links (GSLs) to one or more Ground Stations (GSs) and horizontal ISLs to neighbouring constellation nodes, enabling multi-hop forwarding across the orbital network. User Equipment (UE) in remote locations connects via an emulated radio interface to its serving LEO satellite, injecting uplink traffic into the OVS-based SDN fabric. Traffic may traverse a sequence of ISLs before descending over a GSL to the terrestrial segment.

In the ground layer, each GS acts as the gateway between the space and terrestrial segments, receiving satellite traffic over its GSL(s) and forwarding it into the 5G Core Network (5GCN). Standard 5G CP and UP functions perform session management and initial packet processing. Thereafter, user traffic is steered through an SDN-enabled transport fabric—composed of OVS instances under WIM control—towards the application server, which may reside at the network edge or in a central cloud facility.

As depicted in

Figure 3-2, within this integrated system model, the WIM plays a crucial role by utilising an SDN Controller (SDNC) and protocols like OpenFlow (OF) to manage the SDN-based transport network, which typically consists of OVS instances on LEO satellites and GSs. The WIM receives or computes network state information reflecting the current topology (active GSLs, ISLs, node availability) and translates high-level connectivity requirements or policies into specific OF forwarding rules. This enables dynamic, bidirectional path setup and traffic steering across the complex path involving the NTN (LEO satellites via GSLs and potentially ISLs) as well as TN segments (GS, 5GCN, onward transport). By centralising the orchestration of this dynamic fabric, the WIM addresses scenarios requiring resilient connectivity, particularly where terrestrial options are limited. The WIM's effectiveness lies in its ability to establish, maintain, and adapt communication paths efficiently in response to the constantly changing network topology inherent in LEO systems.

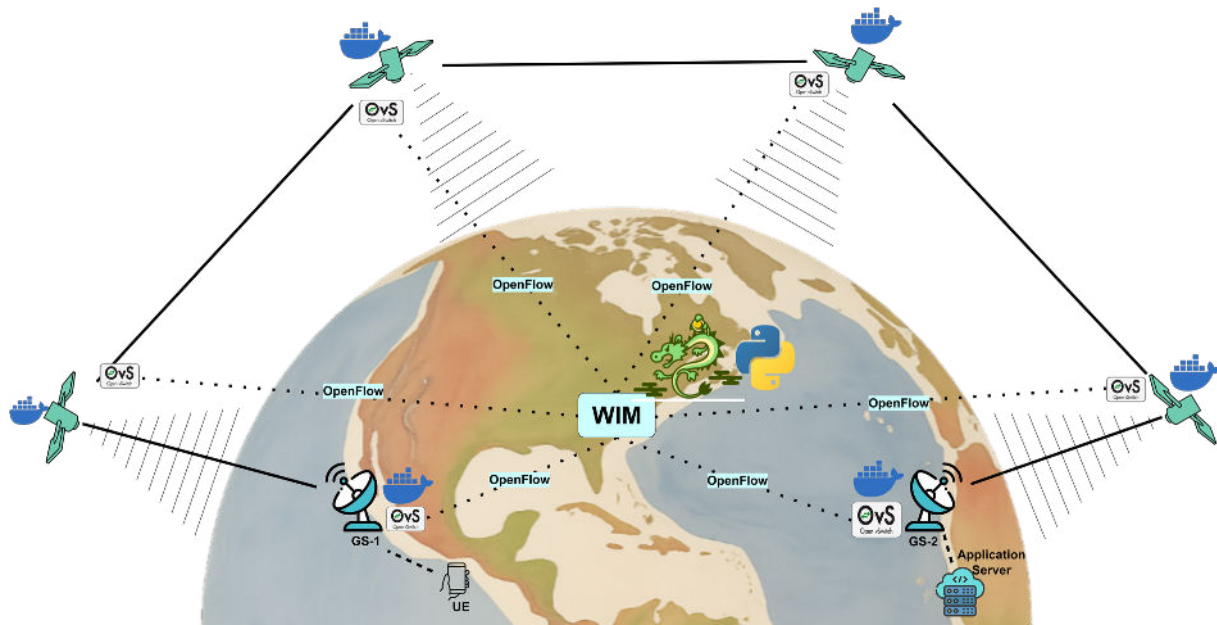


Figure 3-2: High-level system architecture for SDN-based WIM developments.

To focus on core WIM functions, the model excludes other NTN elements such as HAPS, assuming that satellite connectivity is essential between remote UE and application server due to geographical separation. Although propagation delays over both GSLs and ISLs influence E2E performance, detailed cross-layer latency optimisation is beyond this implementation's scope; path selection is governed by basic viability and available link state.

3.2.4 Final Implementation

Since the submission of D4.1, the testbed has been enhanced to create a more capable and reproducible environment for assessing the SDN-based WIM under representative integrated NTN-TN conditions. The dynamic design principle enables the examination of the developed WIM component, while the configuration remains fully deterministic to guarantee repeatable results.

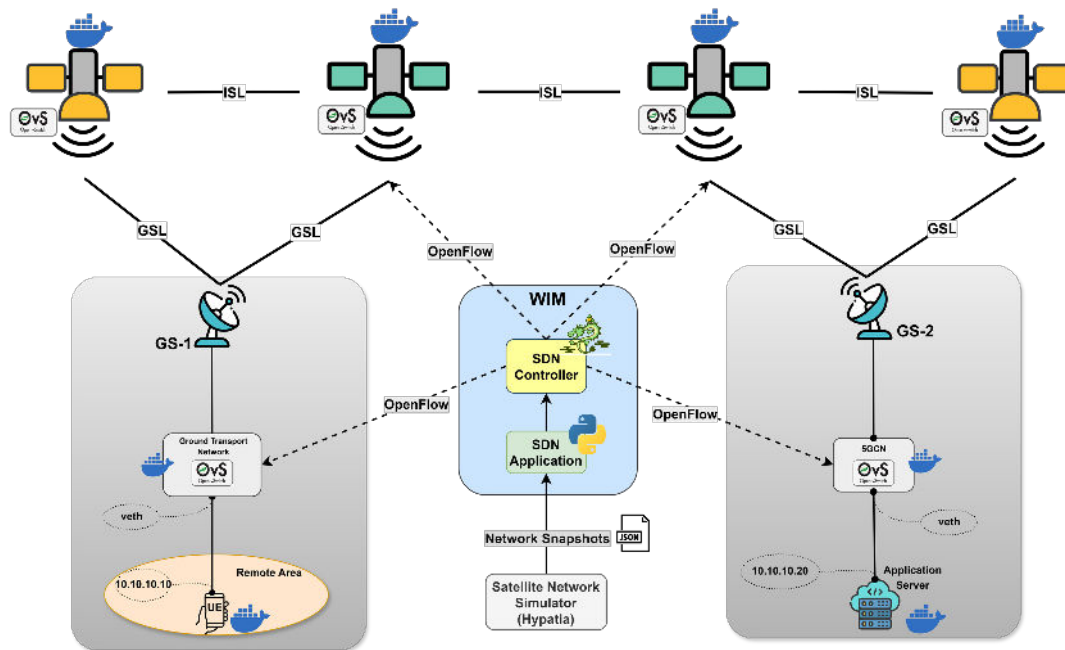


Figure 3-3: Implemented testbed for testing the functionality of the developed SDN-based WIM.

Figure 3-3 illustrates the implemented testbed, a containerised environment designed to assess the functionality of the SDN-based WIM in a controlled and reproducible setting. The entire architecture is deployed within a single Virtual Machine (VM) managed by OpenStack, which hosts a Docker environment. This container-based approach facilitates the efficient emulation of a complex, multi-node network architecture on a single physical host, thereby isolating each network function into a lightweight, standalone entity. This design choice is fundamental to enabling rapid, automated, and repeatable experimentation.

Each key component of the integrated TN-NTN architecture is represented by a dedicated Docker container. These include containers for the UE and the application server, which serve as the traffic endpoints. The terrestrial network segment is emulated by containers representing the Ground Stations (GS-1, GS-2) and the 5GCN. The non-terrestrial segment is formed by a series of *sat-n* containers simulating the LEO satellite constellation. The control plane itself runs on the host itself, which runs the SDN controller and the developed SDN application. A specialised *sat-tran* container is also utilised to manage the overlay network that simulates the satellite transport fabric.

The interconnection of these disparate containers into a cohesive network fabric is achieved through an advanced virtual networking setup. Key network elements, such as the satellites and GSs, are equipped with an OVS instance, transforming them into programmable forwarding planes. The Ryu controller manages these OVS instances via the OF protocol, dynamically installing the necessary flow rules to steer traffic. Basic connectivity between containers and the OVS fabric is established using virtual Ethernet pairs (veth-pair). To accurately model the long-distance and dynamic nature of ISLs and GSLs, the testbed employs Virtual Extensible LAN (VXLAN) tunnels. This overlay technology encapsulates traffic to create a logical network topology that is decoupled from the underlying physical infrastructure, effectively simulating the space and backhaul links within the single VM. This implementation successfully creates a logically distributed topology that mirrors the complexity of a real-world integrated TN-NTN system while retaining the operational simplicity and deterministic nature of a single-host deployment.

3.2.4.1 Automation Scripts

To streamline experimentation and ensure reproducibility, dedicated bash scripts (“create-testbed.sh” and “destroy-testbed.sh”) were developed and refined during the development phase. On one hand, the creation script (create-testbed.sh) automates the creation of required Docker containers (for UE, application server, 5GCN, GSs, LEO satellites, and a dedicated satellite transport switch), the configuration of virtual network interfaces (e.g., veth pairs), the instantiation and setup of the SDNC connection, as well as the configuration of OVS bridges within relevant containers, assigning unique DataPath IDs. Crucially, it also establishes the overlay VXLAN tunnels required to model the satellite backhaul; these tunnels terminate on an OVS bridge within the dedicated *sat-tran* container, simulating the satellite transport network.

An example of the script execution output and the resulting list of created Docker containers is illustrated in Figure 3-4. The destruction script (destroy-testbed.sh), on the other hand, performs the clean teardown of all components when necessary; the corresponding terminal output confirming deletion is shown in Figure 3-5. This automation guarantees consistency across test runs and enables rapid, repeatable experimentation without manual intervention, facilitating efficient development and validation cycles.

```
ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$ bash create-testbed.sh
Please specify the number of satellites as an argument!
Enter number of satellites: 7

| CREATING TESTBED |
-----
INITIALISING CONTAINERS ...
DONE!
-----
SETTING UP GSLs & ISLs ...
DONE!
-----
CREATING & CONFIGURING OVSs ...
DONE!
-----
CONFIGURING IP ADDRESSES ...
DONE!
-----
SETTING UP VXLAN TUNNELS ...
DONE!
-----
| TESTBED CREATED SUCCESSFULLY! |
-----

ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$ sudo docker ps -a
CONTAINER ID   NAMES      IMAGE      STATUS
75e34e2a417f   sat-7      ubuntu    Up 16 seconds
a911e4bbf38f   sat-6      ubuntu    Up 16 seconds
c48cb9c6a4f7   sat-5      ubuntu    Up 16 seconds
81f407526ed0   sat-4      ubuntu    Up 16 seconds
e9570e64a0de   sat-3      ubuntu    Up 17 seconds
f44d2ec368a4   sat-2      ubuntu    Up 17 seconds
fbcad10e83e0   sat-1      ubuntu    Up 17 seconds
14a8b633debd   sat-tran   ubuntu    Up 17 seconds
b3dd52ee3026   gr         ubuntu    Up 17 seconds
ba0e868d1f4c   5gcn      ubuntu    Up 17 seconds
a04ae89cdbf6   server    ubuntu    Up 18 seconds
b3fd68431ef4   user      ubuntu    Up 18 seconds
ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$
```

Figure 3-4: The output of the testbed creation script, followed by the list of created Docker containers.

```
ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$ bash destroy-testbed.sh
| DESTROYING TESTBED |
DELETING CONTAINERS ...
DONE!
CLEANING UP THE HOST VM ...
DONE!
| TESTBED DESTROYED SUCCESSFULLY! |
ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$ sudo docker ps -a
CONTAINER ID   NAMES      IMAGE          STATUS
ubuntu@ether-sdn:~/ether_2025/SDN-based-WIM$
```

Figure 3-5: The output of the testbed destruction script.

3.2.4.2 WIM Control Plane

The control plane logic for the WIM component is realised as an SDN application running on the Ryu SDNC [14]. Ryu was selected for its facilitating characteristics including a lean architecture, comprehensive OF protocol support, and a native Python Application Programming Interface (API) ensuring seamless integration with other components if necessary. The primary function of the SDN application is to ensure resilient E2E connectivity for UE traffic across dynamic integrated TN-NTN platforms.

Transitioning from the initial development phase, which primarily reacted to GSL status changes, the enhanced SDN application leverages comprehensive network state information derived from the Hypatia satellite network simulator [15]. Hypatia provides a framework to simulate and analyse LEO constellations by incorporating their unique characteristics, such as high-velocity orbital motion. It generates discrete network state snapshots (fstate files) at configurable time intervals, capturing the time-varying topology. Each network state snapshot details crucial information including active ISLs, satellite visibility to GSs (active GSLs), and link characteristics based on the simulated orbital mechanics.

The SDN application is designed to ingest these network state snapshots periodically. Using this input, it dynamically updates its internal representation of the network graph, reflecting the current connectivity status of GSLs and ISLs. When required (e.g., upon topology change or for a new flow request), the application calculates the optimal path, defined in our implementation as the shortest path based on current link availability and potentially other metrics like latency, between the relevant network endpoints (e.g., the UE's access point and the application server's network connection). This shortest path calculation explicitly considers both GSLs and ISLs present in the latest network state snapshot. Subsequently, the SDN application translates this computed path into the necessary OF flow rules and injects them into the OVS bridges on the specific LEO satellites and GSs along the selected route. This dynamic, simulator-driven approach enables the WIM's SDN application to ensure seamless connectivity and efficient routing for UE traffic despite the frequent topological changes inherent in LEO satellite networks.

3.2.5 Performance Evaluation

This section details the empirical assessment of the developed SDN-based WIM. The primary objective of this evaluation is to quantify the WIM's effectiveness in maintaining E2E connectivity and managing transport network resources under highly dynamic conditions, characteristic of integrated TN and NTN environments, particularly those involving LEO satellite constellations. The evaluation focuses on KPIs such as Round-Trip Time (RTT), Throughput, Packet Loss and Jitter, which collectively characterise the system's responsiveness, stability, and overall quality of service provisioning.

The measured KPIs include:

- **RTT:** This is the total time taken for a data packet to travel from a source to a destination and for an acknowledgment from the destination to be received back at the original source. It is a critical performance metric, especially for interactive and real-time applications such as online gaming, video conferencing and voice over IP, where lower RTT contributes to a better user experience and system responsiveness.
- **Throughput:** User Datagram Protocol (UDP) Throughput refers to the actual rate at which UDP data is successfully transferred over a network path, typically measured in Megabits per second (Mbps). This KPI is related to the network's capacity for delivering a certain volume of data for applications like streaming or bulk data transfer.
- **Packet Loss:** Packet Loss is the percentage of packets that fail to reach their destination. It significantly impacts application performance, particularly for UDP traffic where lost packets are typically not retransmitted, leading to quality degradation in real-time services.
- **Jitter:** E2E Jitter or One-way Jitter refers to the variation in the transit time of packets in a data stream. It is a critical measure of network stability, as high jitter can cause disruptions such as distortion or buffering in real-time applications like video streaming. Therefore, low and stable jitter is highly desirable.

3.2.5.1 Evaluation Scenario

The WIM's performance was assessed within a virtualised testbed environment, which was explained in detail in Section 3.2.4.1. A network of Docker containers was established representing essential entities, including a UE residing in GS-1 domain, an application server and 5GCN functions both residing in GS-2, and eight LEO satellites forming a single orbit. The dynamic LEO satellite topology is extracted from the fstate JSON file, containing network state snapshots generated by Hypatia. This file is consumed by the SDN application to ingest these topological snapshots and perform the path provisioning task during the simulation period.

The interval for processing new network state snapshots from Hypatia by the SDN application was set to 720 seconds (12 minutes). This inter-snapshot time is chosen to simulate realistic handover frequencies in LEO constellations. The service duration of a single LEO satellite for a GS directly beneath its path is determined by its orbital period (T) and the number of satellites (N) in the same orbital plane designed to provide continuous coverage. The orbital period for LEO satellites, for instance, at altitudes around 550 km, is typically in the range of 90-100 minutes [16]. For an 8-satellite ring constellation, such as the one emulated, the effective service time per satellite before a handover to the next is approximately T/N . With an orbital period of roughly 95 minutes (for a ~550 km altitude), this yields a handover interval of approximately 12 minutes [17]. The 12-minute inter-snapshot interval used in this study is therefore representative of the dynamics in such LEO systems, which inherently involve frequent handovers.

To measure the different KPIs and reflect a more realistic operational environment, where network elements concurrently handle diverse traffic types, the following measurements were performed simultaneously throughout the respective experiment durations:

- **hping3:** RTT was measured using the hping3 tool [18], by sending ICMP echo requests at a 0.5-second interval between the UE (IP address 10.10.10.10) and the application server (IP address 10.10.10.20) and RTT values were extracted from the hping3 log file.
- **iperf3:** A continuous UDP traffic flow at a target rate of 10 Mbps was generated between the UE (IP address 10.10.10.10) and the application server (IP address 10.10.10.20) using

the iperf3 tool [19]. Server-side logs were collected for deriving three different KPIs: UDP Throughput, Packet Loss, and Jitter.

Upon each new network state snapshot provided by Hypatia, the SDN application extracted the shortest path, removed the old OF rules from the OVS instances and deployed the necessary new rules to those along the corresponding shortest path so that they can manage the flow of both IP and ARP traffic between the UE and the application server.

3.2.5.2 Results and Discussion

The performance of the SDN-based WIM was analysed under the previously described dynamic scenario based on the collected data for E2E latency, throughput, packet loss, and jitter. The observed results for each of these KPIs is provided and discussed below.

RTT over Time

The RTT, as illustrated in Figure 3-6, demonstrates the characteristics of the system latency measured by hping3 over the 4-hour (14,400-second) experiment duration. The raw RTT values exhibit inherent fluctuations, a direct consequence of the dynamic path changes within the LEO constellation due to the LEO satellite movements, resulting in topology changes. These changes occur at 12-minute intervals based on the Hypatia snapshots value. The application of a 120-point rolling median (plotted in dark blue) smooths these instantaneous variations, revealing a more stable underlying latency trend. Additionally, the mean RTT recorded throughout the experiment is 4.190 ms, as indicated by the dashed red line.

Occasional transient spikes in RTT are observed following a topology change, however, as can be seen in Figure 3-6, the system quickly returns to its baseline. This behaviour demonstrates that the developed WIM is able to install new OF rules efficiently enough to limit the duration of any significant service degradation. The cyclical pattern in these fluctuations is consistent with the repetition of the sequence of Hypatia-defined network states over the experiment duration.

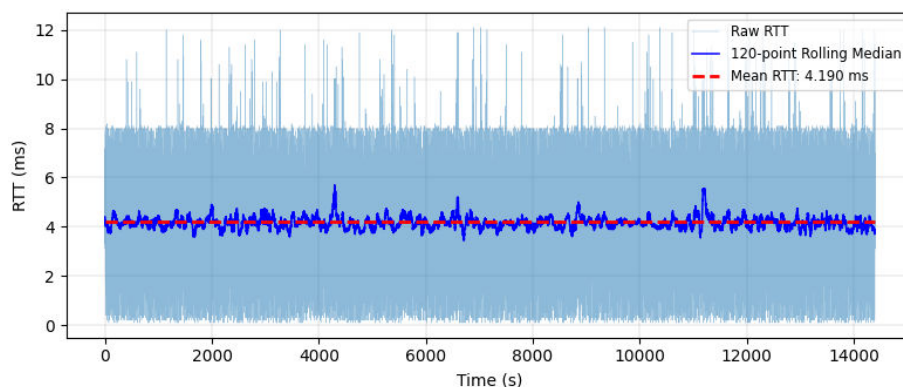


Figure 3-6: RTT during a continuous hping3 experiment for four hours, showing raw ICMP probe measurements, a 120-point rolling median, and the mean RTT.

UDP Throughput and Packet Loss over Time

The UDP throughput and packet loss metrics for the application server (iperf3 server-side) provide insight into the data transmission quality over the approximate 4-hour (14,400-second) test period. As the UE (iperf3 client-side) performance consistently demonstrated the ideal 10.00 Mbps throughput with no packet loss, this discussion focuses on the server-side measurements illustrated in Figure 3-7. The top panel shows that the application server

throughput is highly stable, consistently hovering quite close to the 10.00 Mbps target rate established by the UE, with minor and infrequent downward spikes.

The bottom panel of Figure 3-7 illustrates server-side packet loss. For most of the experiment, packet loss remains at zero. However, distinct and brief spikes in packet loss are observed, for instance, around the 2800s, 8500s, and two events near the 13800-14300s marks, each reaching approximately 0.11%. These loss events are transient and directly correlate with the intervals of path recalculation and OF rule deployment by the WIM, triggered by the topology changes. The minimal magnitude and brief duration of this packet loss suggest that service interruption during these adaptive phases is negligible.

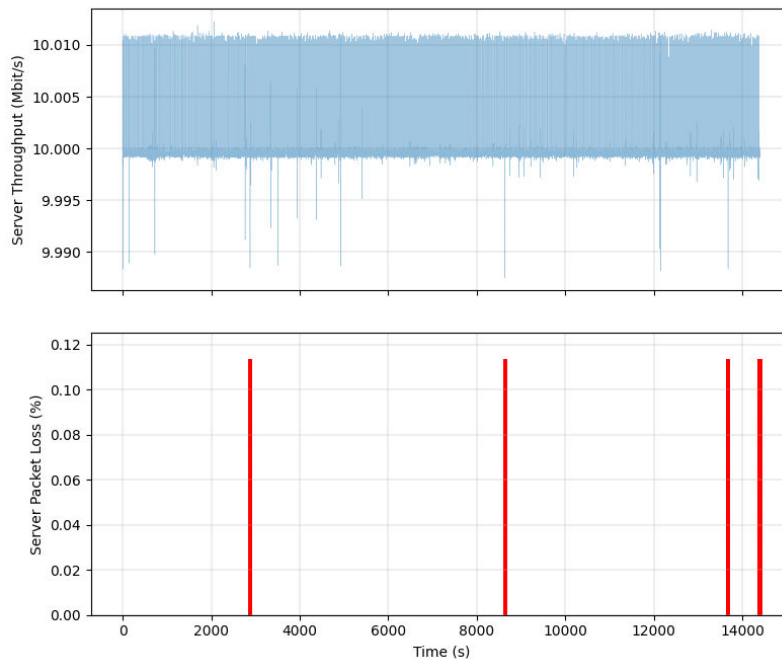


Figure 3-7: Server-side raw and mean values for UDP throughput as well as the packet loss percentages during a continuous 10 Mbps iperf3 experiment session.

Figure 3-8, further details the statistical distribution of the server-side UDP throughput. The Interquartile Range (IQR), which represents the spread of the central 50% of the data (i.e., from the 25th to the 75th percentile), is observed to span a narrow range, indicating that the server throughput is tightly clustered. The median throughput is 9.9995 Mbit/s, while the mean throughput is 9.9999 Mbit/s. The mean being slightly higher than the median suggests a slight positive skew in the throughput distribution, potentially due to a few instances of throughput values at the higher end of the observed range. However, the very small difference between these two central tendency measures, in conjunction with the compact IQR, signifies a highly stable and consistent throughput performance. This performance closely aligns with the target rate of 10 Mbps and demonstrates minimal variation throughout the test period, thereby confirming the findings derived from the iperf3 time-series data.

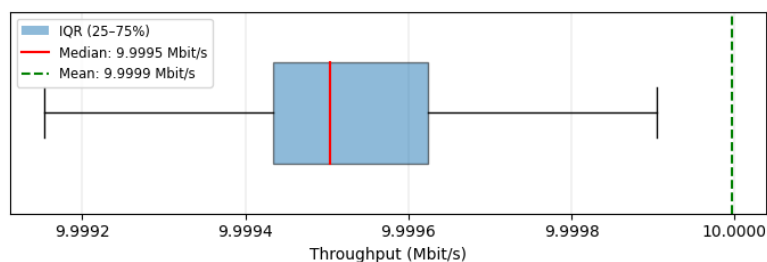


Figure 3-8: Distribution of server-side UDP throughput over the 4-hour experiment, illustrating the median, IQR, and mean.

E2E Jitter

As Figure 3-9 presents, the observed mean jitter over the 4-hour experiment duration is exceptionally low at 0.01 ms, with a 120-point rolling median (blue line) closely tracking this value. Although the jitter generally remains minimal, distinct spikes are evident, for instance, around the 10,500s, 11,500s, and 13,500s marks. These spikes correlate with periods of network adaptation following the 12-minute interval topology changes when new paths are being established. However, the transient nature of these spikes and the low overall average jitter indicate acceptable E2E path stability once a new path is established after a change in topology is detected by the SDN controller.

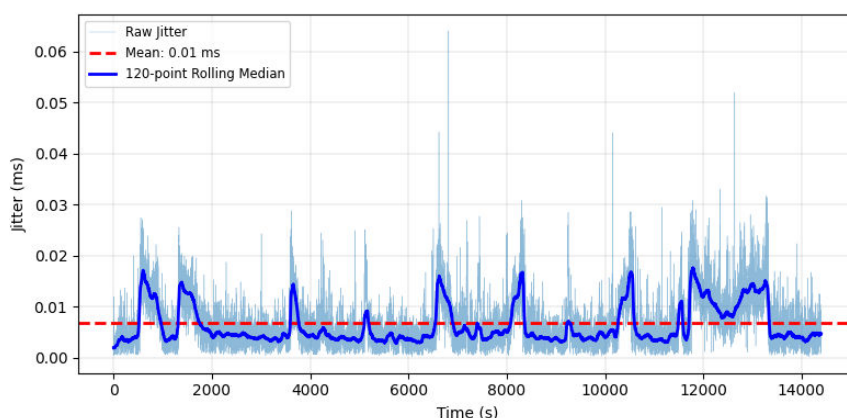


Figure 3-9: One-way UDP jitter measured at the server over the 4-hour experiment, showing raw Jitter values, a 120-point rolling median, and the mean value for Jitter.

In summary, the performance evaluation demonstrates that the SDN-based WIM effectively maintains E2E connectivity and service quality in a highly dynamic integrated TN-NTN environment. The system achieves stable throughput close to the target rate (with optimal performance on the UE side), maintains a low average RTT (around 4.190 ms), and exhibits very low average jitter (0.01 ms). The disruptions in terms of server-side packet loss and jitter spikes are transient and directly associated with the WIM's adaptation to topological changes occurring at defined intervals (e.g., 12 minutes), with the controller's processing of these changes and subsequent path reconfiguration being efficient enough to ensure rapid service restoration. These findings validate the developed WIM's capability to provide seamless E2E connectivity and automate network operations in response to the inherent dynamism of integrated TN-NTN systems.

3.3 QOS-AWARE MULTI-DOMAIN 3D ROUTING (SDN-APP)

This section describes the final version of 3D QoS-aware routing algorithm for hierarchical multi-domain SDN networks, such as ETHER Transport Network segment described in D2.4. Hereby, the final refined approach is presented, which is based on the high-level concept outlined in D4.1. The method formulation is followed by evaluation in simulation-based testbed using realistic topology data and analysis of applicability in the context of avionic communication.

3.3.1 Background

Fast mobility of LEO satellites poses multiple challenges in terms of routing and QoS path allocations – the latter being essential for the implementation of carrier-grade aviation communication. Some of the key challenges include [20] [21] [22]:

(C1) Routing protocols convergence – node mobility in NTN leads to frequent topology changes, requiring constant routing updates that cause instability and signaling overhead (frequent updates of routing tables and link costs) [23]. While SDN enables flexible traffic steering [23], the original single-controller deployments lack CP scalability. Multi-controller SDN architectures partly address this issue at the cost of increased complexity, as E2E QoS routing requires both coordination across controllers and their optimal placement.

(C2) Temporal and predictive routing – mobility causes variable link quality (e.g. bandwidth jitter due to ISL occlusion) and link disruptions. To maintain QoS without excessive signaling, routing should leverage link availability prediction and context-aware, time-scheduled paths (e.g., based on satellite orbits and interface alignment).

(C3) Resilience - increasing network resilience and minimising communication cutoffs are core requirements to enable QoS-driven services. The potential solutions include multi-path and/or node/edge-disjoint routing [24], in-switch buffering mechanisms (i.e., store-and-forward) [25] [26], or adoption of predictive routing schemes.

(C4) Optimisation - conventional TE algorithms cannot be used effectively in NTN due to slow convergence time. The emerging TE methods should consider both QoS constraints and effective traffic distribution to address limited ISL and feeder links capacity. While AI-driven TE is a promising solution, SoTA models usually consider a fixed network structure, which is not scalable, especially in large-scale LEO systems. Hence, optimisation algorithms should extract the information from the nodes and link relationships rather than fixed graph structures to avoid over-complex and monolithic models.

(C5) Asset heterogeneity – supporting E2E QoS provisioning in heterogeneous multi-network 6G environment (i.e., “Network of Networks” [27]), will require dynamic TE, cross-domain cooperation, adoption of insightful abstractions (to model lower-layer dependencies) and intelligent automation to manage diverse assets and meet service demands.

Hereby, we propose 3DQR, an intelligent QoS-aware routing solution for SDN-based 3D networks that optimises load distribution and aligns with operator service priorities. Key features include:

- Adopting a hierarchical distributed SDN architecture, compliant with ETHER architecture, and modular routers that can be deployed as Supplementary Functions (SFs). The approach enables heterogeneous network domains and dynamic attachment of new domains. The two-level control enables flexible routing optimisation at the domain and E2E level [C1, C4, C5].

- Leveraging topology predictions to improve AI reasoning [C2] and selection of reliable links within a specific time frame [C3].
- Combining Double Deep Q-Networks and Graph Neural Networks to (i) provide intelligent routing and path allocation in 3D; (ii) enable variable size input; and (iii) combine both network- and flow-level metrics for optimal routing decisions [C4];

Hereby, we show that 3DQR is well-suited for aviation communication and other QoS-sensitive applications, being in-line with both ETHER UC3 as well as exploitation in emerging integrated 3D 6G networks.

3.3.2 Concept description

The components of the 3DQR are shown in Figure 3-10.

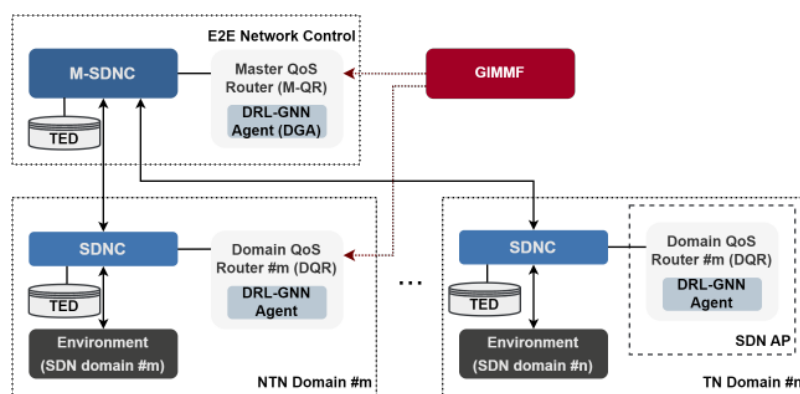


Figure 3-10: High-level view of the 3DQR concept and components.

3DQR operates in multiple SDN domains across terrestrial, aerial, and satellite strata (one or more domains per stratum). Each domain is controlled by a dedicated SDNC (one per domain) that implements domain-specific CP protocols (such as OF [28]) for full control over domain traffic flows and monitoring for accurate network state information acquisition. The E2E control is implemented by the logically centralised and physically distributed global-level Main SDNC (M-SDNC) acting as the umbrella controller for the underlying SDNCs. M-SDNC/SDNCs consume routing services exposed by 3DQR routers: Main QoS Router (M-QR) and Domain QoS-aware Router (DQR), where both embed TE DRL-GNN Agent (DGA). The role of DGR is to provide optimal routes for the flows traversing the domain and domain load distribution, while GQR handles cross-domain routing and cross-domain load balancing (i.e. distributing loads across inter-domain links and per-domains). To maintain SDN CP scalability M-SDNC routing decisions are based on the overlay network view, which refers to:

- Monitoring — instead of link-level metrics, SDNCs calculate the parameters of overlay links between domain ingress/egress nodes denoted as Border Nodes (BNs). While reducing monitoring data volume, it conveys the information about domains' capacity to allocate new flows.
- CP operations — as M-SDNC sees only BNs, overlay links within domains, and links interconnecting domains, the End-to-End Path (E2EP) is enforced by instructing individual SDNCs to set up Intra-Domain Paths (IDPs) between BN pairs. Installing flow entries at BNs establishes inter-domain paths.

Topology data, monitoring metrics, and flow history are stored in local and E2E Traffic Engineering Databases (TEDs), accessible to TE applications as needed. TEDs use retention strategies tailored to TE mechanisms, e.g., DRL-GNN agents (details out of scope of this deliverable). The 3DQR concept is compliant with ETHER architecture described in deliverable D2.4, i.e., M-SDNC/SDNC can be a part or an instance of M-SDNO/SDNO, SDNC domains correspond to ETHER domains, whereas TED and M-QR/DQR are concurrently Supplementary Functions and SDN applications (in SDN Application Plane).

The goal of 3DQR is to perform optimal routing and E2E path allocation in the integrated 3D network, considering three major aspects: degree of link traffic distribution, flow QoS constraints, and flow importance (determined arbitrarily by the network operator). These objectives are achieved by the federation of DQRS and a centralised M-QR that provide optimal domain-level and E2E paths on SDNCs/M-SDNC requests (via interaction over North Bound Interfaces). The best paths are selected by DGAs implementing the Double Deep Q-Network algorithm [29] and embedded in each DQR/M-QR (cf. Figure 3-11).

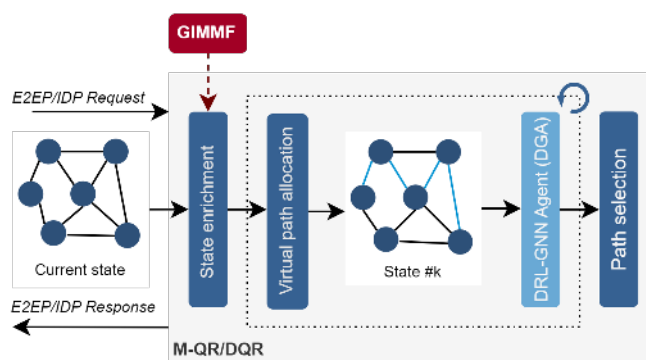


Figure 3-11: Data flow within M-QR/DQR.

Upon receiving a path request, DQR collects the latest network state information (e.g., OF-based statistics) and extends the state graph by additional node/edge features. The most important addition includes nodes/links expiry information – the probability of nodes' reachability or existence of ISLs/feeder links in the considered time frame. This data is provided by Geographic Information System-based Mobility Management Function (GIMMF) – an entity that estimates NTN topology (active ISLs, FLs, etc.) and calculates a persistent network graph within a time frame (i.e., containing only the set of nodes and links that is not expected to change), using orbital/trajectory parameters of NTN nodes. DQR then computes k shortest paths based on delay, virtually allocates the paths according to QoS requirements, and selects the path that leads to the highest Q-value. To tackle the issue of variable size of network states, DGAs embed GNN architecture, which enables efficient extraction of the information about the relationships between nodes and links regardless of topology shape and size. SDNCs and M-SDNC are solely responsible for verifying the feasibility of a path and its enforcement while the reasoning is left to DQRs. Moreover, as each satellite/aerial node can act as an access point capable of serving the UE, they are considered BNs. Finally, QoS classes are aligned with the 3GPP definition, i.e., 5G QoS Identifier (5QI) [30] to enable straightforward integration with 3GPP networks.

3.3.3 E2E Routing and Path Allocation Approach

Paths are requested by a generic External Requester (ExtReq) entity (#1) - UP entities, MANO, or M-SDNC applications through M-SDNC APIs (either higher-level request or internal M-SDNC/SDNC process). The E2EP allocation process is shown in Figure 3-12.

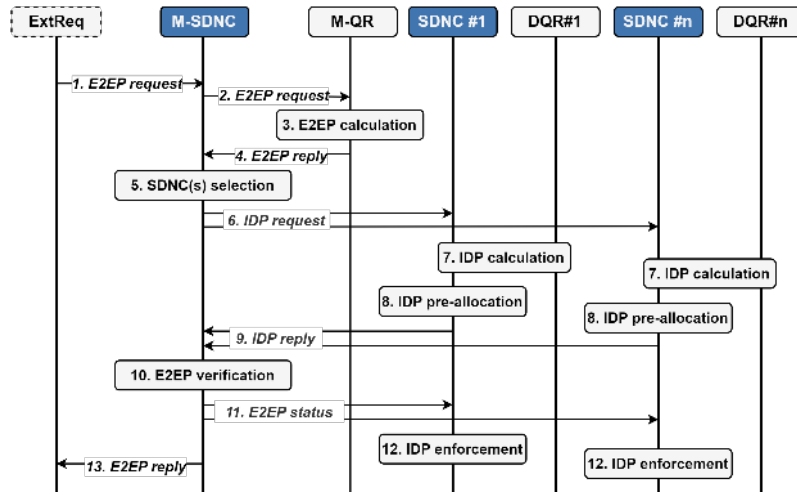


Figure 3-12: E2EP allocation process in 3DQR.

M-SDNC begins by sending the E2EP request—with source, destination, and QoS class info—to M-QR (#2). M-QR computes the E2E path (#3) using BNs and overlay links visible at the M-SDNC level (#4). After verifying feasibility, M-SDNC splits the path by BN domain membership (#5) and requests Intra-Domain Path (IDP) calculations from the respective SDNC-DQR pairs (#6, #7). Each SDNC checks the feasibility of its IDP and reserves the necessary resources (#8). Once successful, SDNCs notify M-SDNC with confirmation and QoS metrics of each IDP (#9). M-SDNC then verifies whether the combined IDPs meet the overall E2E QoS requirements (#10). If they do, it informs the SDNCs to proceed with final path enforcement (#11, #12) and notifies the external requester of successful allocation (#13). If any IDP cannot be allocated—due to mobility, failures, or other dynamics—SDNC reports this, prompting M-SDNC to roll back any partial allocations. The same rollback occurs if E2E verification fails.

While the centralised approach increases procedure duration, it ensures E2EPs even in fragmented NTN topologies or multi-operator scenarios. The M-SDNC's global view supports rerouting through TN or other NTN segments to bridge disjoint networks and fulfil the allocation.

3.3.4 3DQR Algorithm

3.3.4.1 System Model

QoS routing is modelled as a time-expanded minimum-cost multi-commodity flow problem. The network graph $G(V, E, t)$ is composed of a set of sub-graphs, each described by a unique domain identifier $d \in \mathbb{N}$. Each one is either time-variant $G^d(V^d, E^d, t)$ (space and aerial domains) or time-invariant $G^d(V^d, E^d)$ (terrestrial domains). Each graph is constituted by nodes/edges sets: $V = [v_1, v_2, \dots, v_j]$ and $E = [e_{12}, e_{13}, \dots, e_{ij}]$, with e_{ij} representing a bidirectional edge between vertices v_i and v_j . Edge properties are described by features $F = \{f_{ij} | (v_i, v_j) \in E \in \mathbb{R}^{|\mathbb{E}| \times m}\}$ (m – number of features) that include total capacity c_{ij} , total Guaranteed Flow Bit Rate (GFBR) - cf. [30], clause 5.7.2.5 - allocated to flows b_{ij}^{GFBR} , total Maximum Flow Bit Rate (MFBR) - cf. [30], clause 5.7.2.5 – that can be consumed by flows b_{ij}^{MFBR} , latency l_{ij} , packet error rate, δ_{ij} link utilisation $u_{ij} = \frac{b_{ij}^{GFBR}}{c_{ij}}$, and ζ_i – the share of flow types traversing the edge (one value per QoS class). The nodes' features $X = \{x_i | v_i \in V \in \mathbb{R}^{|\mathbb{V}| \times m}\}$ correspond to edge features (cf. Equations (3-1)-(3-3)), except for delay and packet error rate (link measurements often includes links and connecting switches egress/ingress

ports, hence, delay/error rate induced by nodes is included in edge features [31]). Edges incident to v_i are denoted with $E(v_i)$: $E(v_i) = \{e \in E : \exists v_j \in V, e = (v_i, v_j) \in E\}$.

$$c_{v_i} = \sum_{e_{ij} \in E(v_i)} c_{ij} \quad (3-1)$$

$$b_{v_i}^{GFBR} = \sum_{e_{ij} \in E(v_i)} b_{ij}^{GFBR} \quad (3-2)$$

$$b_{v_i}^{MFBR} = \sum_{e_{ij} \in E(v_i)} b_{ij}^{MFBR} \quad (3-3)$$

Overlay metrics used by M-SDNC/M-QR are denoted with a superscript ψ and are calculated by SDNCs using edge metrics. The process involves only the domain's BNs $V^B \subset V$ so that M-SDNC sees the viable allocations between domain gateways. The bandwidth-related metrics between a pair of BNs, $v_i, v_j \in V^B$, are calculated using maximum flow (Push-Relabel) and Dijkstra algorithms, denoted as *maxflow* and *SP* [32]. These include:

- Total capacity c_{ij}^ψ between v_i and v_j (no network traffic):

$$c_{ij}^\psi = \text{maxflow}(v_i, v_j) \quad (3-4)$$

- Aggregate of GFBRs allocated to flows $b_{ij}^{\psi, GFBR}$:

$$b_{ij}^{\psi, GFBR} = c_{ij}^\psi - \text{maxflow}(v_i, v_j) \quad (3-5)$$

- Peak aggregated bandwidth (aggregate of MFBRs) $b_{ij}^{\psi, MFBR}$ that can be consumed by allocated flows:

$$b_{ij}^{\psi, MFBR} = \sum_{(u,w) \in SP(v_i, v_j)} b_{uw}^{MFBR} \quad (3-6)$$

- Utilisation of overlay link u_{ij}^ψ between v_i and v_j :

$$u_{ij}^\psi = (c_{ij}^\psi - b_{ij}^{\psi, GFBR}) / c_{ij}^\psi \quad (3-7)$$

- Delay and packet error rate, estimated by aggregation/multiplication over shortest path:

$$l_{ij}^\psi = \sum_{(u,w) \in SP(v_i, v_j)} l_{uw} \quad (3-8)$$

$$\delta_{ij}^\psi = 1 - \prod_{(u,w) \in SP(v_i, v_j)} (1 - \delta_{uw}) \quad (3-9)$$

Each flow is modelled by a tuple $f = (v_{src}, v_{dst}, t_0, t_d, q)$, where v_{src} - source, v_{dst} - sink, t_0 - arrival time, t_d - duration, and q - an SDN-level QoS class. We follow a simplified 3GPP 5QI definition: q_i is modelled as a tuple $q_i = (b^{GFBR}, b^{MFBR}, l^{max}, \delta^{max})$, where q_i - a QoS class identifier (common across SDN domains/strata), b^{GFBR} - GFBR (bandwidth reserved by SDNC/M-SDNC), b^{MFBR} - MFBR, l^{max} - PDB (cf. [30], clause 5.7.3.4), and δ^{max} - PER (cf. [30], clause 5.7.3.5). Each flow is considered unidirectional, so allocation of two-way exchange requires different q_i values for UL and DL flows. Each flow allocation can result in either acceptance or rejection $f_d \in \{1, 0\}$.

3.3.4.2 DRL problem setup

Hierarchical multi-controller SDN is treated as a combination of standard RL settings - a set of stochastic environments, each modelled as MDP $M = (S; A; T; R)$, where S – states, A – actions, T – transition function, and R – reward function.

State: environment state is composed of network information pulled by SDN controllers, data provided by GIMMF and metrics calculated internally by the router (i.e., state enrichment phase). Additional features include the edge/node expiry $\xi_{ij}, \xi_i \in \{0,1\}$ (probability of node/edge activity in the defined time horizon: 0 for active, 1 for inactive), $\chi(v) \in N$ (node type, e.g., satellite, gateway, relay, etc.), and $\deg(v_i)$ (node centrality). The obtained graph constitutes state $s_t^d = (G^d, X^d, F^d)$ in the DQR and $s_t^\psi = (G^\psi, X^\psi, F^\psi)$ in the M-QR case, where X, F are node and edge features:

$$\begin{aligned} F^d &= \{f_{ij}^d = (c_{ij}, l_{ij}, \delta_{ij}, b_{ij}^{GFBR}, b_{ij}^{MFBR}, u_{ij}, \zeta_{ij}, \xi_{ij}) | (i, j) \in E^d\} \\ X^d &= \{x_i^d = (c_i, b_i^{GFBR}, b_i^{MFBR}, u_i, \xi_i, \deg(v_i), \chi(v_i)) | v_i \in V^d\} \\ F^\psi &= \{f_{ij}^\psi = (c_{ij}^\psi, l_{ij}^\psi, \delta_{ij}^\psi, b_{ij}^{\psi,GFBR}, b_{ij}^{\psi,MFBR}, u_{ij}^\psi, \zeta_{ij}, \xi_{ij}) | (i, j) \in E^\psi\} \\ X^\psi &= \{x_i^\psi = (c_i, b_i^{GFBR}, b_i^{MFBR}, u_i, \xi_i, \deg(v_i), \chi(v_i)) | v_i \in V^\psi\} \end{aligned} \quad (3-10)$$

Before feeding state into DGA, a virtual path allocation of the flow for each of the candidate paths is performed. To improve DGA reasoning input state is extended by flags indicating if the node/edge is a part of the candidate path – $v^P, e^P \in \{0,1\}$. The final state evaluated by DGA for k -th allocation is defined as $s_t^{d,k} = (G^d, X^{d,k}, F^{d,k})$ for domain graphs, and $s_t^{\psi,k} = (G^\psi, X^{\psi,k}, F^{\psi,k})$ for the overlay graph:

$$\begin{aligned} F^{d,k} &= \{f_{ij}^{d,k} = (c_{ij}, d_{ij}, \delta_{ij}, b_{ij}^{k,GFBR}, b_{ij}^{k,MFBR}, u_{ij}^k, \zeta_{ij}, \xi_{ij}, e_{ij}^P) | (i, j) \in E^d\} \\ X^{d,k} &= \{x_i^{d,k} = (c_i, b_i^{k,GFBR}, b_i^{k,MFBR}, u_i^k, \xi_i, \deg(v_i), v^P, \chi(v_i)) | v_i \in V^d\} \\ F^{\psi,k} &= \{f_{ij}^{\psi,k} = (c_{ij}^\psi, d_{ij}^\psi, \delta_{ij}^\psi, b_{ij}^{\psi,k,GFBR}, b_{ij}^{\psi,k,MFBR}, u_{ij}^{\psi,k}, \zeta_{ij}, \xi_{ij}, e_{ij}^P) | (i, j) \in E^\psi\} \\ X^{\psi,k} &= \{x_i^{\psi,k} = (c_i, b_i^{k,GFBR}, b_i^{k,MFBR}, u_i^k, \xi_i, \deg(v_i), v^P, \chi(v_i)) | v_i \in V^\psi\} \end{aligned} \quad (3-11)$$

Action: For each incoming flow, DGA evaluates k candidate paths and selects the one maximising reward. Action a_t is a path calculated for the flow: IDP is denoted as a_t^d and E2EP as a_t^ψ . For each flow request, k candidate paths denoted as a_t^k are considered.

Transition: transition function defines a probability of moving from state s_t to s_{t+1} given the action a_t , $T(s_{t+1}|s_t, a_t)$.

Reward: The reward is given to DGA for each computed path sent to the SDNC for allocation. The reward functions for local and overlay agents are shown below:

$$r_t = \begin{cases} 1 - \text{std}(u) + \text{std}\left(\frac{b^{GFBR}}{b^{MFBR}}\right) + RC + QF; & f_d = 1 \\ - QF; & f_d = 0 \end{cases} \quad (3-12)$$

$$r_t^\psi = \begin{cases} 1 - \text{std}(u^\psi) + RC + QF; & f_d = 1 \\ - QF; & f_d = 0 \end{cases} \quad (3-13)$$

While the primary goal of 3DQR is to provide QoS paths, the method considers load balancing (at domain and overlay levels), QoS class priorities, and SDN CP overhead. Hence, the reward formula includes terms that guide the agents to:

- Distribute path allocations across links to maximise overall throughput - standard deviation of the utilisation of links $std(u)$, and overlay links $std(u^\psi)$;
- Punish often rerouting to save CP resources by the heuristic Rerouting Cost (RC),

$$RC = \begin{cases} \omega: \omega \in (-1,0); & \text{if flow is rerouted} \\ 0; & \text{otherwise} \end{cases} \quad (3-14)$$

- Prioritise traffic and scale the punishments for allocation failures; achieved by the QoS Factor (QF) heuristic, which maps QoS classes q_{id} to arbitrary priorities:

$$QF: q_{id} \mapsto QF(q_{id}) \in (0,1) \quad (3-15)$$

- Maximise bandwidth available for flows - high values of GFBR/MFBR indicate excess bandwidth that can be shared by flows (if GFBR < link capacity).

3.3.4.3 E2E routing

3DQR is based on Double Deep Q-Network (DDQN) [33], model-free and off-policy DRL algorithm, which goal is to approximate optimal action-value function $Q^*: State \times Action \rightarrow R$ by using a Deep Neural Network (DNN) architectures. The approximation can later be used to construct the policy maximising the acquired rewards: $\pi^*(s): argmax Q^*(s, a)$. DDQN agent comprises Local network (calculating Q-values for actions a_t based on state s_t); Target network (for stabilised learning), and Replay Buffer (for storing past experiences). DDQN is robust to overestimation bias [33] which is critical to obtain a high-quality generalised policy.

Each 3DQR algorithm step begins with fetching an allocation request from a buffering queue q_f , instantiated in M-SDNC. M-QR calculates E2EP composed of BNs – a_t^ψ , using the overlay state $s_t^\psi = (G^\psi, X^\psi, F^\psi)$. M-SDNC verifies E2EP feasibility considering flows' QoS requirements and the limitations of the network infrastructure:

$$\begin{aligned} b_{q_i}^{GFBR} + b_{ij}^{\psi, GFBR} &\geq c_{ij}^\psi, \forall i, j \in a_t^\psi \\ \sum_{(i,j) \in a_t^\psi} l_{ij}^\psi &\leq l_{q_i}^{max} \\ 1 - \prod_{(i,j) \in a_t^\psi} (1 - \delta_{ij}^\psi) &\leq \delta_{q_i}^{max} \end{aligned} \quad (3-16)$$

On verification failure, M-SDNC delivers the rejection information $f_d = 0$ to the requester (ExtReq, cf. Figure 3-12). Otherwise, M-SDNC splits E2EP into segments a_t^{seg} composed of BN pairs $(v_{src}, v_{dst}) \in a_t^{seg}$, and sends IDP establishment requests containing allocation parameters $(v_{src}, v_{dst}, f_{q_i})$ to appropriate DQRs (selection is based on the BNs' domain membership). Each IDP provided by DQR – a_t^d , undergoes verification by respective SDNC:

$$\begin{aligned} b_{q_i}^{GFBR} + b_{ij}^{GFBR} &\geq c_{ij}; \forall (i, j) \in a_t^d \\ \sum_{(i,j) \in a_t^d} l_{ij} &\leq l_{q_i}^{max} \\ 1 - \prod_{(i,j) \in a_t^d} (1 - \delta_{ij}) &\leq \delta_{q_i}^{max} \end{aligned} \quad (3-17)$$

The two-level verification mitigates potential discrepancies between the overlay approximate view and actual domain states and speeds up flow rejection when no QoS-meeting path exists. Allocation failures incur penalties only if E2EP exists. Otherwise, DGA's policy remains unaffected. Moreover, if M-QR selects a path that does not meet the QoS constraints, a penalty is given to both M-QR and involved DQRs. Once all IDPs are pre-allocated by SDNCs, M-SDNC performs the final verification by expanding E2EP ($a_t^{\psi,exp}$) - substituting path segments with pre-allocated IDPs (actual domain paths a_t^d), and then checking the conditions:

$$\sum_{(i,j) \in a_t^{\psi,exp}} l_{ij} \leq l_{q_i}^{max} \quad (3-18)$$

$$1 - \prod_{(i,j) \in a_t^{\psi,exp}} (1 - \delta_{ij}) \leq \delta_{q_i} \quad (3-19)$$

Finally, if E2EP satisfies QoS requirements, the flow is admitted into the network, and the result is delivered to the path requester. Transition reward is obtained for each domain and stored in Replay Buffers H (TED) for training of the involved DGAs (using the Bellman equation). The fundamental part of the 3DQR lies in how M-QR/DQR calculate paths. The internals of DQR/M-QR and data flow are shown in Figure 3-13.

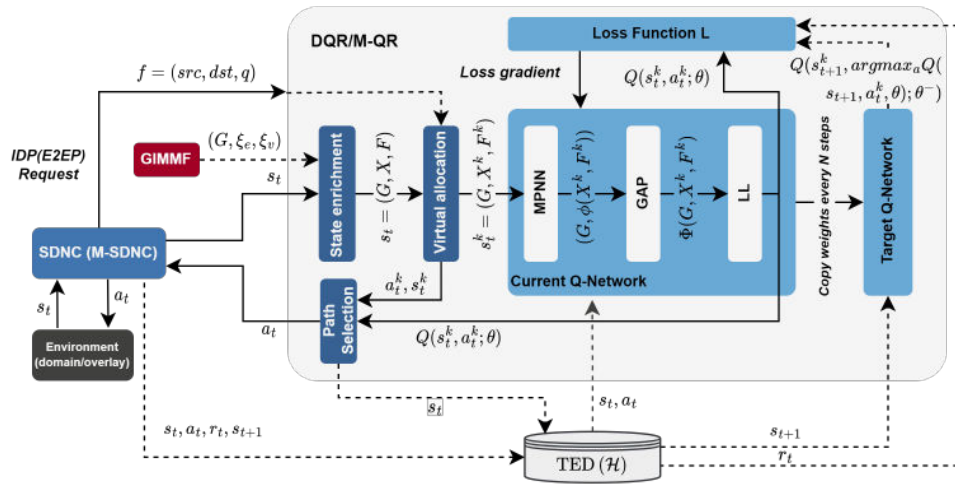


Figure 3-13: DQR/M-QR and DGA internals (Current, Target Q-Networks and Loss Function).

Every IDP/E2EP request includes key flow parameters: source src , destination dst , and QoS class. First, the router (DQR/M-QR) acquires recent network state s_t and extends it with additional features (cf. Section 3.3.4.2), including the information on persistent NTN graphs delivered by GIMMF. Then, the router calculates k candidate paths for src - dst pair with Dijkstra SP algorithm using delay. For each path, virtual path allocation is performed, which involves updating links parameters' (GFBR, MFBR, utilisation, cf. eq. (3-11)) as if path allocation succeeded. The obtained state for path k , s_t^k , is later fed to the DGA Current Q-Network to obtain the single Q-value $qval = Q(s_t^k, a_t^k; \theta)$ of the allocation. The tuples of action a_t^k , state s_t^k , and $qval_t^k$ are buffered for comparison with other virtual path allocations. The final path a_t is selected based on the highest Q-value among the paths and delivered to SDNC/M-SDNC. If no path is available, the router stores the flow metadata to exclude the respective samples from the DGA training process. After the allocation is performed, the relevant experiences in TED are updated using features added by the router (expiry, etc.).

Both Current and Target Q-Networks embed the same GNN architecture that enables time-variant and size-variant graph input and comprises following layers:

- Message Passing Neural Network (MPNN) [34] - used to obtain node embeddings ϕ using edge and node features. Comprises: message passing (eq. 20) and readout phases, where M_t - a message function, U_t - vertex update function, h_v^t - hidden state, m_v^{t+1} - message, and t - passing step.

$$\begin{aligned} m_v^{t+1} &= \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \\ h_v^{t+1} &= U_t(h_v^t, m_v^{t+1}) \end{aligned} \quad (3-20)$$

Both message M_t and vertex update functions U_t are implemented using concatenation, MLP with ReLu activation, and sum as the aggregation operator:

$$M_t = MLP(h_v || h_w || h_{vw}); w \in N(v) \quad (3-21)$$

$$U_t = MLP(h_v^t || m_v^{t+1}) \quad (3-22)$$

- Global Attention Pooling (GAP) [35] for node embeddings aggregation using attention mechanisms, obtaining attention scores and calculating graph embedding Φ . GAP plays the role of MPNN's readout phase.
- Linear Layer (LL) – compression of graph embedding into a singular output (Q-value).

The rationale of the above architecture is as follows. MPNN enables aggregating node/edge information based on the neighbourhood and inter-node relationships and obtaining a more accurate representation of the network state (compared to DNN architectures). The role of GAP is to pick nodes that contribute the most to the graph embedding values – the traffic concentration points. Hence, DGA can learn about the relative relationships of nodes/edges and focus on the nodes/edges that accommodate the highest traffic volume and are at risk of congestion. Combined with reward function and DDQN, the agents are encouraged to prioritise the allocations that omit the most loaded edges and nodes, if possible, and load balance across traffic concentration points. Time is introduced into the model via node/edge expiry predictions within the environment state. While GNNS are often paired with time-series prediction units like LSTM, 3DQR uses a more generic approach. Although LEO satellite orbital periods are constant, visible topologies vary due to Earth's rotation and orbital inclinations, which makes training time selection critical due to the risk of performance drops. Moreover, coupling reasoning with time omits random events, which GIMMF-like functions can predict. 3DQR capitalises on GIMMF topology prediction, enabling broader applicability across 3D systems.

3.3.5 Evaluation approach & results

3DQR was evaluated for scenarios being in line with ETHER UC3, i.e., concerning services essential for the aviation ecosystem. The evaluation was done with a proprietary event-driven simulator developed by OPL (as a part of WP4 activities) using Python 3.10 with libraries: SimPy 4.1.1 (SDN framework), PyTorch 2.5.0, and PyTorch Geometric 2.6.1 (DRL agents), Skyfield 1.48 (GIMMF). To model satellite topologies, publicly available Two-Line Element (TLE) files for Starlink system were used [36]. As shown in Table 3-1, the 3DQR tests covered:

- **Performance gains vs. baseline methods:** (i) SP routing using link delay (further referred to as H-SP); (ii) SP for routing for overlay network and NTN, DDQN for TNs (DNN-based); (iii) Uncoordinated 3DQR (3DQR-U) - domain DGAs and SP routing at the overlay level;
- **Transfer capabilities:** performance tests of DGAs trained in one topology and operation in previously unseen topologies with different topological properties.

Table 3-1: Evaluation scenarios.

Test ID	Topology	Algorithm (TN, NTN, TN-NTN)	Scope
s0-a	T1	H-SP	Performance
s0-b		DDQN	
s0-c		3DQR	
s0-d		3DQR-U	
s1-a	T2	H-SP	
s1-b		DDQN	
s1-c		3DQR	
s1-d		3DQR-U	
s2-a	T1-T32	H-SP	Transfer
s2-b		3DQR (s0-c)	
s2-c		3DQR (s1-c)	

Tests involved both sparse and dense networks with TN nodes across Western and Central Europe. Training and evaluation (s0-x, s1-x) used two TN topologies (32 nodes: 50% relays, 25% gateways, 25% gNodeBs) with average node degrees of 4 (T1) and 7 (T2), and two NTN topologies - Starlink subsets with 32 (T1) and 24 nodes (T2), each with degree 4. Transfer capabilities were tested on 30 unseen topologies (16–80 TN/NTN nodes, average degrees 3–6). Feeder Links (FLs) were set dynamically based on LEO proximity to TN gateways and a visibility threshold (elevation $\geq 30^\circ$). Domain links and FLs had capacities of 100 and 200 kilounits, respectively, and $\delta = 10^{-8}$. Delays were based on spatial distance. UE–LEO links assumed a worst-case 13 ms delay (600 km orbit, 3GPP [30]) and $\delta = 10^{-7}$; UE–TN links assumed 2 ms RAN delay and $\delta = 10^{-8}$. QoS classes of arriving flows reflect NTN services (cf. Table 3-2). Flow durations ranged from 0.5 to 20 minutes (uniform distribution) with equal QoS class share (25% each) and type: 1/3 inter-domain and 2/3 intra-domain flows (1/3 for TN and NTN). GIMMF aggregated topology over 5 s and predicted 5 s ahead.

Table 3-2: Considered services (based on [30]).

5QI	PDB [ms]	PER	GFBR	MFBR	QF	Example service
1	100	10 ⁻²	75	150	0.3	Conversational voice
2	150	10 ⁻³	2000	5000	0.5	Conversational video
4	300	10 ⁻⁶	1000	2000	0.8	Non-conversational video (buffered streaming)
75	50	10 ⁻²	500	1000	0.9	A2X messages, aircraft telemetry

3.3.5.1 Performance

Performance was evaluated, by considering two metrics: **flow rejection rate** (the proportion of failed flow allocation requests) and **standard deviation of link utilisation**, denoted as

$std(u_{ij})$, which reflects the overall traffic imbalance across all network links at a given time step. For each test, the agents were trained in low-traffic environments for 200 episodes lasting 0.5 h each (500 flow requests per episode – $N_{flows} = 500$). The models that accumulated the highest aggregate episodic reward were chosen for comparison. The respective training curves for the 3DQR model are shown in Figure 3-14.

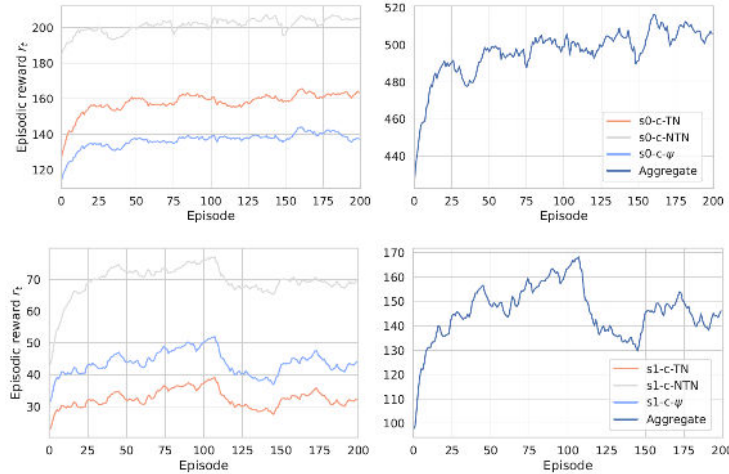


Figure 3-14: Episodic reward obtained by 3DQR model in low-traffic environment: for domain agents for network topology T1 (up) and T2 (down).

In both scenarios, the model reaches satisfactory rewards after around 100 training episodes. In the s1-c scenario, a temporary performance drop – likely due to overtraining in the smaller NTN state space—is observed but recovers after about 40 episodes. Performance differences across domains stem from variations in network topology, flow rerouting frequency, and flow types (inter-/intra-domain, QoS class). Overall, good training stability can be observed.

Best-performing models were selected for further testing under increasing network loads, up to $N_{flows} = 3000$, representing high congestion. As shown in Figure 3-14, 3DQR significantly reduces the flow rejection rate compared to H-SP routing for medium to high loads ($N_{flows} \geq 1000$), with improvements reaching up to 13.5%. This is due to its ability to jointly optimise intra- and inter-domain routing, efficiently offloading traffic and balancing link utilisation. At low traffic levels, a slight performance drop (<2%) occurs due to longer candidate paths increasing delay and packet error rates (could be mitigated by limiting the number of candidate paths). Figure 3-15 also shows that DDQN with H-SP and 3DQR-U underperform – DDQN struggles to adapt to dynamic conditions (raising rejection by up to 10%), while 3DQR-U matches H-SP in rejection rate but improves load balance. Overall, 3DQR achieves stable performance across loads with minimal added complexity, highlighting the value of overlay-level traffic engineering. Its use of MPNN and GAP enables effective generalisation by extracting and scaling key network features, thus addressing a key limitation of SotA DRL approaches.

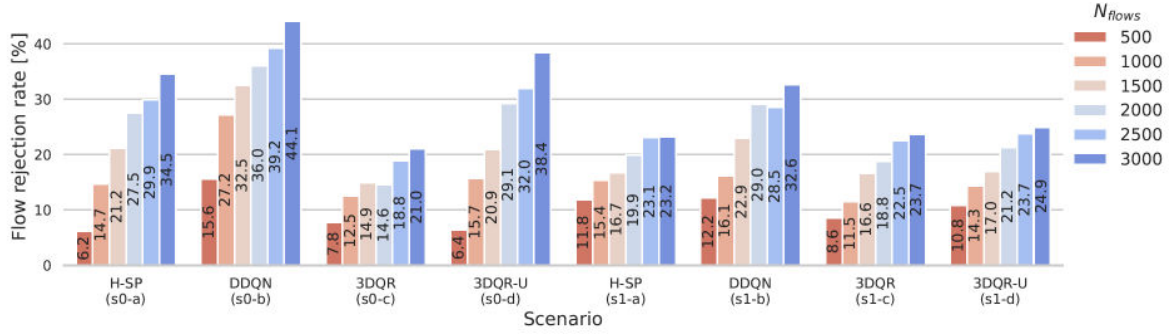


Figure 3-15: Flow rejection rate per scenario.

The second goal of 3DQR is to prioritise most important QoS classes. Figure 3-16 shows that 3DQR reduces rejection rate for critical aviation class 75 - by up to 4.5% under heavy load (at the expense of class 4). As class 4 demands highest bandwidth, it is more prone to rejection in high traffic scenarios like s0-c with $N_{flows} = 2000$. Rejection rates for other classes remain low and rise gradually with traffic volume.

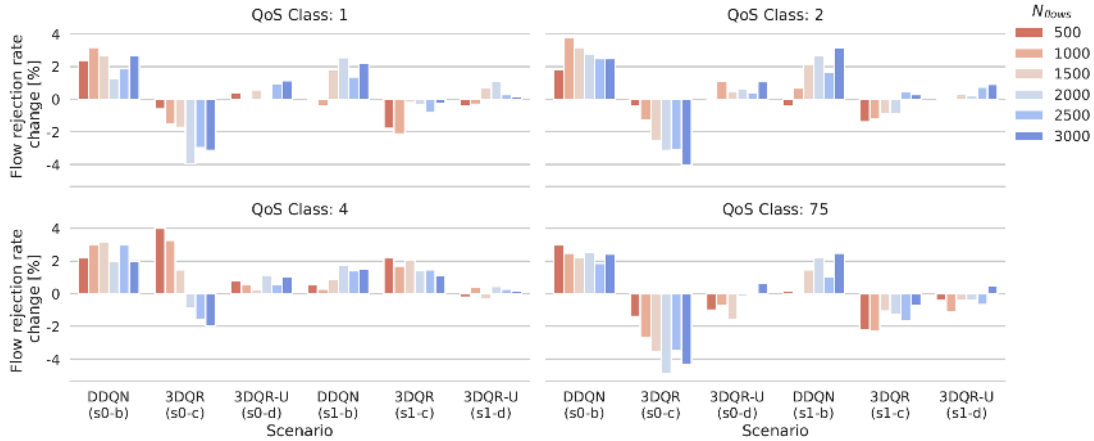


Figure 3-16: Flow rejection rate improvement per QoS class compared to baseline H-SP routing.

The second key objective of 3DQR is load balancing and improving resource utilisation, particularly across ISLs and FLs. As shown in Figure 3-17, 3DQR significantly reduces the mean $std(u_{ij})$, indicating more efficient flow distribution: in sparse networks (s0-c), it achieves up to **52% better load balance** than H-SP. In dense networks (s1-c), improvements are smaller due to overlapping shortest paths (can be improved by node-disjoint routing for candidate paths). 3DQR-U shows modest, consistent gains (4–6%) in sparse networks, but performs similarly to H-SP in dense ones. DDQN suffers from convergence issues and yields poor results. Overall, 3DQR consistently shows lower 25th and 75th percentile $std(u_{ij})$ values, indicating more stable performance across different traffic loads.

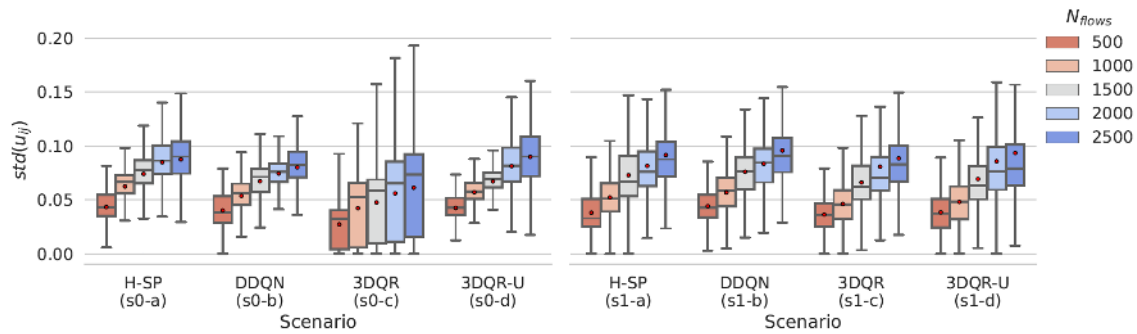


Figure 3-17: Standard deviation of utilisation $std(u_{ij})$ of links (mean marked with red dot).

3.3.5.2 Transfer capabilities

To evaluate 3DQR's generalisation and reasoning in unseen topologies, top-performing models trained on T1 and T2 (see Section 3.3.5) were tested on 30+ new topologies, ranging from 16–80 NTN nodes, 32–80 TN nodes, and average node degrees of 3–6. T1 and T2 were also included to benchmark native versus transferred performance. Each test involved 1500 flow requests using the same setup. Results are shown in Figure 3-18.

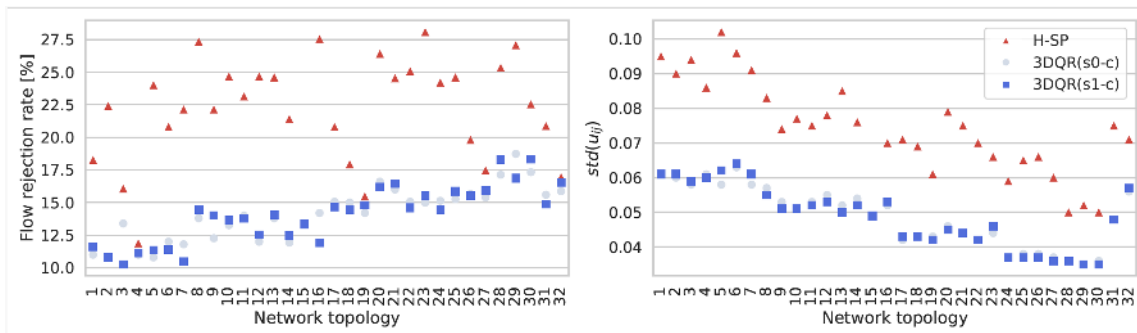


Figure 3-18: Performance comparison of H-SP-routing and 3DQR model in terms of flow rejection rate.

The trained 3DQR models performed strongly across most scenarios, reducing flow rejection by up to 14% and improving load balance by up to 50%. Only four highly sparse topologies (low node degree and few TN/NTN nodes) showed similar rejection rates to H-SP due to limited routing options, though load distribution gains remained. Both models also performed similarly in their native topologies (T1 and T2), confirming strong generalisation. This consistency stems from the agent's attention mechanism, which effectively scales edge and node importance to avoid congested links.

Overall, 3DQR can be used to optimise routing in 3D networks, while prioritising key aviation services. The transfer capabilities and ability to operate for different-sized networks are also of key importance regarding its scaling to optimise carrier-grade integrated networks.

3.4 REINFORCEMENT LEARNING- BASED DATA COMPRESSION AND ROUTING DESIGN FOR MULTI-SATELLITE SYSTEMS

3.4.1 Background

LEO satellite networks (LEONet) are increasingly vital for next-generation wireless communications, offering lower latency and reduced transmission power than geostationary (GEO) and medium Earth orbit (MEO) systems [37]. Enabled by advancements in hardware, chip design, and launch technologies, modern LEO networks comprise large-scale, multi-orbit constellations, with hundreds to thousands of satellites collaborating for seamless global coverage. However, satellite mobility and limited visibility windows introduce discontinuities in end-to-end (E2E) communication. Efficient routing and scheduling are thus essential to optimize network performance, ensure reliable data delivery, and minimize latency in this dynamic orbital environment. Recent studies on multi-hop satellite routing [38], [39] focus on the optimization coverage probability and latency. [38] proposes a minimum-hop count design, while [39] introduces a real-time position-based key node graph. However, these frameworks do not account for dynamic traffic arrival rates, limiting their applicability to real-time scenarios. The increasing data demand from satellite applications can cause transmission delays, degrading QoS. Advanced data compression (DC) reduces data volume, alleviating congestion and improving latency [40], but adds computational overhead, complicating network management [41].

Deep reinforcement learning (DRL) has emerged as a powerful tool for optimising decision-making in dynamic wire-less environments, including multi-orbit satellite routing [42], [43], [44] [45]. In [42], a multi-agent duelling double deep Q network (D3QN)-based approach enables distributed, trustworthy load-balancing routing by minimising queue utilisation and path delay. In [43] Q-learning (QL)-based routing schemes in [44] focus on reducing packet drop rates and E2E latency, respectively. [46] introduces a DQN-based decentralised routing method that selects optimal routing nodes based on spatial position, queuing delay, and bandwidth availability. A constrained multi-agent DRL framework in [47] ensures latency and energy efficiency in integrated satellite-terrestrial networks, while [45] explores continual DRL with offline pre-training and online learning. However, none of these works incorporate data compression (DC), which could further enhance routing performance.

3.4.2 Motivation and Contribution

This work investigates a joint DC and routing design approach for multi-orbit satellite networks to minimise transmission delays. For the first time, a MADRL-based distributed approach for joint DC and routing (JCR) is proposed, taking into account constraints related to maximum hop count and buffer length. This novel framework aims to optimise the trade-offs between DC and routing efficiency, ultimately reducing the transmission delay in LEONet.

3.4.3 System Model

As shown in Figure 3-19, a LEO satellite (LEOSat) network is considered including M orbital planes, and each has N satellites. One denotes the set of LEOSats as $U = \{u_1, u_2, \dots, u_S\}$ where $S = MN$ denotes the total number of LEOSats. Furthermore, the set of ISLs are represented as V and v_{ij} denotes the ISL between LEOSats u_i and u_j . In this setting, each LEOSat can connect to at most four adjacent LEOSats, including two intra-plane LEOSats (the upper and lower ones at the same orbital plane), and two inter-plane LEOSats belonging to the left and right adjacent orbital planes. A scenario where a source LEOSat u_j needs to transmit K information bits to a ground gateway (GW) is considered.

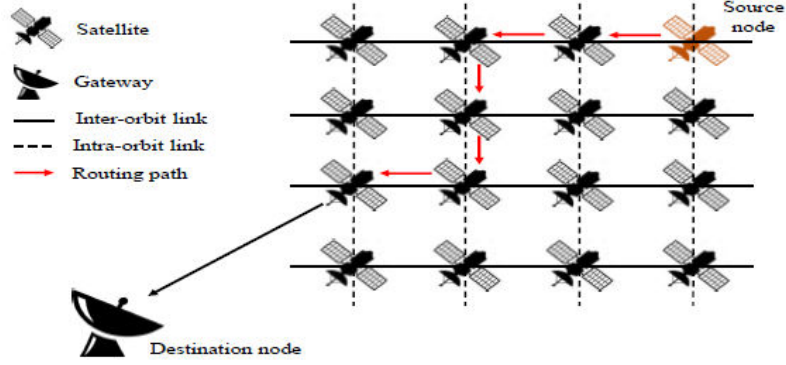


Figure 3-19: Illustration of a multi-orbit LEO satellite network architecture.

A. Communication Model

For ISLs, the achievable transmission rate from LEOSat u_j to u_i is given by

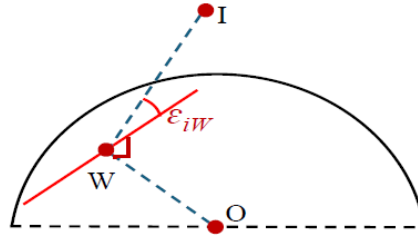


Figure 3-20: Elevation angle definition.

$$R_{ij}(t) = B \log_2 \{1 + P_i G_i G_j / [\theta_{ij}(t) F B N_0]\}, \quad (3-23)$$

where B denotes the channel bandwidth; P_i and G_i represents the transmission power and antenna gain of LEOSat u_i , respectively; G_j is the antenna gain of LEOSat u_j ; F is the noise figure; N_0 is the noise power spectral density; and $\theta_{ij}(t)$ stands for the free-space path loss which is expressed as

$$\theta_{ij}(t) = [4\pi d_{ij}(t) f_c / c]^2, \quad (3-24)$$

where $d_{ij}(t)$ represents the distance from LEOSat u_i to LEOSat u_j , f_c is the ISL carrier frequency and c denotes the light speed.

In addition, the distance $d_{ij}(t)$ is determined as

$$d_{ij}(t) = \hat{r} \sqrt{2(1 - \cos \mu_i \cos \mu_j - \sin \mu_i \sin \mu_j \cos v_i - v_j)}, \quad (3-25)$$

where $(\hat{r}, \mu_j, v_j)$ denotes the sphere coordinate of LEOSat u_i , $\hat{r} = \hat{R} + \hat{h}_i$, $\hat{R}$ is the earth radius, $\hat{h}_i$ indicates the LEOSat altitude, μ_i denotes the polar angle, and v_j shows the azimuth angle. Note that the position of LEOSat u_i is updated over time along its orbit according to its velocity.

For feeder link (FL), the channel coefficient from a LEOSat u_i to the GW in time-slot (TS) t is modelled as [48]:

$$h_{iW}(t) = \sqrt{G_i G_W} LA, \quad (3-26)$$

where G_W is the GW antenna gain; L is the Line of Sight (LOS) free-space propagation, expressed as $L = \ell \exp\{-j2\pi f_c d_{iW}(t)/c\}$ where $\ell = c/4\pi f_c G_W d_{iW}(t)$ denotes the amplitude loss; f_c^{GW} is the FL carrier frequency and $A = |\alpha|e^{-j\xi}$ denotes the atmospheric impairments experienced at the GW antenna, such as cloud, rain, and fog attenuation, where $\alpha \in (0, 1]$ and amplitude $\xi \in [-\pi, \pi]$ are the fading and phase shift, respectively. The achievable rate of the link from LEOSat u_j to the GW is thus expressed as

$$R_{iW}(t) = B \log_2(1 + P_i |h_{iW}(t)|^2 / (FBN_0)) \quad (3-27)$$

Note that the FL transmission can be established when the elevation angle (i.e., the angle between the LOS to the LEOSat and the horizon) from the GW's location, ε_{iW} , exceeds a predefined minimum threshold, $\varepsilon_{\min}$. From (3-28), the elevation angle in radian is calculated as

$$\varepsilon_{iW} = \cos^{-1}(WI \cdot WO / (|WI||WO|)) - \pi/2, \quad (3-28)$$

where **WI** and **WO** denote the vectors from the GW to LEOSat u_i and to the earth center O , respectively.

B. Data Compression Process

Since the capacity of wireless links is limited, DC is applied at the source LEOSat u_s to reduce the amount of transmitted data. The DC can be accomplished either through the elimination of solely statistical redundancy, known as lossless DC, or by discarding unnecessary information, referred to as lossy DC. Meanwhile, the intermediate LEOSats operate as the relays to forward the data to the GW. The data decompression (DD) is then performed at the GW to achieve the original data. Following [41], the DC computational load and DD computational load can be modelled as non-linear functions of the compression ratio (CR), which are given by

$$\kappa_x = \eta_0 [\eta_{x,1} (\beta_s)^{\eta_{x,2}} + \eta_{x,3}], \quad (3-29)$$

where $x \in \{\text{co}, \text{de}\}$ represents DC ($x = \text{co}$) and DD ($x = \text{de}$), η_0 stands for the maximum CPU-cycle number, $\eta_{x,1}$, $k \in \{1; 2; 3\}$, are constants [41], $\beta_s = K/K_{\text{co}} \in \{\beta_{\min}, \beta_{\max}\}$ is the CR, K is the information-bit number, while $K_{\text{co}} < K$ is the number of bits after DC process. Based on (3-29), the computation time at the source LEOSat u_s for DC and at the GW for DD are, respectively, determined as

$$\tau_s^{\text{co}} = \kappa_{\text{co}} / r_s \quad \text{and} \quad \tau_W^{\text{de}} = \kappa_{\text{de}} / r_W, \quad (3-30)$$

where r_s and r_W are the CPU clock speed of the source LEOSat u_s and the GW, respectively.

C. Routing Process

After receiving the data of the source LEOSat u_s , each LEOSat determines the subsequent hop node based on specific routing policies. It then proceeds to forward the data to the selected next-hop LEOSat through the relevant ISL. This process is repeated at each subsequent LEOSat along the routing path until the data reaches a LEOSat which is connected to the ground GW. Note that the data is processed following a first-in-first-out (FIFO) policy. One assumes that a routing path P_p is established to accomplish an E2E communication from the source LEOSat u_s to the ground GW. In particular, the routing path P_p can be defined as

$$P_p = \{e_{u_s j_1}, e_{j_1 j_2}, \dots, e_{j_n W}\}, j_i \in U, i = 1:n, \quad (3-31)$$

Where n is the hop number of routing path P_p and $e_{j_{i-1} j_i} \in P_p$ expresses that node j_{i-1} after receiving the data of the source LEOSat u_s will transmit to node j_i . Note that after receiving the data of the source LEOSat u_s , LEOSat $j_i \in U$ has O_{ji} adjacent LEOSats, and it can transmit to only its adjacent LEOSats.

D. Delay Analysis

Assume the data arrival rate at node i in TS t is defined as $\lambda_i(t) = \hat{\lambda}_i(t) + K^{CO}$ (bits/s), where $\hat{\lambda}_i(t)$ follows Poisson distribution and K^{CO} is the amount of the target data. We consider the queuing delay to guarantee load balancing and congestion control. In particular, we assume that node i decides to forward data to its adjacent node j in TS t at the rate $R_{ij}(t)$ given in (3-23). Thus, the queuing delay is given by

$$T^q(t) = q_i(t)/R_{ij}(t) \quad (3-32)$$

where $q_i(t) = [q_i(t-1) + \lambda_i(t) - R_{ij}(t-1)]^+$ is the queue length, and $[a]^+ = \max\{a, 0\}$. To guarantee a maximum queue length q_{max} , we have the constraint $q_i(t) \leq q_{max}$. In addition, propagation delay, which is the time spent for electromagnetic radiation to travel from node i to node j , is represented as

$$\tau^p(t) = d_{ij}(t)/c. \quad (3-33)$$

Accordingly, the one-hop delay can be given by

$$\tau_{ij}(t) = \tau_{ij}^q(t) + \tau_{ij}^p(t). \quad (3-34)$$

The E2E delay for completing JCR data delivery includes the routing-based transmission due to P_p path, the computation time for the DC process at the source node and the one for the DD process at the destination node, which is given by

$$\tau_p = \tau^{co} + \tau^{de} + \tau_{P_p}, \quad (3-35)$$

where τ_{P_p} is the delay due to the routing path, which is expressed as $\tau_{\{P\}_{\{p\}}} = \sum_{v \in \{P\}_{\{p\}}} \{\tau_{\{v\}}(t)\}$.

E. Problem formulation

This work aims to minimise the E2E delay of the routing-based transmission process and computation execution for DC and DD performed at the source and destination nodes. Casting this design objective, the optimisation problem is defined as

$$\begin{aligned} \min_{P_p, \beta} \tau_p \quad \text{s.t.} \quad & D_p \leq D_{\max}, \\ & \theta_{\min} \leq \theta_s \leq \theta_{\max}, \forall s \\ & q_i(t) \leq q_{\max}, \forall i, \end{aligned} \quad (3-36)$$

where β is the DC policy. The constraint shows the Age of Information (AoI) constraint for a packet p , where $D_p = |P_p|$ denotes the routing hop number of P_p and $D_{\max}$ represents the maximum hop number corresponding to limit AoI.

3.4.4 Proposed Solution

A. Multi-Agent Deep Reinforcement Learning

In practical scenarios, collecting global information for a centralised routing strategy becomes costly. Additionally, the limited resources on LEOSats reduce data transmission efficiency, especially with users' growing demands for information exchange, leading to increased overall delay. To address this issue, this work develops for the first time a MADRL-based distributed approach for JCR to enhance transmission efficiency and overall network performance by minimising total delay. In particular, each LEOSat acts as an agent, learning to optimise its individual routing strategy, while the source LEOSat learns to determine the optimal policies for both DC and routing methods. The main elements of the DRL framework, i.e., state, action, and reward, are defined as follows.

1. **State:** The state of agent i at TS t is defined as the combination of information on the current queue lengths and FL propagation delays shared by its adjacent LEOSats,

$$s_i(t) = \{Q_i(t), \Lambda_i(t)\}, \quad (3-37)$$

where $Q_i(t)$ represents the queue lengths of adjacent LEOSats to LEOSat i ; and $\Lambda_i(t)$ indicates the estimated propagation delays from these LEOSats to the GW, derived based on their distances to the GW.

2. **Action:** There are two action types: (i) Selecting the next-hop LEOSat and (ii) Choosing the CR. To simplify the learning process, the CR at the source LEOSat is divided into a set of discrete levels. Specifically, it is partitioned into L levels, defined by $\beta_\ell = \beta_{\min} + \ell(\beta_{\min} + \beta_{\max})/(L - 1)$, where β_ℓ represents the ℓ -th CR level. Given this context, the action of agent i at TS t is determined as

$$a_i(t) \in A_i = \{1, \dots, l_o, \dots, LO_i\}, \quad (3-38)$$

where O_i is the number of adjacent LEOSats to LEOSat i . Note that L is included in the action space if LEOSat i is the source node only, otherwise $L = 1$.

3. *Reward*: With the objective of minimising the total delay, the reward function for agent i is defined as follows

$$r_i(t) = \begin{cases} \varphi_1/\tau(t) + \varphi_2/\hat{d}_i(t), & \text{if constraints are satisfied} \\ \Phi & \text{otherwise,} \end{cases} \quad (3-39)$$

where $\tau(t) = \tau^{\text{co}} + \tau^{\text{de}} + \tau_{ij}(t)$ if agent i is the source LEOSat, otherwise $\tau(t) = \tau_{ij}(t)$, $\hat{d}_i(t) = d_{a(t)W}/c$, $d_{a(t)W}$ denotes the spatial distance from the next-hop LEOSat selected by the action $a(t)$ to the GW, φ_1 and φ_2 are the weight factors, and Φ represents a penalty value.

B. Proposed MADRL-based JCR Method

To address problem (3-36), a multi-agent (MA) Double Deep Q-Network (DDQN) algorithm for the JCR strategy, namely MADDQN-JCR is applied. Specifically, each agent i builds its own DDQN model including two deep neural networks (DNNs), [29] namely online and target networks with the corresponding weight matrices θ_i and θ'_i , respectively. The online network is responsible for selecting actions, while the target network is utilised to assess those actions.

The action selection can be conducted by applying the ϵ -greedy policy. This involves selecting a random action with a probability of ϵ , while the best action with the highest Q-value is exploited with a probability of $1 - \epsilon$. Thus, the action $a_i(t)$ based on this policy can be expressed as

$$a_i(t) = \begin{cases} \text{random action, with prob. of } \epsilon, \\ \text{best action, with prob. of } 1 - \epsilon, \end{cases} \quad (3-40)$$

where $a_{\text{best}} = \arg \max_{a \in A} \{Q(s_i(t), a; \theta_i)\}$ and $Q(s_i(t), a_i(t); \theta_i)$ indicates the Q-value of the state-action pair $(s_i(t), a_i(t))$.

The goal of the DDQN algorithm is to minimise the loss function defined as the squared difference between the target Q-value and the Q-value achieved from the online network, which is formulated as [29]

$$L(\theta_i) = [y_i(t) - Q_i(s_i(t), a_i(t); \theta_i)]^2, \quad (3-41)$$

where $y_i(t)$ represents the target Q-value given by

$$y_i(t) = r_i(t) + \gamma Q_i(s_i(t+1), a_i(t+1); \theta'_i), \quad (3-42)$$

In which $\hat{Q}_i(t+1) = Q_i(s_i(t+1), a_i(t+1); \theta_i)$ and γ denotes the discount factor. The MADDQN-JCR algorithm, outlined in Algorithm 1, operates iteratively over multiple episodes to ensure convergence. At the first TS of episode i , the source LEOSat observes the environment, selects a CR and the next-hop LEOSat, then compresses and transmits the data. Upon successful transmission, it receives a reward as defined in (3-39). The selected LEOSat subsequently determines the next hop for data forwarding in the next TS. This process continues until the data reaches the GW or constraint (3-36) is violated.

Algorithm 1 MADDQN-JCR ALGORITHM

```

1: Initialize the weight matrices of the online and target networks.
2: Initialize source LEOSat  $x$  and routing path  $P = \{\emptyset\}$ .
3: for  $i = 1, \dots, E_p$  do
4:   Initialize the state for current agent  $z = x$ , i.e.,  $s_z(t)$ .
5:   for  $t = 1, \dots, D_{\max}$  do
6:     if  $\varepsilon_{zW} \leq \varepsilon_{\min}$  then
7:       if  $z = x$  then
8:         Agent  $z$  takes an action of
           jointly selecting a CR level,
           denoted by  $\theta_\ell$  ( $1 \leq \ell \leq L$ ),
           and next-hop LEOSat from
           its adjacent set, denoted by
            $j$ , based on (3-40).
9:       else
10:        Agent  $z$  selects next-hop LEOSat  $j$  from its adjacent LEOSats set using (3-40).
11:      end if
12:      Agent  $z$  observes the reward  $r_z(t)$  in (3-39) and moves to the next states  $s_z(t+1)$ .
13:      Store an experience tuple  $(s_z(t), a_z(t), r_z(t), s_z(t+1))$  to the memory of agent  $z$ .
14:      Add agent  $z$  to the routing path  $P$ .
15:    else
16:      Agent  $z$  transmits data to the GW.
17:      Break
18:    end if
19:    Update agent  $z = j$ .
20:  end for
21:  for all agents in  $P$  do
22:    Randomly sample a mini-batch of experiences from the memory to train the online network.
23:    Update  $\theta_z$  ( $z \in P$ ) by using gradient descent to minimize the loss function.

```

3.4.5 Simulation Results

This section presents the simulation results to assess the system performance in terms of reward and total delay. The DDQN model includes three fully-connected hidden layers with 256, 128, and 64 neurons. Each LEOSat i has a random data arrival rate $\hat{\lambda}_i \leq \lambda_{\max}$, where $\lambda_{\max} = 15$ KB/s. The data set of queue length is generated based on the above assumption. For atmospheric impairments, its amplitude fading α varies uniformly over time within the range of $[-5, 0]$ dB. The remaining experimental parameters are listed in Table 3-3. In the following figures, we provide the outcomes of our proposed algorithm and the conventional MADRL-based routing method without DC (WoC) for comparative purposes. In the WoC method, the routing path is determined by using the MADDQN method with the original data size.

Table 3-3: Simulation parameters.

Parameters	Value
Number of orbital planes (M)	6
Number of LEOSats in each orbital plane (N)	11
Earth radius (R^e)	6378 km
LEOSat altitude ($\hat{h}_i$)	780 km
LEOSat speed	7.8 km/s
Transmission power (P)	10 W
Channel bandwidth (B)	180 kHz
Antenna gain (G_i and G_W)	37.1 and 32.8 dBi
Noise figure (F)	6 dB
Noise power spectral density (N_0)	-174 dBm/Hz
ISL and FL carrier frequencies (f_c & f^{GW})	26 & 20 GHz
Light speed (c)	3×10^8 m/s
Min. & max. CR (β)	3.4 & 11.2
Number of CR levels (L)	6
Number of information bits (K)	6 KB
Minimum elevation angle ($\varepsilon_{\min}$)	8.2°
Number of layers	3

Number of neurons per hidden layers	{256, 128, 64}
Learning rate	0.001
Discount factor	0.9
Optimizer	Adam

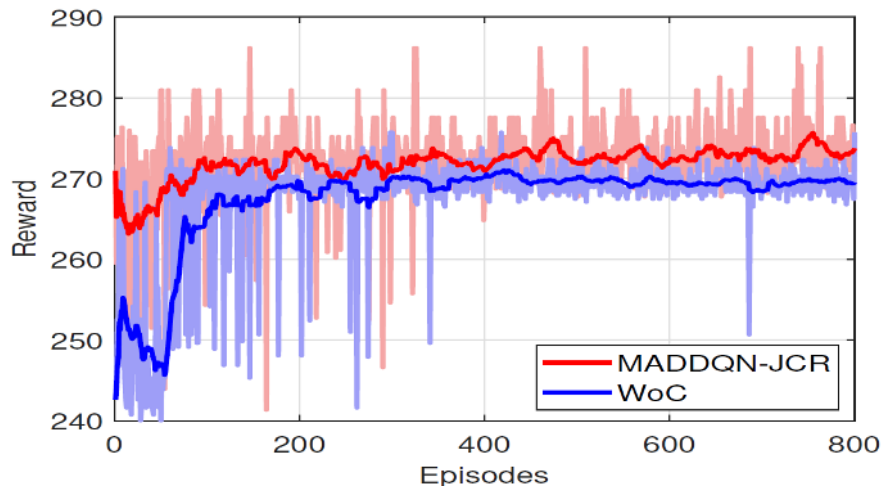


Figure 3-21: Convergence analysis.

Figure 3-21 illustrates the reward-based convergence behaviour of the proposed MADDQN-JCR and WoC methods. Both approaches demonstrate stable convergence, confirming their effectiveness in establishing routing paths.

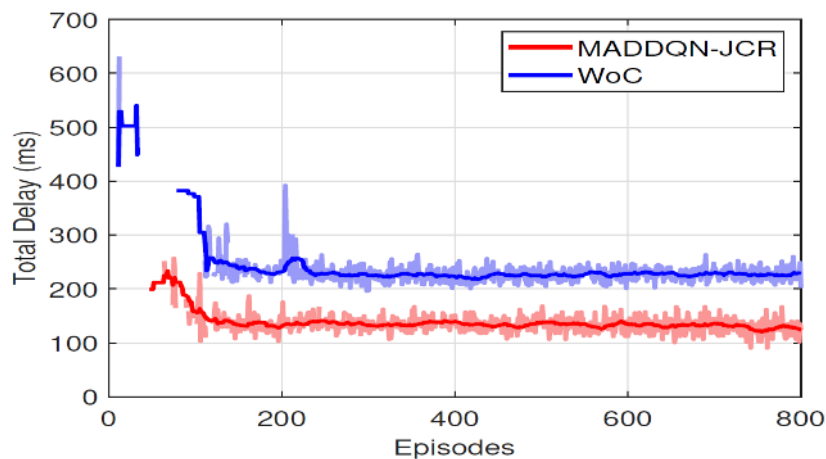


Figure 3-22: Delay analysis.

Figure 3-22 presents the total delay trend during the training phase for both methods. The results indicate that MADDQN-JCR achieves a lower total delay than WoC, highlighting the benefits of the JCR mechanism. Notably, delay data for the initial training episodes is not recorded, as shown in Figure 3-22, due to the agents' limited experience. At this stage, the agents rely more on random actions for exploration, resulting in a lower probability of successfully determining routing paths.

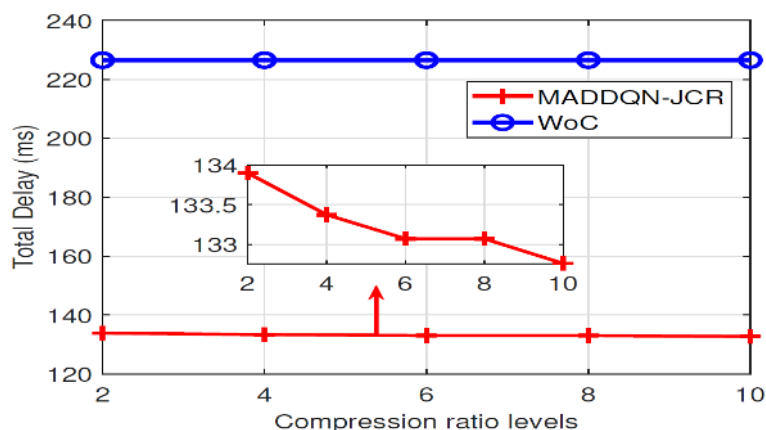


Figure 3-23: Effect of CR levels.

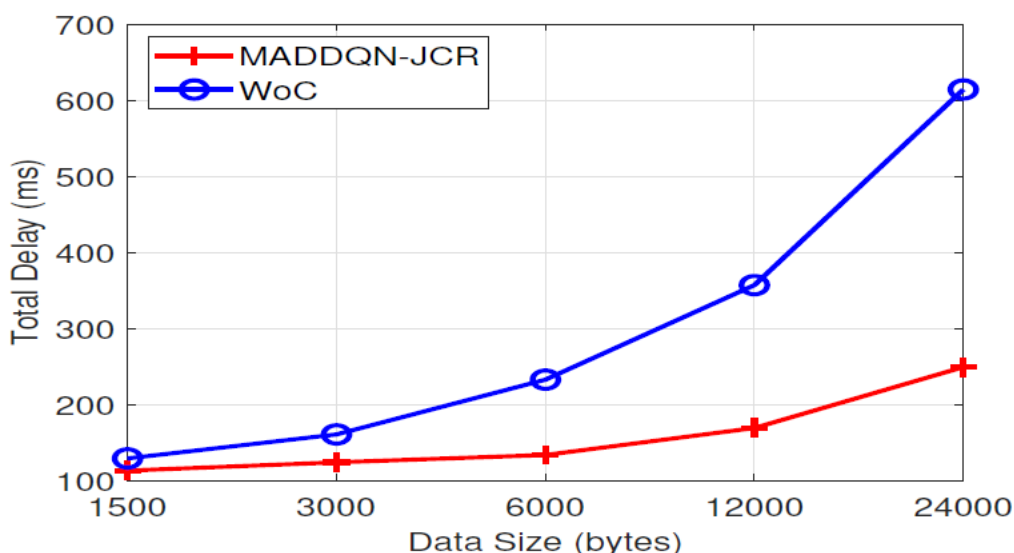


Figure 3-24: Effect of data size.

Figure 3-23 presents the total delay as a function of the number of CR levels (L), illustrating its impact on system performance. The results indicate that increasing L reduces the total delay achieved by the MADDQN-JCR method. In contrast, the WoC method exhibits consistently higher delay, remaining unaffected by L due to its lack of a DC scheme. However, while a greater number of CR levels enhances MADDQN-JCR's performance, it also expands the action space of the source LEOSat agent, potentially slowing algorithm convergence. Additionally, Figure 3-24 shows the total delay as a function of data size (K), revealing that larger data sizes lead to increased delay. The MADDQN-JCR method consistently outperforms the WoC method, demonstrating lower delay across varying K .

Finally, Table 3-4 demonstrates the impact of the maximum queue length on average delay and target data loss rate. Specifically, the amount of target data $K = 6$ KB and the results are determined over 1000 samples in the testing phase. Table II shows that the MADDQN-JCR method significantly reduces the average delay and data loss rate compared to the WoC approach, owing to its JCR strategy. Specifically, with $q_{\max} = 15$ KB, the MADDQN-JCR method records an average delay of 132.45 ms and a data loss rate of 5.9%, whereas the WoC approach yields an average delay of 223.23 ms and a data loss rate of 24.5%. At $q_{\max} = 45$ KB, the MADDQN-JCR method results in an average delay of 133.96 ms with no data loss, compared to the WoC method's average delay of 225.48 ms and a data loss rate of 0.1%.

Table 3-4: Effect of queue length ($q_{\max}$).

Parameters		MADDQN-JCR	WoC
$q_{\max} = 15$ KB	Average delay	132.45 (ms)	223.23 (ms)
	Data loss rate	5.9%	24.5%
$q_{\max} = 45$ KB	Average delay	133.96 (ms)	225.48 (ms)
	Data loss rate	0%	0.1%

3.5 MODEL-BASED 3D NETWORK MANAGEMENT AND SDN INTEGRATION

Reliable aeronautical networks connectivity enables real-time communications to support airspace safety critical operations including flight safety, navigation, and data traffic management. However, next generation avionic applications are data rate intensive and delay sensitive associated with an increased aircraft fleet, which challenge current aeronautical SATCOM and VDL-2 networks connectivity and services provisioning capabilities. Fifth and Sixth Generation (5G and 6G) mobile wireless communication networks have been emerged to support ubiquitous, resilient, and reliable connectivity. The integration of 5G terrestrial (A2G (air-to-ground) links) and non-terrestrial networks (LEO satellites and High-Altitude Platforms) forms 6G networks and communication systems with heterogeneous capabilities. These can be leveraged and further developed to enhance network capacity and reliability of aircraft communications, as well as increase data rate received per space, aerial and ground users, mitigating low wireless communication and network service coverage in remote areas. The initial report, D4.1 [1] introduced the fundamentals and basic operations of the proposed data driven end-to-end solutions, focusing on SDN 3D network orchestration, resilient handover and message routing solutions.

- An SDN orchestrator in coordination with SDN controller can leverage the physical and higher network layers' QoS metrics to select or reconfigure routing paths based on critical cross-layer parameters to support real-time aircraft operations.
- A graph with multifactor edges-based handover prediction method and evaluation for aircraft message routing. The main handover decision considers traditional link weights. The resulting schedule includes the base station identifier and the time when the handover request should be triggered.
- Aircraft message routing methods that ensure continuous network coverage with active gNB handover planning.

This work study considers optimising critical system parameters, particularly nodes link or channel spectrum resources and performance metrics. It explores various graph edge parameter weights that reflect additional network preferences such as node spectrum, handover to a cell from the same operator and end-to-end communication costs. It contributes towards planning, deployment and maintenance of future 5G and 6G aeronautical networks, which can be costly for aeronautical network infrastructure providers (ANIPs) and network service providers (ANSPs) to manage and offer guaranteed aircraft E2E communication services. The emerging aeronautical networks require agile SDN and network function virtualisation (NFV) paradigms to effectively manage communication and network resources. To generate optimal techno-economic value of 5G or 6G aeronautical networks, effective cost-driven resources management approaches have been developed through effecting aircraft message routing functions to efficiently offer various data and communication traffic flows to multiclass aeronautical users, considering optimal network and data communication links resources allocation to support the emerging avionic applications and cost-effective aircraft operations.

3.5.1 Background

Various aircraft types including commercial, unmanned aerial systems (UAS), electric taxi with aircraft capabilities require different data and communication links to support flight safety and meet business requirements. Terrestrial network capacity and services provisioning capabilities are advancing but ground as well as aerial and space users increase at exponential pace. This requires in turn the integration of terrestrial, aerial and space networks to support efficient management for network links resources and affordable connectivity provisioning to support aeronautical networks deployment. Aircraft flight trajectories span different altitudes at which aircraft can communicate through multiple network layers comprised of terrestrial, HAP and satellite networks. However, accommodating future high data rate applications and an increased aircraft fleet may pose challenges for current aeronautical networks, such as SATCOM and VDL-2. Sixth Generation (6G) mobile networks are being developed to achieve ubiquitous, resilient, and sustainable connectivity throughout the world. Undoubtedly, the 6G integrated networks will have a high initial investment as deploying the non-terrestrial network will involve significant upfront costs for satellite launches, ground stations, and infrastructure from many different stakeholders. The operational costs of maintaining and operating the new emerging networks, including satellite and HAPS maintenance, can be also expensive. Therefore, to allow competitive and sustainable aeronautical market, joint network topology and traffic flows engineering associated with cost-driven handover methods are crucial to operate efficient and cost-effective integrated terrestrial and non-terrestrial networks.

3.5.2 Motivation and Contribution

This work aims to harness the benefit of the integration for the three-dimensional (3D) networks comprised of terrestrial, HAP, and satellite networks. It focuses on deploying intelligent joint traffic flows engineering and seamless handover approaches that can support efficient management of network's links spectrum, power, and available capacity across the different network links. Thus, the work is motivated by the following points:

- Introduce approaches that can leverage the integration of the terrestrial (air-to-ground links (ATG)) and non-terrestrial networks, including LEO satellite links and High-Altitude Platform (HAPS) to support efficient aeronautical networks connectivity.
- Introduce approaches that can support 6G network technologies to be aligned with the proposal of future aeronautical networks.
- Support cost-effective seamless handover in 3D networks.
- Provide cost-effective aircraft communication connectivity services.

This work proposes traffic flows engineering supported by handover and route switching framework that can guarantee cost effective message routing and resilient communications in the integrated 6G aeronautical networks. The proposed 3D network routing approaches have been evaluated by using MATLAB simulation environment, considering the performance communication metrics data collected from different flights from Dublin to Heathrow.

3.5.3 System Model

The system model is an SDN-enabled 3D integrated network comprised of terrestrial, satellite, and HAP network nodes. The system is managed by a controller and network orchestrator (NO) that manage cost-driven message routing, nodes selection and routing paths computation. The system model is represented by a 3D SDN network model composed of network infrastructure data plane containing ATG, HAPS, and LEO satellite nodes, control and application planes. A 5G core serves the different network nodes deployed in the three

network layers, which is also managed by the centralised controller, as shown in Figure 3-25. The NO, running within the Management and Orchestration (MANO) application, leverages geographic coordinates information of data plane nodes and aircraft position tracking. It can be also empowered with an artificial intelligence (AI) inference model trained using network links performance metrics analytics to support predictive network and traffic-aware aircraft message routing and handover decisions.

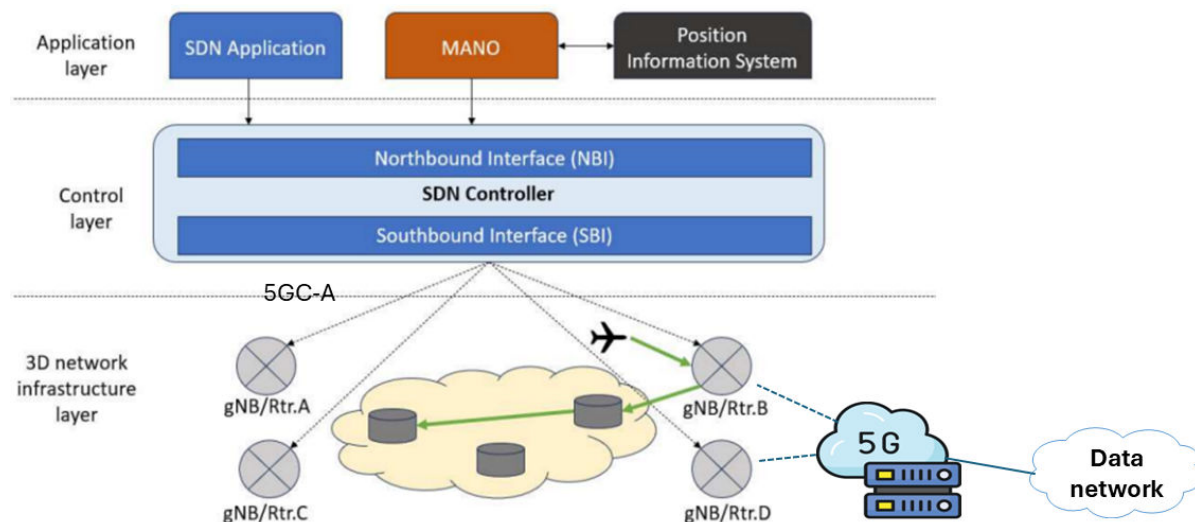


Figure 3-25: SDN-enabled Integrated 3D network architecture.

The control layer exposes the information and status of network data plane through the northbound interface to the application layer, which is processed by the NO to support effective data plane management, adaptive traffic routing and automated network resources (i.e., ports, network links and paths) management. The applications including the NO apply specific algorithmic logic to meet their objectives and performance metrics through the SDN controller. This applies in turn applications algorithm logic to the data plan by using the OF protocol via the southbound interface through which specific rules are configured at network nodes ports enforced by resources management policies.

The aircraft airborne router has multiple interfaces that can be simultaneously connected to the different base stations deployed in the 3D network layers. A message from air services is forwarded to one of these interfaces according to a routing policy. The optimisation of aircraft message routing policy requires input from the service requirements and network conditions.

3.5.4 Problem Description

3D communication networks offer different performance metrics that yield different cost of message transmission and routing. Aircraft communications should be maintained according to the required communicator performance (RCPs). It also requires seamless and constant connectivity that supports efficient route planning, fuel management, and maintenance, with aim to reduce flight delays and data communications operational costs. 6G integrated networks may have high initial investment for deploying non-terrestrial networks and additional terrestrial ground stations, though operational and maintenance costs can be also expensive. Next generation aircraft and general aeronautical aviation will benefit from 3D integrated networks through meeting aircraft's RCPs, though network data links traffic granularity transmission, utility, resources allocation should be factored in dynamic routing metrics that can assess the total techno-economic value of end-to-end communications along aircraft flights. This work investigates solutions to these challenges, considering simulated 3D integrated network offering data links based on various cost-driven routing and handover methods. These offer

message routing and communications to different aircraft flights from Dublin to Heathrow airport, aiming to enable competitive 6G 3D network services provisioning to aircraft.

3.5.5 Proposed Solution

Joint traffic flows engineering associated with seamless handover and e2e routing planning is proposed to support cost effective resilient aircraft communications. Based on the proposed system model, it is assumed that the 3D network infrastructure is composed of unified 5G/6G radio access networks (RANs) that are also integrated with the SDN-based transport network and the common core network. In the transport and core networks, the software-based architecture allows flexible network reconfiguration to accommodate dynamic service demands on the different network capabilities. The trajectory of aircraft flight path can be predicted based on the flight management system, where the trajectories of LEO satellites and High-Altitude Platforms (HAPs) aircraft can be considered deterministic, given their constellation topology construction and HAP trajectory planning.

A data-driven network orchestrator is fed by multiple data and information sources. These include aircraft information including current position, flight trajectory, link availability, and network meta and dynamic information including base station positions, satellite trajectory, remaining network resources, end-to-end communication performance predictions. This overall information enables the NO to define and apply message routing and handover plans. The associated base station of each link is given by a successful handover decision. The NO assesses the Routing Plan Update Decision and steers the traffic among those available links according to their status and the service requirements.

The proposed solution is based on a graph-based handover framework that creates a service time-based graph with the vertices representing the base station instances over a predicted period; and the edges' weight are computed according to the costs of communication usage and handover. The decision about the sequence and timing of handovers is designed by computing the shortest path in the graph. This decision aims to minimise the cost of utilisation, subject to the connectivity requirements and network resource constraints.

3.5.5.1 Aircraft Message Routing Policies

Two methods for message routing and resource optimisation process are studied as shown in Table 3-5. In the first method, the objective is given by the handover decision (e.g., the visibility time) and only switch to an alternative link if the current conditions do not satisfy the service requirements. In the second method, the load balance and end-to-end latency is jointly optimised by calculating a utility function for each link according to the current conditions and switching to the link that minimises the utility function.

Table 3-5: Two methods for message routing and resource optimisation process.

Method 1: Routing Policy Priority		Method 2: Joint Load Balance and E2E Latency	
$\max_g RP(g)$		$\min_g \alpha \cdot \frac{D(a,g)}{D} + (1 - \alpha) \cdot \frac{L(a)}{C(g)}$	
subject to	$g \in G$ $P_{\{RX\}(g)} > \overline{P_{\{RX\}}}$	subject to	$g \in G$ $P_{\{RX\}(g)} > \overline{P_{\{RX\}}}$

	$\sum_{a \in A} L(a) \leq C(g)$ $R'(a, g) > \bar{R}$ $D'(a, g) < \bar{D}$		$\sum_{a \in A} L(a) \leq C(g)$ $R'(a, g) > \bar{R}$ $D'(a, g) < \bar{D}$
--	---	--	---

In Method 1, the routing policy, RP , is the aircraft routing priority decided by the handover algorithm. In method 2, α is the objective weight, $D(a, g)$ is the predicted end-to-end delay from the aircraft a to the associated gNB g , $\bar{D}$ is the service's delay requirement, $L(a)$ is the aircraft's required data load, and $C(g)$ is the total gNB capacity. Both methods are subject to the service requirements described below:

1. The associated gNB g belongs to the set G of connected gNBs over multiple links of the aircraft.
2. The received power of the associated gNB $P_{\{RX\}}(g)$ shall be higher than the sensitivity threshold $\bar{P}_{\{RX\}}$, indicating availability.
3. The sum of the data load $L(a)$ of all aircraft a at the gNB g should be less than the total capacity $C(g)$ of the gNB.
4. The predicted data rate of the associated gNB $R'(a, g)$ shall be higher than the service data rate requirement $\bar{R}$.
5. The predicted end-to-end delay of the associated gNB $D'(a, g)$ shall be less than the service delay requirement $\bar{D}$.

3.5.5.2 Cost-Driven Handover and Base Station Association

In the airborne router, each interface is connected to an independent radio equipment. This equipment should be associated with a base station in the unified radio access network of the integrated 3D system. It is assumed that the base station association follows the traditional Xn intra-RAN handover process, as specified in the 3GPP standard [49].

To support NTN UCs, in Release 16 of the 3GPP standard for 5G/6G mobile communications [50], new handover condition triggers have been introduced to take advantage of the predictable mobility of satellites and other aerial base stations. In addition to the traditional measurement-based trigger, it is expected that the handover can be triggered by location-based events, elevation angle, and time-based events.

Specifically, in this work, it is assumed that handovers are triggered by time-based events. These events are generated by a proactive handover scheduler that requires the input from the geographical information (i.e., aircraft flight plan, the deterministic LEO satellite positions in orbit and the HAP and ATG base station static locations) and network conditions provided by the Network Orchestrator. Given that input, it predicts the visibility time of the neighbouring base stations and schedules a handover event to be triggered once the link is available, as illustrated in Figure 3-26. This process is executed periodically within a given time window.

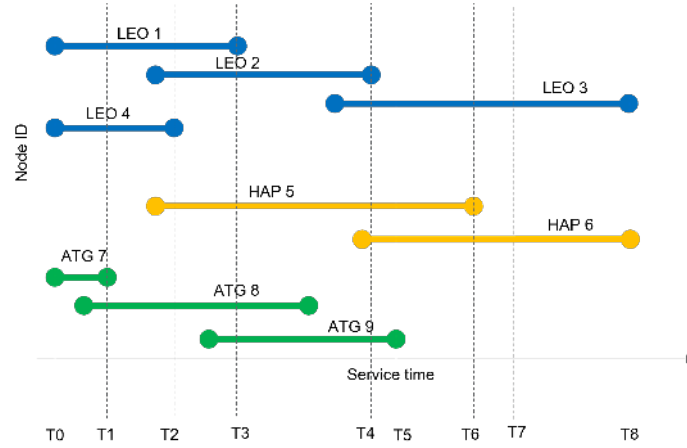


Figure 3-26: Predicted service times of visible gNBs to the aircraft.

To generate the schedule of handover decisions, the graph-based algorithm of a previous work [51] is employed. A directed graph is created where the nodes represent potential target base stations, and directed links represent possible handovers between two base stations with an overlap in visibility time, as described in [52]. The handover decision is abstracted by computing the shortest path between the start and end points, as illustrated in Figure 3-27.

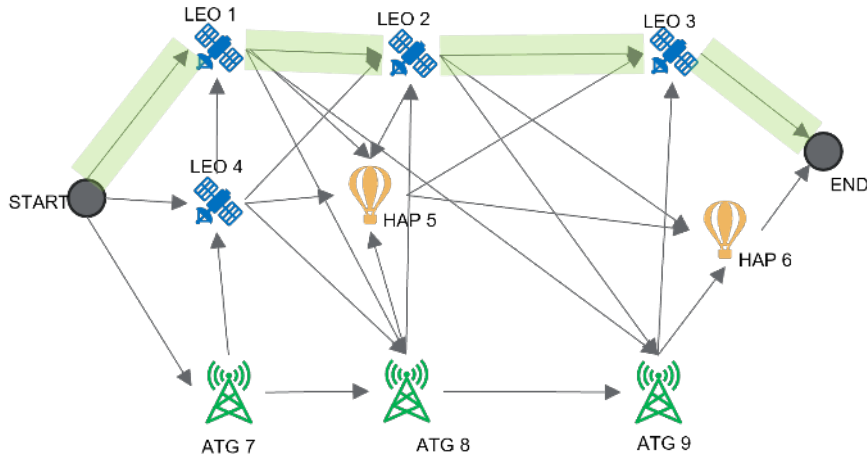


Figure 3-27: Directed graph generated from predicted service time, with the main handover decision (highlighted in green) is calculated as the shortest path of the graph.

This framework allows for flexible implementation of multi-factor handover optimisation. Different link weights are defined for various objectives, such as minimising the number of handovers while maintaining the required reliability (as proposed in [51]). A cost-driven objective is proposed to minimise the costs of communication and handover procedures. Thus, the graph edge weight W is defined as the rate of consumption in the remaining time:

$$W = \frac{C_c + H_c}{T_{rem}} \left[\frac{\text{€}}{\text{second}} \right];$$

$$C_C = U_C \cdot \frac{B_{req}}{B} \cdot \frac{T_{rem}}{T_{interval}} \cdot T_{\{delay\}} \cdot n_{\{hops\}} \cdot 2; \quad (3-43)$$

$$H_C = U_C \cdot \frac{B_{ho}}{B} \cdot n_{\{hoPkts\}} \cdot T_{\{hoDelay\}} \cdot 2,$$

where C_C is the communication cost, H_C is the handover cost, U_C is the usage cost per second, B_{req} is the required bandwidth, B is total bandwidth, T_{rem} is the remaining visibility time, $T_{interval}$ is minimum packet interval time, T_{delay} is the maximum packet delay, n_{hops} is the number of hops, n_{hoPkts} is the number of handover control packets, B_{ho} is the required bandwidth for handover, $T_{hoDelay}$ is the handover control packet delay, and all is multiplied by 2 as the usage cost of a base station on both uplink and downlink is considered.

It is assumed that the network usage cost is a function of the Total Cost of Ownership (TCO) of a base station per year. The yearly TCO of each type of base station (ATG, HAPS and LEO) is estimated according to the literature to derive the usage cost per second (U_C).

Table 3-6: Cost of operation for different gNBs.

gNB type	Duration of operation	Total cost	Cost per year	Cost per second
ATG	n/a	n/a	€168 k	€5.3e-3
HAP	10 years	€5 M	€500 k	€15.9e-3
LEO	10 years	€1.82 M	€182 k	€5.8e-3

Given the yearly TCO above, the usage costs are summarised in Table 3-6. The value of the Usage Cost per second (U_C) is used as a reference to weigh the handover and message routing decisions among the different sub-networks.

The proposed solution is summarised as follows:

- The main handover decision is derived from the shortest path in the directed graph, considering the graph edge weights defined in the previous step.
- The resulting schedule will trigger the handover of the base station connected to the primary interface on the airborne router.
- For the cell association in other interfaces, an alternative and orthogonal schedule can be generated.
- A completely different set of base stations from the main decision should be scheduled by removing the already scheduled base stations to generate a new graph, as illustrated in Figure 3-28.

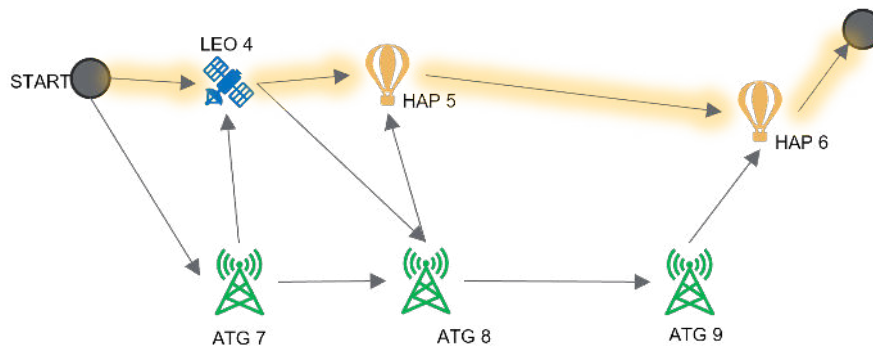


Figure 3-28: Directed graph generated by removing the main handover decision. The alternative handover decision (highlighted in yellow) is calculated as the new shortest path of the new graph.

- Hence, the handover decision for the second interface is derived from the shortest path in the new graph.
- To schedule the handover decisions in the remaining interfaces, the algorithm executes the same process iteratively. At the end of all iterations, each interface will be simultaneously connected to a different base station, creating alternative paths for aircraft messages and more resilient communication.

Algorithm 1: Compute handover schedule

Output: handover plans for main link and alternative links

```

1  For each user  $u$ 
2      Find the set of gNBs within a radius  $R$  (e.g.,  $R = 6000$  km)
3      Calculate visibility time of the gNBs with Algorithm 2
4      Generate the handover graph  $G$  with Algorithm 3
5      Define the graph edge weights based on the utility function  $W$ 
6      Define the main handover decision plan by computing the shortest path of the graph  $G$ 
7      For each alternative link  $a$ 
8          Generate a handover graph  $H$  by removing from  $G$  the nodes of the previous handover plan
9          Define the alternative handover decision plans by computing the shortest path of the graph  $H$ 
10     end
11 end

```

Algorithm 2: Calculate visibility time

Input: set of gNBs within the radius R from the user u

Output: visibility time

```

1  For each gNB  $g$  within the radius  $R$ 
2      Estimate the received power from the gNB  $g$  to the user  $u$ 
3      Predict the period of time that the received power is above the receive sensitivity
4      Define this period as the visibility time
5  end

```

Algorithm 3: Compute handover graph

Input: set of gNBs within the radius R from the user u and their visibility time

Output: handover graph G

```

1  Calculate adjacency matrix  $A$ :
2  For each gNB  $i$ 
3      For each gNB  $j$ 
4          If visibility of gNB  $i$  overlaps with visibility of gNB  $j$  Then
5               $A(i,j) = 1$ 
6          Else
7               $A(i,j) = 0$ 
8          End
9      End
10 End
11 Include virtual origin node in  $A$ 
12 Include virtual ending node in  $A$ 
13 Define a directed graph  $G$  from the adjacency matrix  $A$ 

```

3.5.6 Simulation Results

The performance evaluation of the proposed methods is evaluated by using our developed MATLAB simulator, considering the communication performance metrics collected for aircraft flights, Dublin to Heathrow airport. Noting that the flight is served by ATG, HAP and LEO

constellation base stations simulated along the flight trajectory. Simulation scenarios have been conducted to investigate and analyse the handover frequency, throughput, delay, and final costs of operation. Furthermore, the performance of proposed method is compared to the standard handover methods, showing the trade-off between cost and network performance. To characterise the cost effectiveness of the proposed method, a combination of key metrics is analysed: 99th percentile of delay, average data rate, average network load, and the cost of network usage.

The end-to-end delay is calculated, considering the transmission, propagation, processing, relay network, and core network delays. The data rate is calculated based on by Shannon capacity given the SNR of network data links. The network load is computed by summing the bandwidth used by each aircraft in each subnetwork. The cost of network usage represents the accumulated transmission and reception costs of data and control packets throughout the duration of the simulated flight. For each data packet, the delay and bandwidth used is calculated, factoring the total communication cost similarly. The total handover cost is calculated as the number of handovers multiplied by the handover cost.

Depending on which subnetwork the packet is transmitted from, or the handover was executed, the usage cost per second (UC) varies according to Table 3-6. Therefore, the cost of network usage (NC) is the sum of the total communication cost and total handover cost.

3.5.6.1 Simulation Setup

The performance of the proposed methods is evaluated during aircraft flights from Dublin to London, as illustrated in Figure 3-29, using Collins Aerospace in-house simulation framework. The aircraft mobility data trace is extracted from real flight information available in the Flight Aware tool [53].

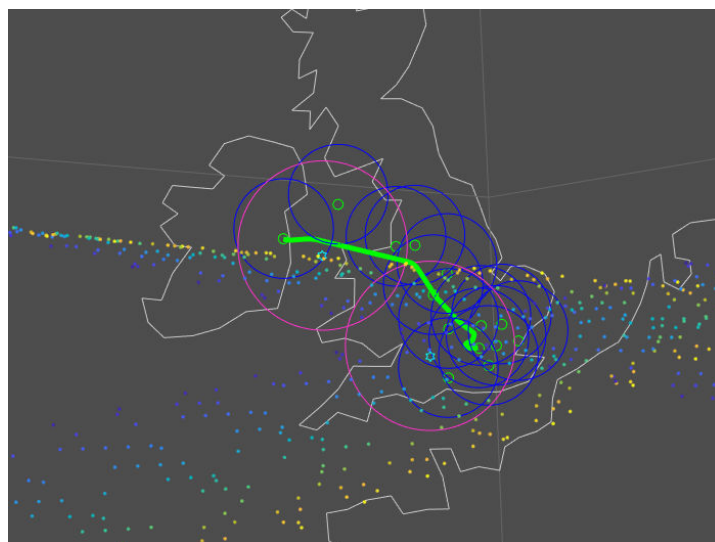


Figure 3-29: Regional flight BAW827 from Dublin to London (green line). The pink circles illustrate the HAPS coverage, the blue circles illustrate the ATG coverage, and the yellow-blue dots are the LEO satellites.

To load the network, 50 aircraft trajectories are simulated. One aircraft follows the real flight trace while the other 49 are randomly placed around the main traced aircraft. The 49 aircraft follow the main one keeping the same distance throughout the flight. It is assumed that the ATG base stations are deployed at airports. The deployment of two HAP stations to cover the entire flight length and to ensure redundant links to the ATG and LEO stations is considered.

It is assumed a typical LEO satellite constellation. The specific system parameters are detailed in **Error! Reference source not found..**

Table 3-7: System Parameters

PHY transmit parameters			
LEO satellite		HAP station	
Altitude	550 km	Altitude	20 km
Satellites/orbit	22	Bandwidth	20 MHz
Long. shift/orbit	5°	Carrier frequency	2 GHz
Orbit inclination	54°	Transmit power	16 dBW
EIRP/MHz	14 dBW/MHz	Antenna	Phased array
Bandwidth	400 MHz	Beamwidth	173.5 degrees
Carrier frequency	20 GHz	ATG station	
Antenna (data)	Circular	Bandwidth	20 MHz
Aperture	0.5 m	Carrier frequency	3.5 GHz
Max gain	38.5 dBi	Antenna	Phased array
Antenna (control)	Phased array	Beamwidth	166.5°
Beamwidth	45°	Transmit power	16 dBW
PHY receive parameters			
Aircraft – LEO receiver		Aircraft – HAP receiver	
Antenna (data)	Circular	Antenna	Phased array
Aperture	0.6 m	Beamwidth	180°
Max gain	39.7 dBi	Aircraft – ATG receiver	
Antenna (control)	Phased array	Antenna	Phased array
Beamwidth	180°	Beamwidth	180°
Aircraft - common			
PHY sensitivity	-120 dBm	Noise power	-107.013 dBm
Cable loss	6 dB		
Handover parameters			
Measurement-trigger		Time-trigger	
RSSI threshold	-120 dBm	Prediction period	5 min
Sampling period	30 s		

3.5.6.2 Performance Evaluation

The cost of network usage is evaluated by comparing the standard measurement-based handover method with the previously proposed visibility-based method and the cost-based method. Figure 3-30 presents the average network load considering the 50 aircraft using 1 MHz bandwidth and requiring a minimum rate of 1 Mbps each during the Dublin-London flight including each subnetwork; and Figure 3-31 shows the respective average cost of network usage per aircraft over the flight.

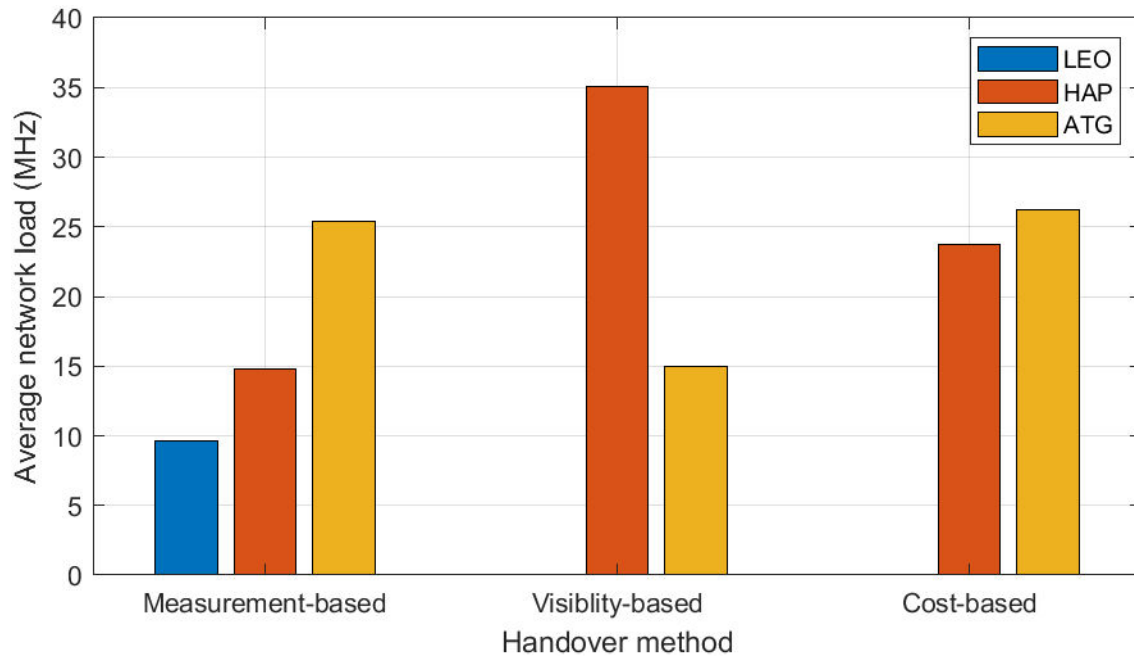


Figure 3-30: Average network load distribution among the different subnetworks for each handover method.

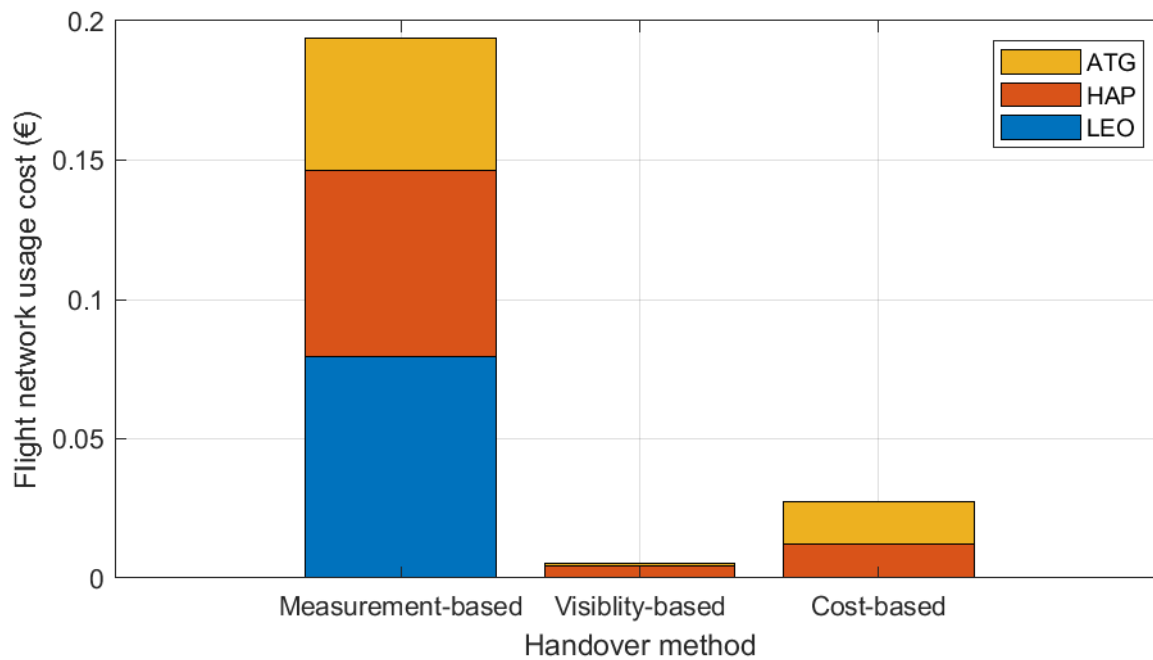


Figure 3-31: Cost of network usage over the flight for each handover method.

One can observe that in the measurement-based method the network load is distributed according to the link quality as some aircraft are covered by ATG stations while others are only covered by LEO or HAP stations. The average cost per aircraft is higher than the other methods as it uses more HAP network, which is expensive data link comparing to the others. Furthermore, the measurement-based method does not minimise the number of handovers as the other methods. Consequentially, the increased number of handovers contributes to a higher cost using this method.

In the visibility-based method, more aircraft are assigned to the HAP network since its cells have bigger coverage than the ATG cells, compared to the cost-based. Consequentially, there are fewer handovers in the HAP network due to the longer visibility time compared to the ATG or LEO network, leading to a smaller cost.

The cost of network usage of the cost-based method as low as that of the visibility method, indicating that minimising the number of handovers contributes to the cost reduction. This method mainly assigns the aircraft to the ATG network where the usage cost is cheaper while assigning the few aircraft that are out of the ATG coverage to the LEO network, to maintain the connectivity, meeting performance requirements, as depicted in Figure 3-32 and Figure 3-34.

Figure 3-32 shows a higher delay using the LEO network in the traditional measurement-based method compared to the other methods. This is because the measurement-based method only hands over from a gNB when the channel quality is already below a threshold, and in the case of a LEO gNB that moves in high speed, the channel degrades quickly as the distance increases, causing more retransmissions and increasing delays. The long delays in the LEO network are the main factor for having higher costs using the measurement-based method. The cost-based method is capable to handover from a LEO gNB before the channel quality degrades, as it understands that more retransmissions will mean more usage costs and thus reducing the 99.9th percentile delay from 10 to 2 seconds.

Figure 3-33 depicts a closer look at the smaller values to visualize the visibility-based results in comparison to the cost-based. The delays of the HAP and ATG networks are very low (in the order of 100 ms) and have negligible difference comparing the handover methods.

Figure 3-34 shows the average data rate, and it is observed that all handover methods can meet the requirement of 1 Mbps on average. The measurement-based method data rate is higher compared to the other methods since the handover decision is mainly based on the gNB with the best signal quality, while in the others, this criterion is secondary.

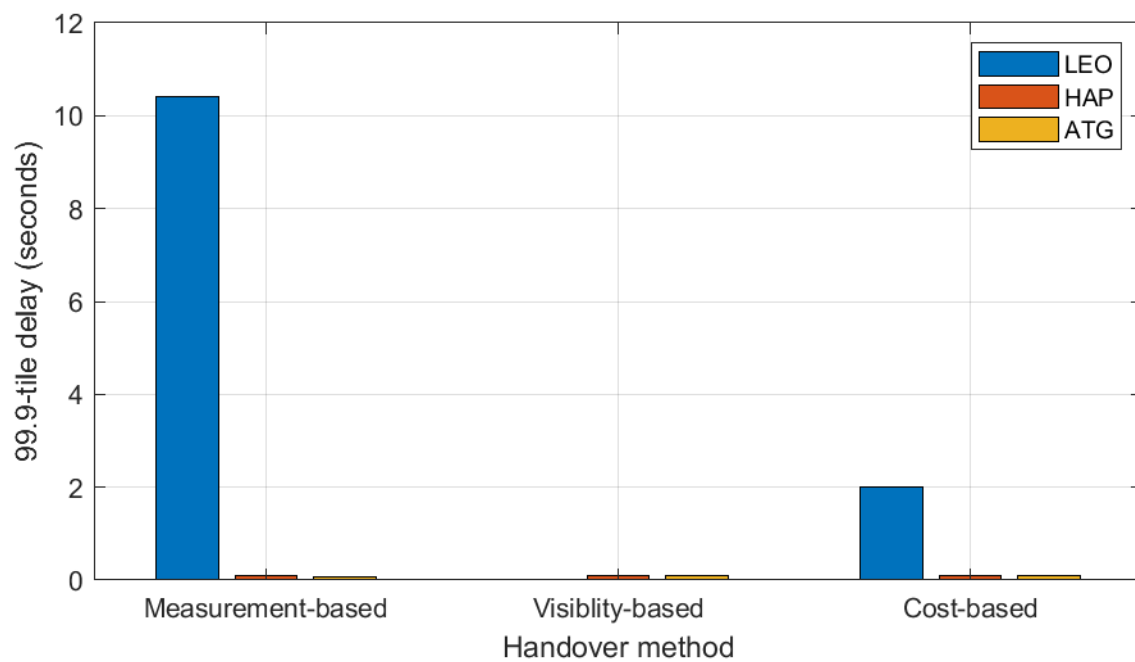


Figure 3-32: 99th percentile of delay over the flight for each handover method.

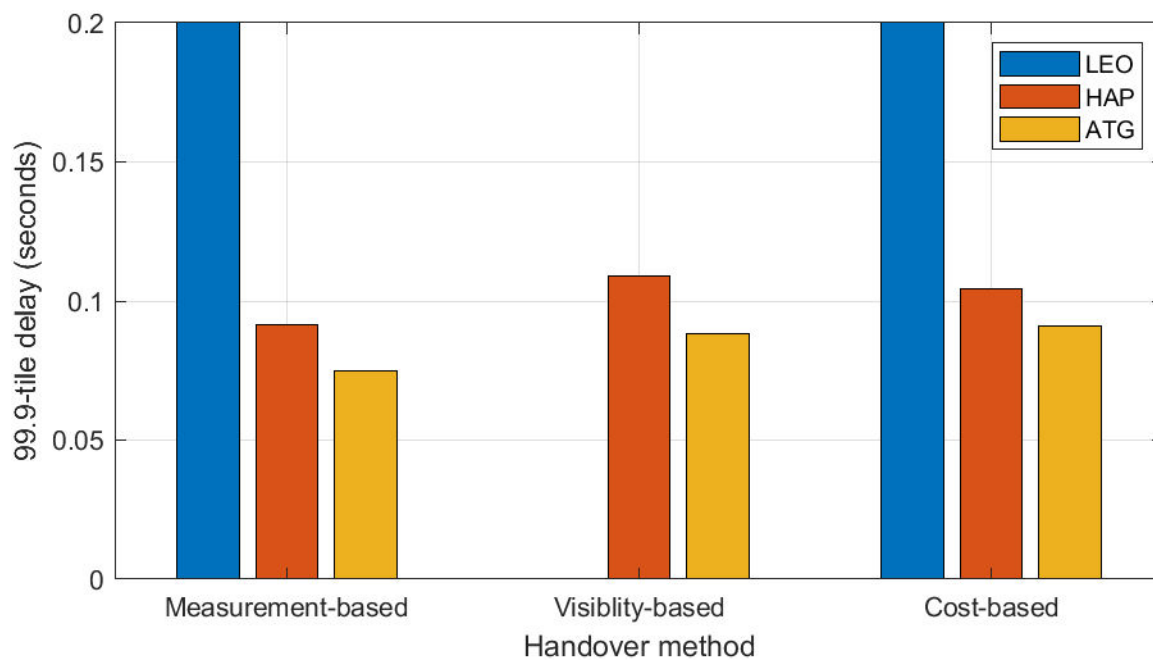


Figure 3-33: [close up between 0 and 0.2 seconds] 99th percentile of delay over the flight for each handover method.

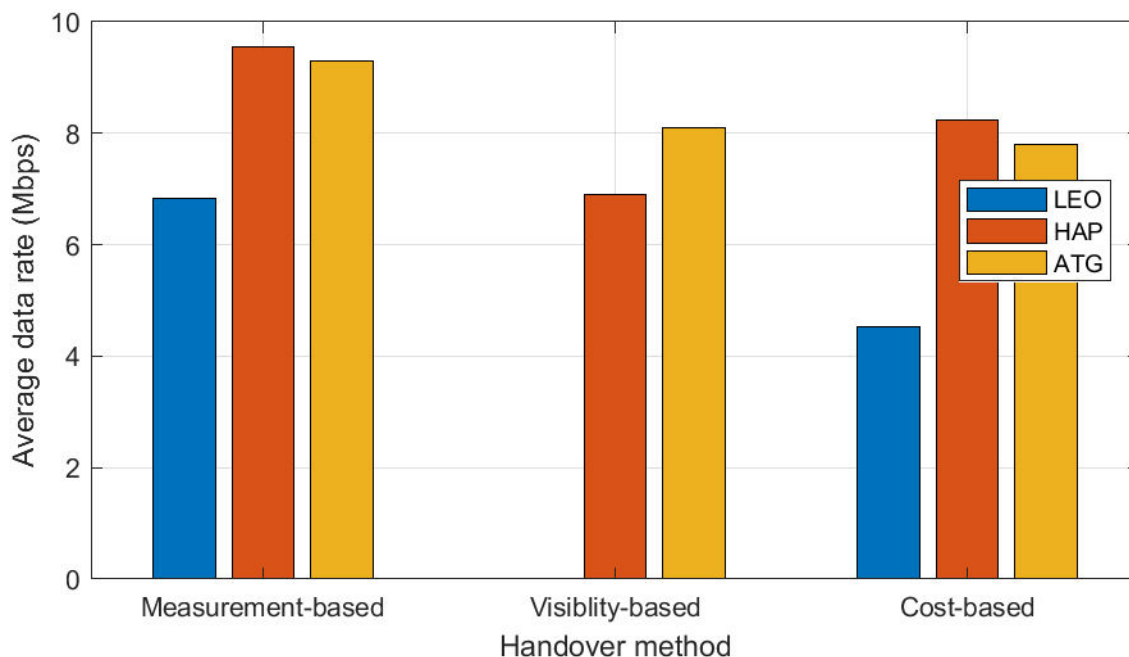


Figure 3-34: Average data rate over the flight for each handover method.

3.6 DISTRIBUTED SDN CONTROLLER PLACEMENT

Satellite networks provide connectivity to remote areas, but efficient network management is crucial for seamless operations and resource optimisation. SDN offer flexibility, making them appealing for satellite network management. The dynamic placement of SDN controllers is crucial for an SDN-enabled satellite network, as it must adapt to dynamic topology changes, potential disruptions, limited capacity, and considerable latency. Existing literature lacks dynamism for controller placement in satellite networks considering multi-orbit scenarios, time-varying load and topology and facilitating swift adaptation to network variations. This work optimises network performance by strategically positioning SDN controllers in multi-orbit satellite networks using a genetic algorithm based on network topology, traffic distribution, and link stability constraints, considering low-earth and medium-earth satellites. The proposed scheme is evaluated through simulations using operational constellation parameters.

3.6.1 Background

Satellite networks are crucial for modern communication infrastructure, providing coverage and connectivity to remote and underserved regions. SDN is a promising paradigm for efficient network management in complex environments. SDN's centralised control and programmability enable agile resource optimisation and dynamic adaptation. Due to the distance between satellites and limited onboard resources, deploying SDN controllers directly onto satellites is essential for reduced communication latency and real-time decision-making capabilities. This is especially important in environments where immediate responses to network events are critical, such as disaster management systems or remote healthcare monitoring. However, the strategic placement of SDN controllers is a fundamental challenge in deploying SDN in satellite networks, especially in non-geostationary orbit (NGSO) satellite networks, which include various satellite types with distinct latency, capacity, and coverage area characteristics.

3.6.2 Motivation and Contribution

In SDN-enabled satellite networks, satellite and ground nodes can act as SDN switches and controllers. Nodes acting as SDN switches store rules for routing network traffic. As the SDN controller is responsible for managing the rule space of SDN switches, having a single controller at a ground node increases the flow setup delay of large-scale satellite networks (e.g., Starlink constellation) where the distance between satellites and ground nodes is significantly high. Additionally, the limited visibility of ground controllers with satellites exacerbates the challenge, as frequent satellite handovers reduce the controller's ability to maintain consistent network awareness. Controller placement problem (CPP), which involves determining the number and locations of multiple controllers, is a well-investigated research problem for terrestrial networks. However, placing multiple controllers in satellite networks involves additional challenges due to time-varying load and changes in link availability because of the movement of satellites and atmospheric effects. Due to these dynamic factors, static controller placement, where the location of the controllers and controller-switch associations do not change over time, is inefficient for SDN-enabled satellite networks. Therefore, it is necessary to dynamically update the controller placement strategy in satellite networks. The resource limitation of satellites is another critical parameter that affects the controller's service rate. Moreover, satellite network operators opt for multi-orbit networks to match diverse UCs. Integrating multi-orbit scenarios in SDN controller placement strategies is crucial for efficient network management and optimisation in the evolving landscape of satellite networks.

In this work, an approach for controller placement in SDN-enabled multi-orbit satellite networks (CoMOSat) to deploy the SDN control plane hierarchically on MEO, LEO satellites, and ground stations with a minimum reconfiguration for varying traffic and topology is proposed. The proposed scheme classifies LEO satellites and ground nodes into multiple domains based on proximity. Subsequently, a genetic algorithm (GA)-based solution is proposed to elect domain controllers and master controllers for each domain. Finally, a control plane update strategy with minimal reconfiguration is designed. The main contributions of this work are:

- A controller placement cost is proposed based on flow or forwarding rule setup delay, the number of changes in mappings between the controllers and managed nodes, and the delay to synchronise the master controllers.
- A GA model is formulated to jointly place domain controllers on the ground or LEO nodes and master controllers on MEO nodes. This model's fitness function is designed based on the controller placement cost.
- An update strategy is designed to modify the existing set of domains and the controller placement, considering the effects of time-varying topology and traffic, and targeting minimal reconfiguration overhead.

3.6.3 System Model

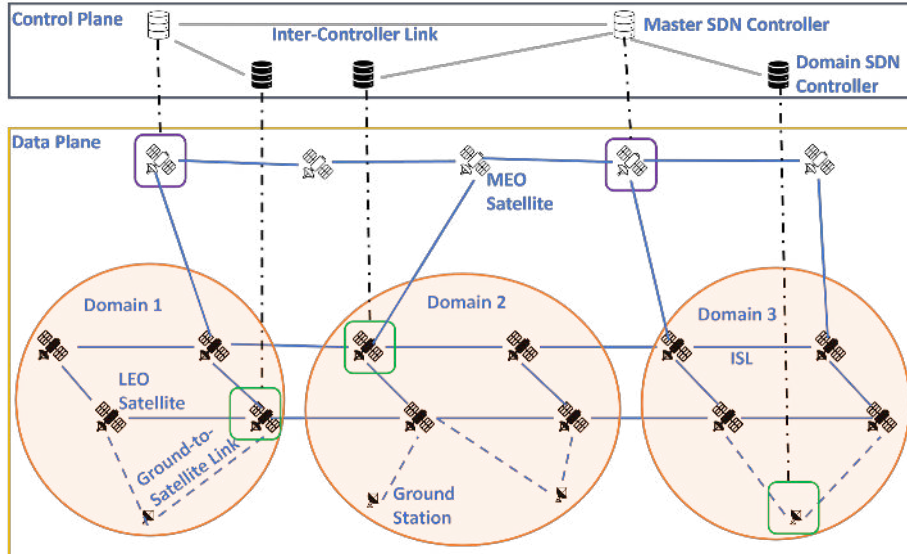


Figure 3-35: CoMOSat - System architecture.

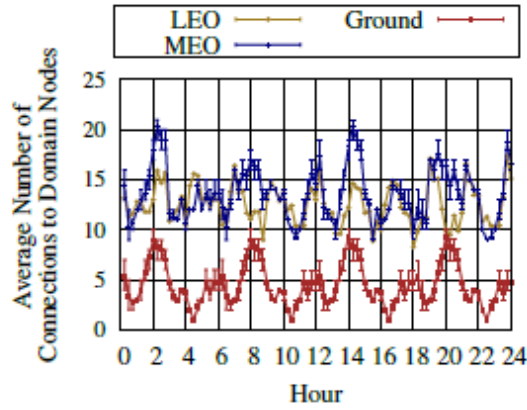


Figure 3-36: CoMOSat - Number of connections to domain nodes.

For this work, NTN is modelled as a time-varying graph $G[t] = (\mathcal{N}[t], \mathcal{E}[t])$, where $\mathcal{N}[t]$ is the set of nodes (satellite and ground) and $\mathcal{E}[t]$ is the set of available links at time slot t . Let $\tau_{ab}[t]$ denote the link stability, defined as the number of additional time slots (including the current slot t) during which the link $e_{ab} \in \mathcal{E}[t]$ remains connected with a data rate exceeding a predefined threshold. As shown in Figure 3-35, a multi-orbit satellite constellation consisting is considered of both LEO and MEO satellites. Let $\mathcal{S}_M[t] \subset \mathcal{N}[t]$, $\mathcal{S}_L[t] \subset \mathcal{N}[t]$, and $\mathcal{S}_g[t] \subset \mathcal{N}[t]$, denote the sets of MEO satellites, LEO satellites, and ground nodes at time slot t , respectively. Each node $n_i \in \mathcal{N}[t]$ serves as SDN switch and stores the forwarding rules for traffic flows. Therefore, $G[t]$ constitutes the data plane. Furthermore, the LEO satellites and ground nodes are classified into g groups termed ‘domains’. Let $\mathcal{D}[t]$ denote the set of domains at time slot t . This work proposes an incremental scheme to update the domains based on topology changes. Selected nodes in $\mathcal{N}[t]$ serve as SDN controllers and form the control plane. For this work, a hierarchical control plane is assumed where the controllers are divided into two groups: (1) domain SDN controllers and (2) master SDN controllers. A domain SDN controller is placed at an LEO satellite or ground node and manages the rule space of SDN switches in the

respective domain. It is also assumed that a domain controller cannot serve as the controller for an MEO satellite. Each domain has only one domain controller. The domain controllers share control information with master controllers placed at selected MEO satellites. The decision to place master controllers exclusively on MEO satellites is due to their superior connectivity within a multi-orbit satellite network, as shown in Figure 3-36. MEO satellites demonstrate significantly higher connectivity to domain nodes compared to ground nodes. For this example, an equal number is considered of Starlink LEO satellites, SES O3b MEO satellites, and ground nodes (20 of each). In contrast, ground nodes exhibit limited connectivity, making them less suitable for hosting master controllers. Each master controller manages and coordinates a set of domain controllers, ensuring scalability, global policy enforcement, and seamless communication across different network segments. A master controller also manages forwarding rules in nearby MEO satellites, which serve as SDN switches. Such a hierarchical control plane promotes scalability as the master controllers define global policies and facilitate coordination across multiple domains. In contrast, the domain controllers handle local management tasks, ensuring efficient network operation and growth. The SDN controller placement is denoted by:

$$x_a[t] = 1 \text{ if } n_a \in \mathcal{N}[t] \text{ serves as a controller at } t, \quad (3-44)$$

= 0 otherwise.

The domain (master) controller-switch (domain controller) association is expressed by:

$$y_{ab}[t] = 1 \text{ if domain (master) controller placed at } n_a \in \mathcal{N}[t] \text{ manages switch (domain controller) placed at } n_b \in \mathcal{N}[t] \text{ at } t, \quad (3-45)$$

= 0 otherwise.

Let $s_{ak}[t] = 1$ denote that domain d_k is allocated to n_a . On receiving an incoming traffic flow, a node serving as an SDN switch transmits a Packet-In message to the associated controller if no matching forwarding rule is present at the node. In reply, the controller installs the required flow rule at the switch. an in-band control plane is assumed where the same ISL or ground-to-satellite link transmits data and control traffic. The data traffic consists of traffic flows routed over the satellite network. On the other hand, control traffic consists of the Packet-In messages and the flow-rule installation messages.

3.6.3.1 Link Budget Model

Each $e_{ab} \in \mathcal{E}[t]$ has an available data rate $r_{ab}[t]$. The data rate of an ISL varies due to thermal noise and free-space path loss. Considering these effects, the channel gain of an ISL between two nodes, $n_a, n_b \in \mathcal{S}_M[t] \cup \mathcal{S}_L[t]$ is,

$$g_{ab}[t] = \frac{c\sqrt{G_{ab}}}{4\pi l_{ab}[t]f_{ab}\sqrt{\zeta K_B B_{ab}}}, \quad (3-46)$$

where G_{ab} , f_{ab} , B_{ab} , and $l_{ab}[t]$, are the fixed antenna gain, the center frequency, the channel bandwidth, and the distance of $e_{ab} \in \mathcal{E}[t]$, respectively. In addition, c is the speed of light, ζ is the thermal noise in Kelvin, and K_B is the Boltzmann constant. Therefore, the available data rate of the ISL is:

$$r_{ab}[t] = B_{ab} \log_2(1 + P_a |g_{ab}[t]|^2), \quad (3-47)$$

where P_a is the transmission power of n_a .

The ground-to-MEO/LEO link is represented using a line-of-sight (LoS) channel model. The channel gain from a ground node n_a to a satellite node n_b is given by:

$$g_{ab}[t] = \sqrt{(l_{ab}[t])^{-\rho}} e^{-j\frac{2\pi f_{ab}}{c} l_{ab}[t]}, \quad (3-48)$$

where ρ is the path loss exponent. Therefore, the data rate from $n_b \in \mathcal{S}_G[t]$ to $n_a \in \mathcal{S}_M[t] \cup \mathcal{S}_L[t]$ is expressed as:

$$r_{ba}[t] = B_{ba} \log_2 \left(1 + \frac{p_b G_{ba} |g_{ba}[t]|^2}{\sigma_0^2 B_{ba}} \right), \quad (3-49)$$

where σ_0^2 is the Gaussian noise power. Similarly, the data rate from $n_a \in \mathcal{S}_M[t] \cup \mathcal{S}_L[t]$ to $n_b \in \mathcal{S}_G[t]$ is expressed as:

$$r_{ab}[t] = B_{ab} \log_2 \left(1 + \frac{p_a G_{ab} |g_{ab}[t]|^2}{\sigma_0^2 B_{ab}} \right). \quad (3-50)$$

3.6.3.2 Traffic Model

The load of each controller depends on the Packet-In messages generated by the associated switches. The number of Packet-In messages a satellite switch generates depends on its traffic load, which varies with the coverage area. At a time slot t , the coverage area for a satellite $n_b \in \mathcal{N}[t]$ at height $h_b[t]$ with minimum elevation angle θ_{min} is $S_b[t] = 2\pi R^2(1 - \cos \psi_b)$, where R is the radius of Earth in kilometres and $\psi_b = \arccos \left(\frac{R}{R+h_b[t]} \cos \theta_{min} \right) - \theta_{min}$. Let d be the population density in the covered area, i.e., people per square kilometre. Therefore, the traffic demand in the covered area is $\sigma_b[t] = S_b[t]d$, where ϵ is the packet arrival rate at n_b for the traffic generated by a single user. The work assumed that a user request is sent to a single satellite even if the user is in the coverage area of multiple satellites. Let p be the probability of generating a new request by a user, which is handled by an SDN switch n_b . In this case, n_b sends a packet-in message to its associated controller. Hence, the arrival rate of Packet-In messages at a controller n_a is given by:

$$\lambda_a[t] = \sum_{n_b \in \mathcal{S}} y_{ab}[t] p \sigma_b[t]. \quad (3-51)$$

3.6.3.3 Cost Model

The cost of placing an SDN controller in a multi-orbit satellite network has three components: flow setup, switch remapping, and master controller synchronisation.

1. **Flow Setup Cost:** Flow setup cost refers to the flow setup delay or the time taken to establish a new flow or modify an existing flow in the network. When a packet arrives at a switch and there is no pre-existing flow entry in the flow table to handle it, the switch sends a Packet-In message with the packet metadata to the controller. The controller then computes the appropriate actions for the packet and installs a flow entry into the switch's flow table. The time it takes from the Packet-In message being sent to the flow entry installed is the flow setup delay. Therefore, the flow setup delay is the sum of transmission,

propagation, and queuing delay at the controller. The values of these delay components are computed considering a switch n_b managed by controller n_a .

- **Propagation and Transmission Delay:** This refers to the latency of transmitting a Packet-In message from n_b to n_a and transmitting a FlowMod message from n_a to n_b . The total propagation and transmission delay is estimated as:

$$\delta_{ab}^{ptd}[t] = 2 \frac{l_{ab}[t]}{c} + \frac{m}{r_{ba}[t]} + \frac{q}{r_{ab}[t]}, \quad (3-52)$$

where m is the size of a Packet-In message and $q \geq m$ is the size of the FlowMod message.

- **Queuing Delay at the Controller:** This is the delay experienced by Packet-In requests as they wait in a queue to be processed by the controller. This delay occurs when multiple Packet-In messages arrive at the controller simultaneously and must be processed sequentially. At t , the total queuing delay at the controllers is:

$$\delta_{ab}^{que}[t] = \frac{\lambda_b[t]}{\mu(\mu - \lambda_b[t])}, \quad (3-53)$$

where μ is the service rate at a controller. Each control traffic is assumed as an independent and individual Poisson process and the M/M/1 queuing model is considered.

Therefore, the total flow setup delay for installing a flow rule at switch n_b by controller n_a is $\delta_{ab}^{fsd}[t] = \delta_{ab}^{ptd}[t] + \delta_{ab}^{que}[t]$. Accordingly, the total flow setup cost at t is defined as follows:

$$Q^{setup} = \frac{1}{\sum_{n_a, n_b \in \mathcal{N}[t]} \delta_{ab}^{fsd}[t]} \sum_{n_a, n_b \in \mathcal{N}[t]} x_a[t] y_{ab}[t] \delta_{ab}^{fsd}[t]. \quad (3-54)$$

The cost of placing an SDN controller in a multi-orbit satellite network has three components: flow setup, switch remapping, and master controller synchronisation.

2. **Switch Remapping Cost:** Existing domain (master) controller-switch (domain controller) mapping should be changed when the respective nodes move out of communication range due to the movement of satellites. The current controller sends the new controller's information to the switch for remapping. The switch sends a connection request to the new controller, which confirms the request upon receipt. Therefore, the node's remapping depends on the respective path's stability, i.e., the number of time slots in which the links are connected. Accordingly, the remapping cost at t is defined as:

$$Q^{remap}[t] = - \frac{\sum_{n_a, n_b \in \mathcal{N}[t]} y_{ab}[t] \min_{e_{ij} \in \text{path}_{ab}[t]} \text{stability}(e_{ij})}{\sum_{n_a, n_b \in \mathcal{N}[t]} \text{stability}(e_{ab})}, \quad (3-55)$$

where $\text{path}_{ab}[t]$ denotes the set of links in the shortest path from n_b to its controller n_a , and $\text{stability}(e_{ij})$ denotes the number of time-slots a link e_{ij} remains connected.

3. **Master Controller Synchronisation Cost:** Each master controller receives a domain network view from the associated domain controllers. So, the master controllers synchronise regularly to have a global network view. This cost refers to the transmission latency required to synchronise all master controllers placed at selected MEO nodes. Formally, the master controller synchronisation cost is:

$$Q^{syn}[t] = \frac{\sum_{n_a, n_b \in \mathcal{N}_{\mathcal{M}}} x_a[t] x_b[t] \frac{l_{ab}[t]}{c}}{\sum_{n_a, n_b \in \mathcal{N}_{\mathcal{M}}} \frac{l_{ab}[t]}{c}}. \quad (3-56)$$

3.6.4 Problem description and formulation

This work aims to place as few SDN controllers as possible in the multi-orbit satellite network at a minimum cost. Accordingly, a multi-objective optimisation problem is formulated to minimise the number of controllers and the controller placement cost. The objective function is defined as:

$$\text{Minimise } \psi X[t] + (1 - \psi) Q[t] \quad (3-57)$$

$$x[t], y[t]$$

subject to

$$C1 : \sum_{n_a \in \mathcal{N}[t]} y_{ab}[t] = 1, \forall n_b \in \mathcal{N}[t], \quad (3-58)$$

$$C2 : x_a[t] \leq \sum_{n_b \in \mathcal{N}[t] \setminus n_a} y_{ab}[t], \forall n_a \in \mathcal{N}[t], \quad (3-59)$$

$$C3 : \sum_{n_a \in \mathcal{N}[t]} s_{a,k}[t] x_a[t] = 1, \forall d_k \in \mathcal{D}[t], \quad (3-60)$$

$$C4 : \sum_{n_a \in \mathcal{N}[t]} x_a[t] \leq |\mathcal{N}[t]|, \quad (3-61)$$

$$C5 : x_a[t], y_{ab}[t] \in \{0,1\}, \forall n_a, n_b \in \mathcal{N}[t], \quad (3-62)$$

where $X[t]$ is the controller count, $Q[t] = (Q^{\text{setup}}[t] + Q^{\text{remap}}[t] + Q^{\text{syn}}[t])$, and ψ is the weightage variable that indicates the relative importance of controller count and placement cost. Constraint C1 states that each node is assigned to a single controller. Constraint C2 assures that if a domain (master) controller is not deployed, it does not control any switches (domain controllers). If a domain (master) controller is deployed, it must manage at least one switch (domain controller). Constraint C3 states that precisely one controller exists in each domain. Constraint C4 indicates that the controller count cannot exceed the number of nodes in the NTN topology. Constraint C5 states the binary decision variables.

3.6.5 The proposed scheme

This section outlines the proposed strategy for hierarchical controller placement in a multi-orbit satellite network. Initially, LEO satellites and ground nodes are grouped into multiple domains using the k-means clustering algorithm, with the optimal number of domains determined via the elbow method. A GA model is designed to select the domain and master controllers. An incremental update strategy is introduced to update domains and controller placements, ensuring minimal remapping between controllers and the managed nodes.

3.6.5.1 GA for Domain and Master Controller Placement

GA is used to place master and domain controllers, considering the domain classification. GA is suitable for the CPP in satellite networks because GA is efficient for large solution spaces and dynamic environments, like the network model considered in this work.

Algorithm 1 Genetic Algorithm-Based Controller Placement

- 1: **Input:** $G[t]$, population size PS, max generations GenMax, crossover probability P_c , mutation probability P_m , convergence tolerance ϵ_{GA}
- 2: **Output:** Optimal controller placement x, y
- 3: Initialise population, set $g \leftarrow 0$, converged $\leftarrow$ False, compute initial fitness fit_{cur} based on random placement.
- 4: **while** $g \leq \text{GenMax}$ **and** $\neg \text{converged}$ **do**
- 5: Select parents based on fitness, apply crossover based on P_c , and mutate offspring based on P_m .
- 6: Evaluate offspring, update best solution ω^* and best fitness fit_{best} .
- 7: **if** $|\text{fit}_{\text{cur}} - \text{fit}_{\text{best}}| < \epsilon_{GA}$ **then** converged $\leftarrow$ True.
- 8: **end if**
- 9: Increment g , update $\text{fit}_{\text{cur}} \leftarrow \text{fit}_{\text{best}}$.
- 10: **end while**
- 11: Form and return x, y based on ω^* .

Algorithm 1 shows the steps for GA-based domain and master controller placement. In the GA model, the chromosome is represented as a one-dimensional array of size $2D$, where D is the total number of domains that the k-means clustering algorithms identify at $t = 1$. The chromosome is a sequence of genes and is structured as $[DC_1, DC_2, \dots, DC_D, MC_1, MC_2, \dots, MC_D]$. Here, DC_k denotes the node (LEO or ground) ID acting as the domain controller for domain d_k , while MC_k represents the node (MEO) ID serving as the master controller for the same domain. If $DC_k = a$ at time t , this implies that $x_a[t] = 1$ and $y_{ab}[t] = 1$ for all nodes n_b satisfying $s_{bk} = 1$. Similarly, if $MC_k = c$ at time t , then $x_c[t] = 1$ and $y_{ca}[t] = 1$. Step 3 initialises a population of size PS by randomly selecting an LEO or ground node from each domain to serve as the domain controller and random MEO nodes as master controllers for each domain. The fitness function of an individual ω is negatively correlated to the objective function and is formulated as $\text{fitness}(\omega) = U - \psi X[t] + (1 - \psi)Q[t]$, where U is a large constant. The initial fitness fit_{cur} is estimated based on the random placement of master and domain controllers. Step 5 randomly selects two individuals from the population, designating the one with higher fitness as the first parent and the other as the second parent.

With probability P_c , a crossover operation is performed on the selected parents to produce two offspring. During the crossover, two points, $CP1$ and $CP2$, are chosen randomly from the set $\{1, 2, \dots, 2D\}$. To form the first offspring, genes from positions 1 to $CP1 - 1$ are taken from the first parent, genes from $CP1$ to $CP2$ are taken from the second parent, and genes from $CP2 + 1$ to $2D$ are taken from the first parent. Similarly, for the second offspring, genes from positions 1 to $CP1 - 1$ are taken from the second parent, genes from $CP1$ to $CP2$ are taken from the first parent, and genes from $CP2 + 1$ to $2D$ are taken from the second parent. Each offspring

undergoes mutation with probability P_m . A random position MP is selected from the set $\{1, 2, \dots, 2D\}$ or mutation. If $MP \leq D$, the selected position corresponds to a domain controller. The domain associated with this controller is identified, and the gene at MP is replaced with the ID of a randomly selected node from the same domain. If $MP > D$, the selected position corresponds to a master controller, and the gene at MP is replaced with the ID of a randomly selected MEO. Step 6 evaluates the offspring and selects the best individual, ω^* , with a fitness value of fit_{best} . The algorithm converges if the difference between fit_{best} and the current fitness fit_{cur} is within the convergence tolerance ϵ_{GA} (Step 7). Otherwise, the current generation g is incremented, and the best fitness is stored as the current fitness in Step 9. The GA algorithm proceeds for up to GenMax generations or until convergence. Ultimately, the controller placement is obtained in the form of x and y , representing the optimal solution determined by the algorithm.

3.6.5.2 Incremental Update of Domains and Controller Placement

For domain formation, k-means clustering is performed at $t = 1$. For each $t > 1$, if the distance between a node n_a (where $s_{ak}[t] = 1$) and the cluster centroid of d_k exceeds a threshold β , the cluster centroid nearest to n_a is identified, and n_a is reassigned to the respective domain. The threshold β is set based on the constellation parameters and the average inter-node distance. The controller placement is updated when the number of nodes that change domains exceeds a threshold γ , defined based on network size and traffic dynamics.

Initially, the CPP is solved offline, and the optimal controller locations and associations are proactively distributed to all nodes in $\mathcal{N}[t]$ before deployment. At the start of each subsequent time slot, the existing configuration is refined following the incremental update strategy. The current controllers then disseminate the updated association information to the associated nodes, and mappings are changed only when necessary.

3.6.6 Performance evaluation

3.6.6.1 Simulation Settings

Table 3-8: CoMOSat - Simulation Parameters.

Parameter	Value
Time slot duration	15 minutes
Number of time slots	40
Minimum elevation angle	10°
EIRP plus receiver antenna gain	3.74 W
Carrier frequency	20 GHz
Channel bandwidth	20 MHz
Thermal noise	354.81 Kelvin
Transmission power	100 W (satellite), 2 W (ground node)
Path loss exponent	4
Gaussian noise power	-174 dBm/Hz
Additional parameters	$\psi = 0.5$, $\beta = 10$ km, $\gamma = 10\%$ of $\{S_L[t] \cup S_G[t]\}$

GA parameters	$P_S = 20$, $GenMax = 10$, $P_c = 0.8$, $P_m = 0.1$, $\epsilon_{GA} = 0.1$
---------------	--

A multi-orbit scenario is considered involving 20 MEO satellites from the SES O3b MEO constellation and 20-80 uniformly distributed LEO satellites from the Starlink constellation. Additionally, 20 uniformly distributed ground nodes based on Starlink gateway locations are considered. The scenario is simulated using MATLAB Satellite Communication Toolbox, and the two-line element (TLE) data of 19th October 2024. The scenario considers a real-world population density for each coverage area based on the covering satellite or ground node location. Table 3-8 states the simulation parameters.

3.6.6.2 Benchmark Schemes

The performance of CoMOSat is compared with two benchmark schemes: (1) fixed placement (FP) and (2) static placement with dynamic assignment (SPDA) [54]. In the FP scheme, controllers are placed randomly, and the placement remains fixed. Like CoMOSat, FP uses k-means clustering to partition domains. Also, random MEO nodes are selected as master controllers for each domain in FP. SPDA, on the other hand, statically places controllers on selected satellites to minimise the average and maximum switch-controller latencies while dynamically reassigning switches to controllers to adapt to variations in network topology and traffic load. FP places controllers at fixed locations without reassignment, while SPDA allows dynamic switch-to-controller reassignment but keeps controller locations static. Unlike CoMOSat, neither scheme optimises controller count or placement dynamically.

3.6.6.3 Performance Metrics

Average number of controllers: This metric represents the average number of controllers, including domain and master controllers, over the entire simulation duration.

Average flow setup delay: This metric measures controller-switch pairs' average flow setup delay throughout the simulation duration.

Average number of remaps: This metric captures the changes for domain controller-to-switch and master-to-domain controller mappings.

3.6.6.4 Result and Discussion

1) Average number of controllers:

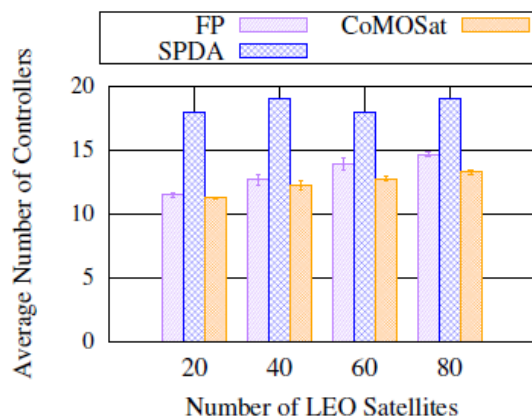


Figure 3-37: CoMOSat - Controller Count.

Figure 3-37 shows that the average controller count for CoMOSat is 6.17% and 33.13% less than that for FP and SPDA, respectively. This is because CoMOSat optimally minimises the number of domain controllers by determining the optimal number of domains and adjusting the number of master controllers based on the fitness function. It is important to note that the controller count for FP is comparable to CoMOSat, as both schemes depend on the number of domains, which are assumed to be the same in the simulations. The key difference lies in CoMOSat's optimisation of master controller placement, which leads to a larger performance gap as the constellation size increases. SPDA determines controller placement for a given fixed number of controllers but does not dynamically optimise the total controller count.

2) Average flow setup delay:

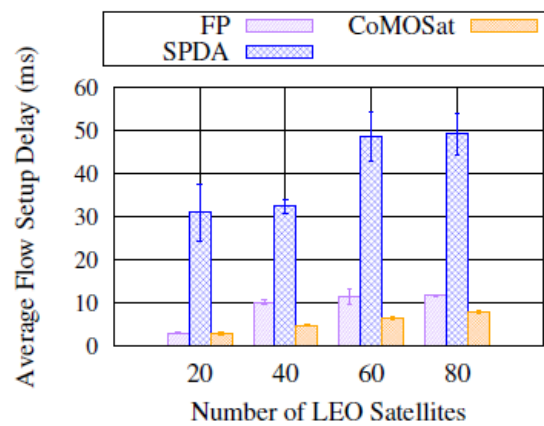


Figure 3-38: CoMOSat - Flow setup delay.

From Figure 3-38, it is observed that the average flow setup delay for CoMOSat is 42.85% and 87.06% less than that for FP and SPDA, respectively. This is because CoMOSat dynamically selects and updates controller locations to minimise the controller placement cost $Q[t]$, which includes the total flow setup cost Q^{setup} . In contrast, FP randomly selects the controller locations and does not update the control plane if the flow setup delay increases. While SPDA places controllers statically, it only reassigns switches dynamically without adapting the controller locations, limiting its flexibility in reducing flow setup delays under varying network loads.

3) Average number of remaps:

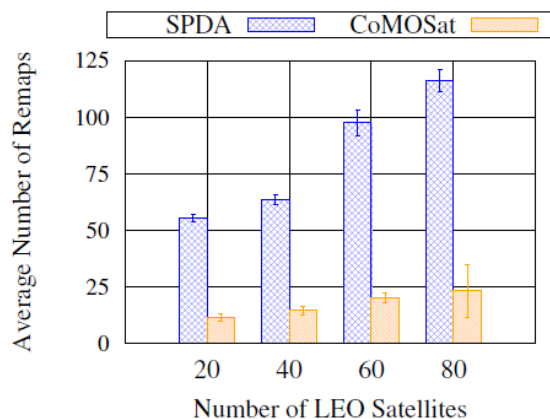


Figure 3-39: CoMOSat - Average number of remaps with varying LEO count.

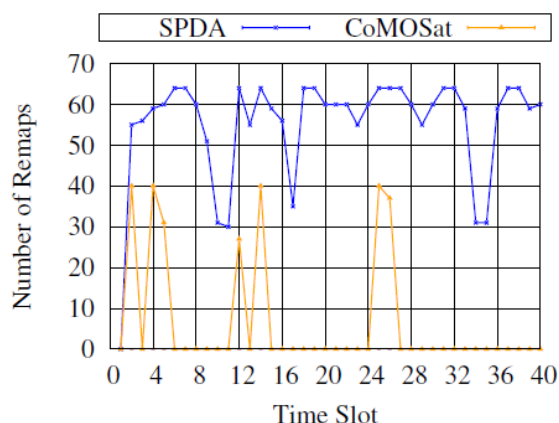


Figure 3-40: CoMOSat - Average number of remaps across time slots.

Figure 3-39 and Figure 3-40 compare the remap count for CoMOSat and SPDA with varying numbers of satellites and across time slots with 20 LEOs. Figure 3-39 shows that CoMOSat performs significantly fewer remaps than SPDA, as CoMOSat prioritises stable controller placement and controller-switch mappings. Additionally, Figure 3-40 shows that CoMOSat's remap frequency is lower and stabilises after the initial few slots. CoMOSat only rearranges the control plane when more than γ nodes change domains, reducing the reconfiguration overhead.

3.7 SOFTWARE DEFINED NETWORK CONTROLLER FOR THE 3D ETHER ARCHITECTURE

The integration between TN and NTN networks is a crucial step for communications, as 6G applications envision hyper-connected environments, that merge human involvement and intelligent systems for an integrated society. The ETHER architecture developed in D2.4 [6] crucially integrates Edge, Cloud, RAN and Transport Network domains with NTN solutions, such as UAVs, HAPS, and satellites, especially non-geostationary such as medium and low earth orbit (MEO/LEO) ones. Expanding the current telecommunications network to involve non-geostationary satellites aims to provide better services such as TV broadcasting, weather forecasting or remote sensing, unlocking the potential for better and new revenue opportunities for network operators.

The developments of the ETHER project focused both on developing new mechanisms that take advantage of cutting-edge developments in TN and NTN technologies for the 6G network infrastructure and on the necessary integration components to achieve a complete solution for the 6G network. This complete solution needs to be fully software programmable, built with zero-touch automation support mechanisms and capable of achieving low latency and better energy efficiency than traditional networks. In doing so, ETHER is aligned with 6G European values and allows for the creation of new business models that take advantage of this potential ecosystem.

3.7.1 Background

Edge computing is a paradigm that takes advantage of computational and storage resources located close to the end-user, such as cameras, sensors or edge servers, and using them to run network services and applications. These devices, usually located closer to network use-

case applications, are capable of achieving lower latency values, which in turn enables better support of real-time applications. One of the ETHER pillars is the support of multi-layered distributed edge computing and caching, offloading large computational tasks on multiple devices closer to where data is produced, easing the strain on the network.

Cloud computing is a solution that works opposite to edge computing: by centralising compute and storage power, in public or private clouds, these clusters can gather a large pool of resources that the network, in moments of strain, offload heavy tasks, such as specific network functions, AI/ML models and service orchestration. By using a virtualisation approach, physical hardware is divided into standardised virtual units, simplifying the orchestration of these virtual machines and enhancing the flexibility of the deployed applications and services. The ETHER project envisions the integration of both edge and cloud capabilities into their integrated TN-NTN 6G architecture, enabling the pre-processing and storage of large sets of information, while reducing the overall latency of the network and improving energy efficiency through intelligent offloading algorithms.

The RAN connects the wide array of IoT and UE devices with satellites and the 5GCN. Innovations such as Flexible Payloads (T-4) and Improved Data Analytics (T-5) (outlined in ETHER Deliverable D3.1 [7]) combine advancements in the orchestration of network systems and satellite payloads with the distributed nature of devices and edge computing for energy efficiency gains and E2E latency reduction without impacting transmission of information. A scenario like Demo 1 [8] ground stations, assisting with service orchestration and context sharing for the running applications. In the background, connecting the RAN to the CN is the Transport Network, the backbone infrastructure that carries data, connecting network and service providers.

3.7.2 Motivation and Contribution

The development of these infrastructures is not just focused on TN and NTN solutions, but on the progress of 6G networks as a whole; in doing so, the ETHER project needs to tackle some issues relevant to the integration of Edge, Cloud and RAN technologies with NTN platforms.

Some open issues regarding edge computing concern the dynamic placement of satellites, given the frequent changes of constellation topology and frequent inter-satellite connection switching and routing design required to preserve the seamless nature of 6G networks [55] [56] [57].

Both cloud computing and NTN systems are quite recent developments in the context of telecommunications. While 5G standardisation efforts have been going on for about a decade, the integration of both of these technologies has only recently gained traction; this is even more important for future networks, given that 6G envisions applications and services that require lower latencies and higher bandwidth, they also require seamless connection across a global service coverage, according to the Ubiquitous Connectivity scenario in IMT-2030 Framework [58] [59].

Including satellites in the network also requires some adaptation from RAN and the Transport Network: ISLs and feeder links need to support the backhaul of the network, to connect satellites and ground stations; in RAN, including satellite and other airborne platforms in their software stacks is necessary to enable access to UEs and the CN. Topology awareness can be used to optimise path selection for better routing latency [60] [61].

SDN is a network management approach, that uses software to configure control networks, rather than relying solely on hardware-based devices. By decoupling the control plane and the data plane from its hardware, SDN enables centralised control and programmability through

an SDN controller; this SDN controller is tasked with managing the entire network, handling instructions previously allocated to the hardware components. By leveraging this softwarisation of network management, programmability can also be enhanced, as automation mechanisms can be implemented, making it easier to manage and maintain the network.

This proposal leverages the SDN approach, introducing a Software Defined Network Controller for the 3D ETHER environment. The main goal of this SDNC is to provide a secure, scalable, and intelligent environment for managing the TN and NTN infrastructure, unifying network control across the Ground, Air and Satellite domains. This proposal enables efficient data routing among ETHER infrastructures, allowing for a better structured approach to edge and cloud offloading across the network, but also secure communications between devices and the network, facilitating onboarding and lifecycle control.

3.7.3 System Model and Components

The proposed model for this SDNC, represented in Figure 3-41, serves as the centralised management unit for the network. To simplify management of the different network domains that are envisioned to be interfacing: the Ground Controller, the Air Controller and the Satellite Controller.

The Ground Controller handles the traditional terrestrial infrastructure of the network, such as the RAN infrastructure, Transport Network backhaul, Edge servers and IoT devices. Through this controller, it is possible for the ETHER network to detect terrestrial nodes and establish links between TN devices, as well as providing real-time information on the topology of the network. Specific traffic flow routing mechanisms can be implemented, in order to allow for the optimisation of applications according to latency and bandwidth requirements, or prioritising specific service communications. Handover between TN slices is also supported, allocating resources according to service requirements, as well as managing the lifecycle of these slices, monitoring link status and providing fault detection mechanisms.

The Air Controller handles the intermediate devices that are located at mid-altitude in the 3D ETHER space, such as UAVs and HAPS. It allows for communication traffic that links them with the network, enabling dynamic mobility positioning and exposing telemetry information to aid aerial flight control and network topology updates. This controller also interfaces between ground and satellite nodes, monitoring and predicting link quality, adjusting routing based on availability to enhance QoS of network services. This interfacing is also essential in order to optimise paths between device, enabling or disabling network relays to maintain global coverage while maximising network efficiency.

Finally, the Satellite Controller manages air-to-air links, between Satellites and mid-altitude devices, as well as air-to-ground links, between Satellites and ground stations or UEs. The network should be capable of tracking satellite location, using this information to predict link quality and availability, and distributing necessary network traffic among the available satellites according to service requirements; in this way, satellite paths can be used to free up terrestrial communications when severely congested. Topology data of Satellites should also be used to manage Inter-satellite communications.

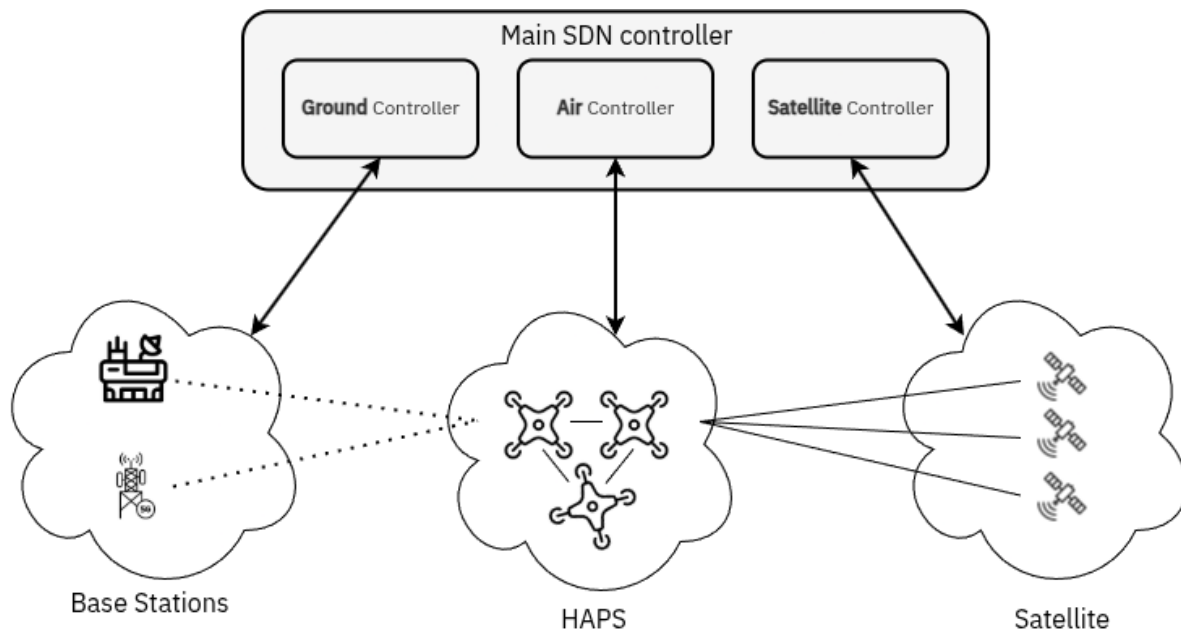


Figure 3-41: SDN controller mapping to the 3D ETHER network.

Together, all three ETHER domain controllers' functions are centralised in a single module, enabling the seamless integration of TN-NTN devices. In doing so, the network has access to a unified global view of the network topology to compute the optimal E2E path routing. SLAs, cost policies and service requirements are also centralised, which allows for dynamic resource and link allocation from the network, as well as mechanisms that automate traffic between TN and NTN devices. Finally, this SDN should be deployed with a unified trust framework in mind, using AAA and CIA mechanisms across the network, for secure device communications.

3.7.4 Communication Flows

To further illustrate the functionalities of this SDN controller, below are 3 essential processes outlined from device communications with the network, using all 3 modules (Ground, Air and Satellite) as interfaces with the SDNC of the ETHER 3D network. The first one outlines device registration on the network, the second how data flows between devices and the SDN controller are managed, and finally, a scenario for a beneficial communication handover process.

Device Onboarding

For this process, a connection to the nearest access point (Ground, Air or Satellite) is established to start the onboarding process to the ETHER network, registering, authenticating and configuring the device. This onboarding mechanism will also register devices' computational and storage capabilities; after validating this information, the best access module for device communication with the network is determined (Ground, Air or Satellite), and this decision is configured and applied according to device specifics. The SDNC will send this configuration so that the appropriate access module can finalise the registration process with the device, an establish a data link between the device and the network. This process is described in Figure 3-42.

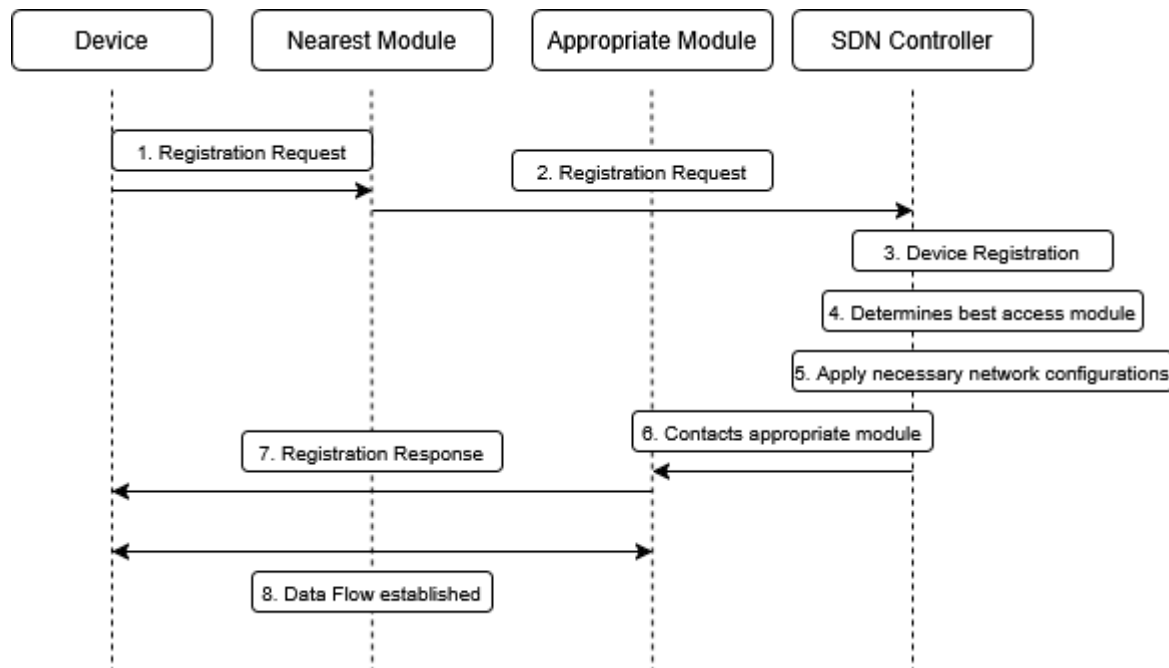


Figure 3-42: Device onboarding flow diagram.

Data Flows

Application data can be optimised between TN and NTN links, based on network state, latency, policies and device profile. When devices initiate data transmissions, the designated access point detects network traffic, analysing the request and passing it to the SDNC. The SDNC will then collect real-time metrics regarding latency values, congestion and energy use of this transmission; using this information, it will evaluate them and enforce current network routing policies, deciding best end-to-end path for that specific transmission, delivering the data to the target according to this decision. This data link can dynamically switch between TN and NTN according to conditions of network communications.

Communication Handover

In this scenario, a device location triggers a service handover from the ground module to air module coverage, due to a trigger on link quality. The device is registered on the ground module, so this module informs the SDNC of the signal degradation, setting off an evaluation of best candidate paths for device communications; after a result is achieved, for instance, towards the air module, the SDN triggers the vertical handover process towards the device, configuring it to establish a data flow, now with the air module. This process is described in Figure 3-43.

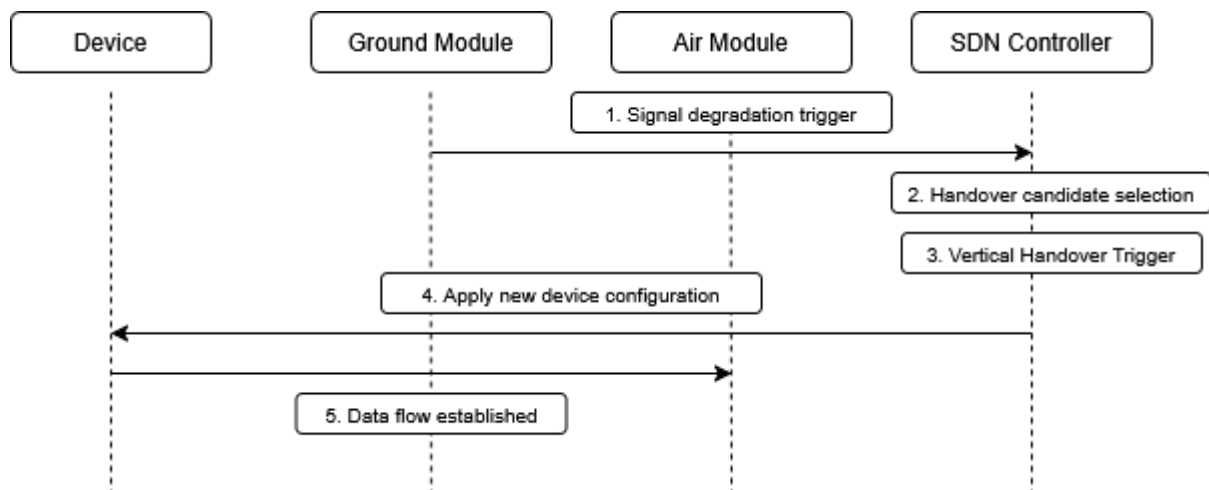


Figure 3-43: Communication handover flow diagram.

3.7.5 Implementation details and testing

For this implementation, our SDNC implementation is based on ONOS (Open Network Operating System) [62].

- **Scalability and Auto-discovery**, as ONOS is built for managing and controlling networks with thousands of devices, encompassing the whole ETHER environment through multiple controller instances, making sure the network has high availability and fault tolerance, while also enabling discovery of new network devices, updating the network topology in real-time.
- **Programmability and Flexibility**, given the open-source nature of ONOS, full customisation is available through tailor-made applications and modules that enable specific link routing, dynamic topology handling and critical path optimisation. Specific high-level intents can be defined by operators and translated by ONOS to simplify and improve dynamic multi-domain and multi-layer orchestration through intent-based management.
- **Real-Time Control**, by enacting changes on the ONOS topology and controllers in real-time, allowing for operators to affect the network directly, or programming reactive triggers that optimise network flows in real-time.
- **Modular Architecture**, by including not only over 175 applications for monitoring, managing and steering traffic, but also support for custom application development, like handover logic, or policy-based QoS
- **Multi-protocol and Multi-Domain**, integrating heterogeneous communication technologies through standardised southbound protocols (OpenFlow, NETCONF, gNMI) as well as custom proprietary APIs and southbound plugins for specialised equipment. Besides physical networks, ONOS also integrates well with industry standard orchestration frameworks (Kubernetes, OpenStack and NFV MANO) and with simulators/emulators (Mininet, CORE) for testing purposes before deployment.

Important ONOS applications that are being taken advantage of are **Reactive Forwarding**, which enables the automatic installation of flows in response to packet-in events, as well as providing on-demand connectivity without prior configuration, to facilitate the integration of new network devices on the network topology; **Host Location Provider**, which helps discover and store information about the location of hosts on the network; and **OpenFlow Provider**, acting as the main interface between the ONOS controller and OpenFlow switches, as well as translating high-level commands into OpenFlow instructions.

Current setup topology simulates a 3D network, including the Ground, Air and Space domains. There are five **OpenFlow switches** across these three domains, with four **hosts** distributed in the ground, with a **primary link** between the ground switches, and **redundant paths** across the other domains. The scenario tested was focused on maintaining link availability between two hosts, through the transmission of ICMP pings, observed in real-time through Wireshark [63] pings between hosts are working as normal through a direct terrestrial link; in the **second phase**, this link is disabled, simulating a critical failure, and ONOS Reactive Forwarding recalculates the optimal link path automatically, using this backup for communication between the two hosts ; in the **third phase**, the link is restored, and traffic automatically returns to the optimal network path. Testing setup is illustrated in the Figure 3-44

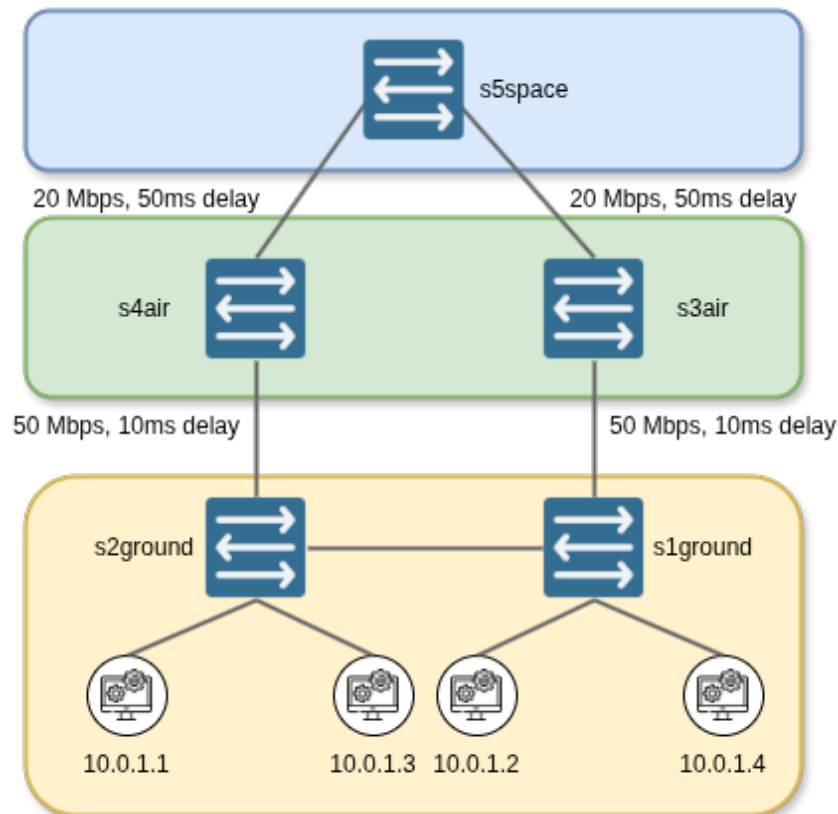


Figure 3-44: ONOS Implementation setup.

4 AI-ENABLED E2E NETWORK OPTIMISATION

4.1 INTRODUCTION

This chapter is focused on the third key development in WP4, which addresses AI-Enabled end-to-end network optimisation. This development is structured around four core activities: AI-Enabled 3D network performance and topology forecast (A3-1), Real-time energy-efficient resource allocation (A3-2), AI-based offline user association for the 3D sustainable ETHER network (A3-3) and AI-Driven Beamforming for Integrated Sensing and Communication in Terrestrial-Satellite Networks (A3-4). The capabilities highlighted in A3-1, A3-2 and A3-3 are critical for safety-related operations and effective airspace management, which are the primary objectives of UC 3. Contributions of A3-3 and A3-4 are aligned with the works done in UC2.

- **AI-Enabled 3D network performance and topology forecast:** this activity aims to develop network monitoring algorithms and prediction techniques for efficient network management, as well as a real-time active monitoring framework based on AI-enabled data analytics for ETHER network architecture. The outcomes of this monitoring framework will be then used in the optimisation procedures for the true end-to-end cross-layer optimisation techniques developed in activity A3-2.
- **Real-time energy-efficient resource allocation and routing:** this activity focuses on the joint network, computational and storage resource allocation problem so as to achieve high energy efficiency. In addition, aircraft message routing supported by AI-based data analytics (A3-1) and graph-based handover planning (A2-2) is optimised subject to service requirements and network dynamicity. This work is an extension of the graph-based handover method which designs a long-term and semi-static E2E handover plan for aircraft. As the 3D network dynamicity or resource is time-varying, that requires a real-time AI-enabled E2E network optimisation solution. Both online algorithms, which will offer self-configuration at run time, as well as optimal solutions, to be used as a benchmark are developed.
- **AI-based offline user association for the 3D sustainable ETHER network:** this activity leverages RL to associate the UEs in the 3D ETHER network in an energy efficient way. In particular, a low-complexity DNN trained with Proximal Policy Optimization is proposed, capable of associating UEs with the access network based on their service requirements, while minimizing network power consumption as well as avoiding violating the capacity constraints. A three-phase curriculum learning strategy was employed for the DNN to effectively learn the intricate relationships between PRB allocation and energy consumption. The proposed approach demonstrates significant improvements over current SotA methods, achieving significant improvements in energy saving (by reducing power consumption by up to 47.7%) while meeting user service requirements.
- **AI-Driven Beamforming for Integrated Sensing and Communication in Terrestrial-Satellite Networks:** this activity focuses on a cognitive satellite–terrestrial network system, aimed at enabling simultaneous target sensing and communication services for satellite users. The integration of RSMA addresses the dual-functionality challenge by supporting both high-throughput communication and sensing capabilities. A deep learning-based methodology is introduced to optimise the weighted sum rate of satellite users while ensuring a minimum beampattern gain for effective sensing and maintaining an interference threshold with terrestrial operations, which are underpinned by distributed MIMO networks. This approach formulates a joint optimisation problem encompassing precoding, power allocation, and common rate control. A hybrid solution, combining deep CNNs with the semidefinite relaxation technique is proposed to enhance system throughput under strict interference and sensing requirements. The RSMA-based strategy demonstrates superior

performance over conventional non-RSMA techniques, reinforcing its potential for next-generation integrated network architectures.

Activities A3-1, A3-2 and A3-4 will be integrated for the development of the unified ETHER MANO, while A3-3 in combination with the work performed at WP2 will be used on the ETHER network deployment phase. To that end, procedures to integrate the provisioning of network services with a consistent configuration of the RAN resources will be implemented. This will enable the full, end-to-end management of network slices, from the RAN to the core segments, for different kinds of 6G services and accounting for very diverse (terrestrial, aerial, satellite) radio access domains. The correspondence of the activities of this development and the MANO architecture is shown in Figure 4-1.

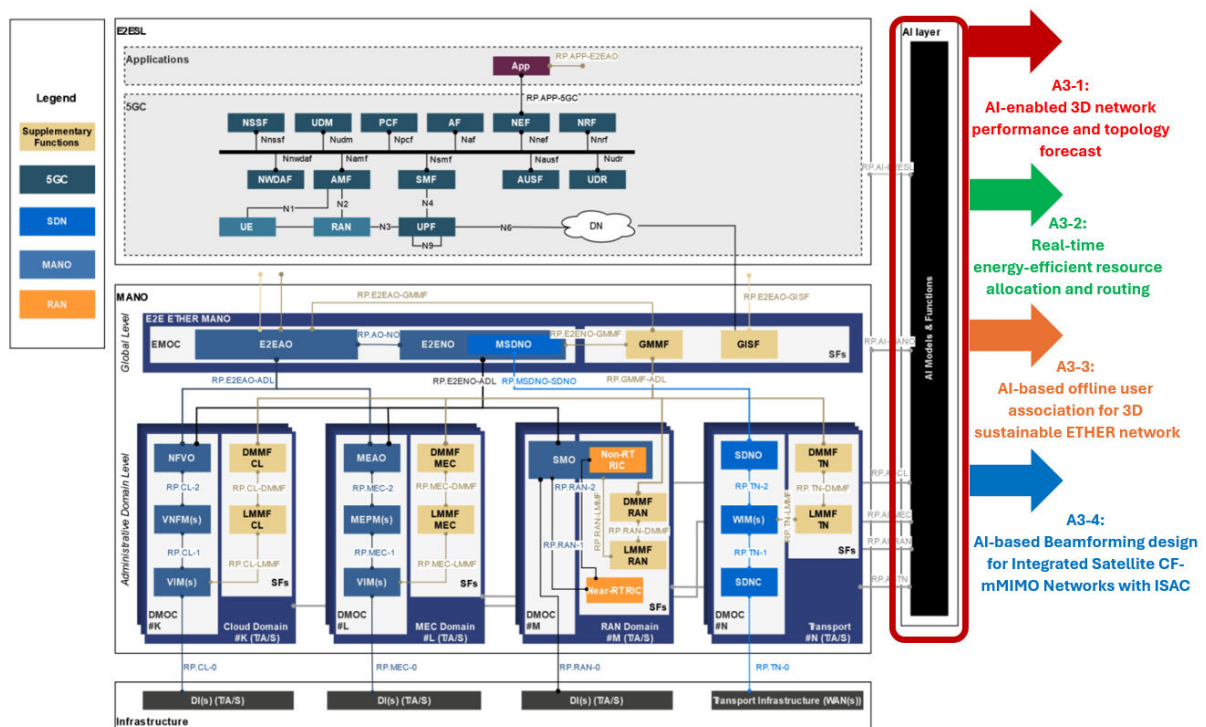


Figure 4-1: Correspondence of the activities of the A3.1, A3.2, A3.3 and A3.4 developments with the MANO architecture

4.2 AI-ENABLED 3D NETWORK PERFORMANCE AND TOPOLOGY FORECAST

4.2.1 Background

The advent of 6G and Non-Terrestrial Networks (NTNs) promises to revolutionise connectivity across the globe. However, these networks introduce new challenges in traffic management, particularly due to their dynamic and heterogeneous nature. Accurate traffic forecasting is essential for efficient resource allocation, congestion avoidance, and ensuring quality of service (QoS). Artificial Intelligence (AI) techniques, especially machine learning (ML), have emerged as powerful tools for addressing these challenges [64] [65].

AI-driven traffic forecasting in 5G networks has seen significant advancements. Deep learning models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, are well-suited for capturing temporal dependencies in traffic patterns. For instance, Li et al. [66] showed the effectiveness of LSTMs in predicting mobile traffic in 5G networks.

Convolutional Neural Networks (CNNs) can extract spatial features from traffic data, enhancing forecasting accuracy. Hybrid models, combining deep learning with traditional statistical methods like ARIMA, further improve robustness [67]. Attention mechanisms can be incorporated to focus on relevant historical data points, leading to more accurate predictions.

AI-driven traffic forecasting in Non-Terrestrial Networks presents unique challenges due to long propagation delays, dynamic topology, and limited ground station coverage. Graph Neural Networks (GNNs) can effectively model the complex relationships between satellite nodes and ground stations as demonstrated by Zhang et al [68]. Federated learning can be used to train models across multiple ground stations, improving privacy and efficiency. Hybrid approaches, combining AI with traditional network simulation tools, can provide more accurate and comprehensive traffic forecasts [69].

AI-driven traffic forecasting is a promising avenue for optimising NTN networks. However, the use of robust ML techniques allowing to proactively manage traffic, allocate resources efficiently, and ensure seamless user experience remains unexplored. Addressing the unique challenges of these networks, such as data quality, real-time processing, and dynamic topologies, will be crucial for the successful deployment of AI-powered traffic forecasting solutions.

4.2.2 Motivation and Contribution

As mentioned just before, satellite-based network KPIs can be more susceptible to noise than their terrestrial equivalents, making accurate predictions challenging. The innovations developed in this activity enable data scalable, accurate forecasting in the presence of KPI noise. Following the univariate KPM forecasting study on the cellular backhaul (CBH) site satellite, detailed in D4.1, this work presents the results of KPM forecasting for links between aircraft and LEO, High Altitude Platform (HAP), and Air-to-Ground (ATG) entities.

A comprehensive review of state-of-the-art deep learning techniques for forecasting on satellite datasets has been conducted. Models have been fine-tuned to optimise for forecasting accuracy on the satellite networking task, more precisely traffic KPI and link quality prediction. Noise reduction techniques have been applied that boost forecasting accuracy in the presence of noise. By utilising spectral noise filtering, KPI time series can be smoothed, improving forecasting performance, without losing the key time-series dynamics required for model training.

Deep learning forecasting techniques developed in this work can leverage big data to outperform purely statistical approaches while maintaining dynamism to changes in KPI dynamics. The approaches explored can be applied to univariate and multivariate use-cases allowing intra-KPI dynamics to be exploited for further forecasting performance gains.

Spectral noise filtering allows deep learning models to continue to make robust forecasts in the presence of KPI noise which typically degrades model performance.

4.2.3 Methodology and approach

A wide-scoped univariate KPM forecasting analysis has been undertaken on two distinct scenario and datasets. The key features of both analysis are as follows:

1. **Data cleaning and augmentation:** First, it focuses on preparing the data for training the forecasting models. This will involve techniques to fix errors, inconsistencies, and missing values in the data. It might also create new data points to improve the training process.
2. **Trying different types of forecasting models:** Next, it experiments with various forecasting models, especially those using cutting-edge deep learning techniques. This will help us identify the best model for the specific forecasting task.

3. **Evaluating forecasts from different dimensions:** Then, it thoroughly assesses the performance of our forecasting models. This evaluation will consider factors like the type of model used, the amount of historical data provided, the forecasting time horizon, and the specific location being analysed.
4. **Improving forecasts by removing noise:** Finally, it explores methods to remove unwanted noise from satellite data (KPMs) that might be impacting the accuracy of the forecasts. Techniques like spectral de-noising might be used to achieve this.

4.2.4 KPM forecasting on the cellular backhaul (CBH) sites

The preceding version of WP4 Deliverable namely D4.1 [1] comprehensively outlined the dataset used for this task. This included data description, collection methodology, and characteristics. Furthermore, D4.1 presented the preliminary data analysis, detailing the initial exploratory steps undertaken to understand the dataset properties to drive the design of different forecasting approaches. This preliminary analysis also presented descriptive statistics and initial visualisations that highlighted key trends and relationships within the data. Finally, D4.1 provided the detailed results of the forecasting task leveraging 3 different deep learning models and the findings obtained.

4.2.5 KPM Forecasting of flight link quality

This section presents an evaluation of machine learning models for predicting Key Performance Metrics (KPMs) of flight link quality. Specifically, the focus is on forecasting Signal-to-Noise Ratio (SNR) and Delay for flight paths between Dublin (EIDW) and London (EGLL). The study utilises a dataset of 223 flight simulations over a 15-day period, considering three distinct station types: LEO, HAP, and ATG. This dataset has been provided by Collins Aerospace. The performance of two state-of-the-art time series forecasting models are benchmarked, TSMixer and NBEATS, in a one-step-ahead forecasting task. The results highlight the importance of appropriate data pre-processing, particularly log normalisation for KPMs with large value ranges, and reveal performance differences between the models based on station type and KPM.

Accurate prediction of flight link quality is crucial for proactive network management in aviation communication systems. Degradations in link quality, quantified by KPMs such as SNR and Delay, can lead to disruptions in communication and potentially compromise safety. Machine learning techniques offer the potential to forecast these KPMs, enabling timely interventions to mitigate adverse effects. The following details an evaluation of two deep learning models, TSMixer and NBEATS, for the task of short-horizon forecasting of SNR and Delay in a realistic flight scenario.

4.2.5.1 Dataset Description

The dataset employed in this study comprises flight link simulations of aircraft traversing routes between Dublin Airport (EIDW) and London Gatwick Airport (EGLL). The dataset spans 15 days and includes a total of 223 individual flights. The geographical distribution of the flight paths is illustrated in Figure 4-2.

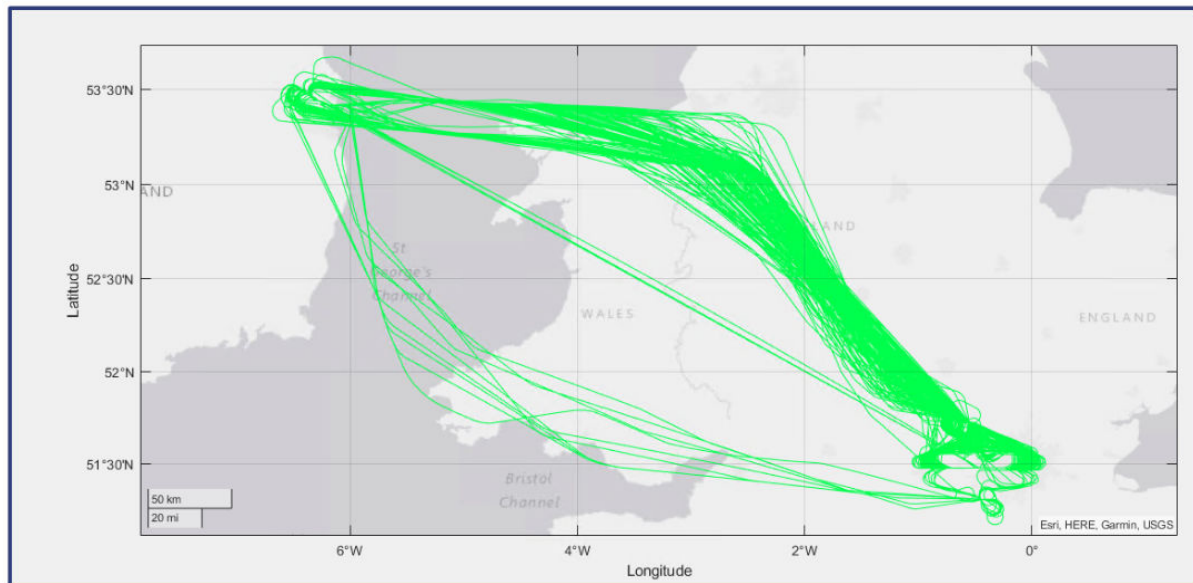


Figure 4-2 Geographical distribution of the flight paths.

The flight link infrastructure is modelled with three distinct types of ground stations:

- Low Earth Orbit (LEO): A constellation of 198 LEO stations provides wide-area coverage. These stations are characterised by their orbital motion.
- High Altitude Platform (HAP): Two HAP stations are strategically positioned at fixed locations along the flight paths.
- Air To Ground (ATG): Fourteen ATG stations provide ground-based connectivity along the flight routes.

The heterogeneous nature of this infrastructure, with varying station types and mobility characteristics, presents a complex scenario for link quality prediction.

4.2.5.2 Forecasting models

Models were chosen to represent a range of neural network architectures: Seq2Seq LSTM (RNN), AutoFormer (Transformer), and NBEATS (MLP). For each CBH site, and each KPM, a validation set of 480 steps (20 days) was held out for evaluation and the models were trained on the remaining data.

4.2.5.2.1 Seq2Seq LSTM

The Seq2Seq LSTM model is an RNN-based deep learning approach originally applied to machine translation, but readily applicable to time series forecasting tasks, as detailed in [70]. It employs an encoder-decoder architecture where inputs are encoded into an 'encoder vector' by a multi-layered LSTM. This encoding aims to condense the key temporal dependencies of the input time series in a fixed-length vector. The 'encoder vector' is decoded via a separate multi-layered LSTM followed by a dense layer that outputs the final forecast. Decoupling the encoding and decoding step into two different networks allows the model to handle arbitrary input steps and forecasting horizons. The architecture is demonstrated in Figure 4-3 [71].

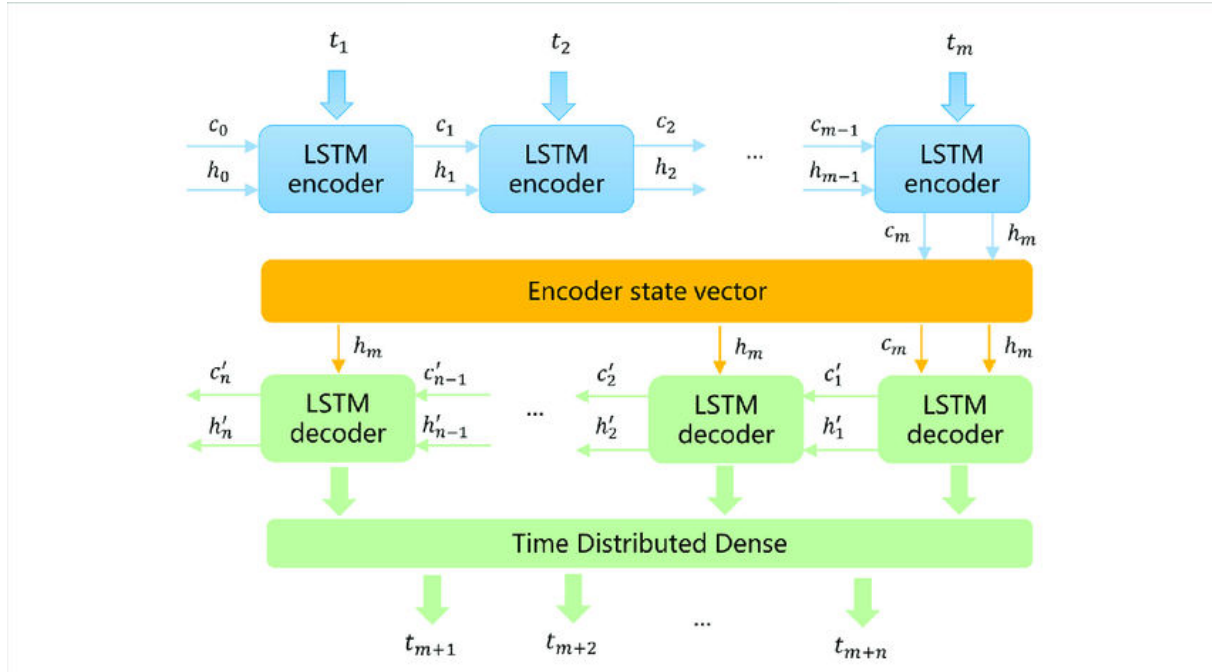


Figure 4-3: Seq2Seq LSTM architecture.

4.2.5.2.2 AutoFormer

The Autoformer model is a transformer-based deep learning network originally applied to time series forecasting proposed in [72]. Autoformer modifies the typical transformer forecasting architecture with two key improvements: A series decomposition block and replacing self-attention with autocorrelation. The series decomposition block progressively extracts trend and seasonality components from local inputs throughout the network with an efficient and simple decomposition operation. The self-attention operation is replaced with an auto-correlation operation which is more suitable for the periodic dependencies typically present in time series. Specifically, the auto-correlation mechanism aids in the discovery of sub-series dependencies based on periodicity as opposed to the point-wise dependencies discovered by self-attention (Figure 4-4). The architecture is demonstrated in Figure 4-5.

The decomposition operation is defined by the following equation.

For input series χ , trend (χ_t) and seasonality (χ_s),

$$\chi_t = \text{AvgPool}(\text{Padding}(\chi)) \quad (4-1)$$

$$\chi_s = \chi - \chi_t$$

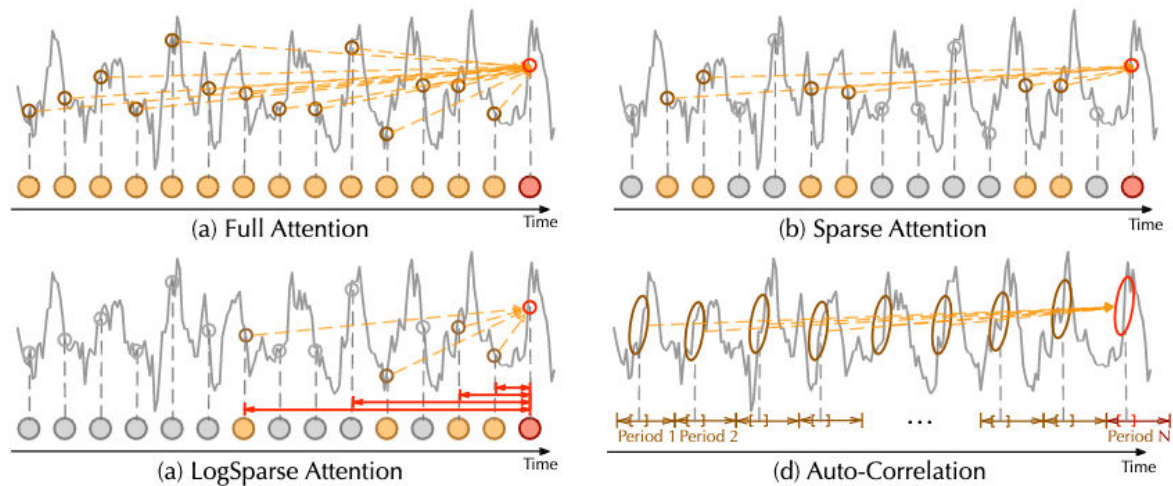


Figure 4-4: Auto-correlation vs. self-attention.

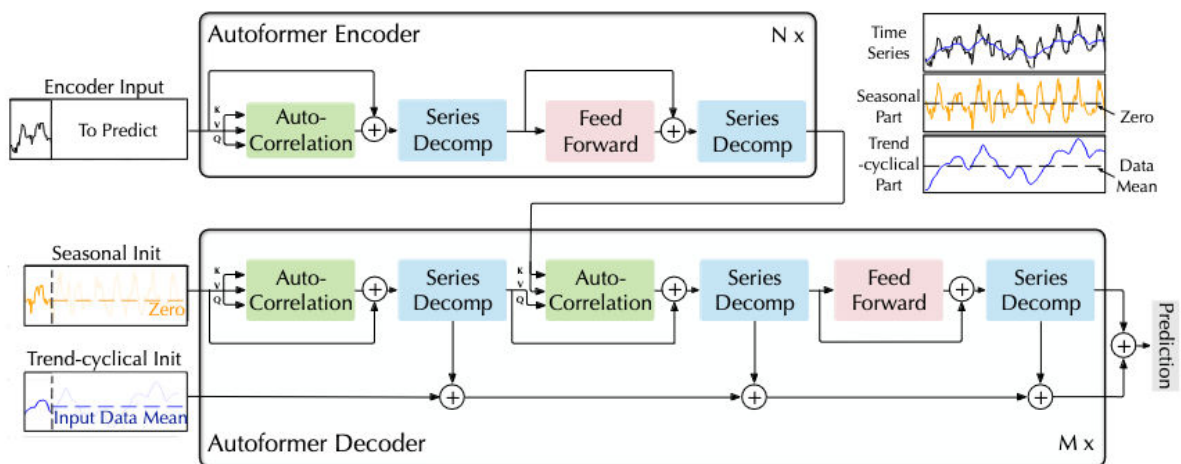


Figure 4-5: Architecture diagram.

4.2.5.2.3 N-BEATS

The N-BEATS model is an MLP-based deep learning approach to time series forecasting, detailed in [73]. It utilises a unique architecture built on a "fork-like" block that can generate both forecasts and recreate past data (backcasts) from local inputs. These blocks can be configured to consider factors like trends and seasonality, making the forecasts interpretable if desired. The blocks are further stacked using a "doubly residual stacking" method, where the past block's backcast is subtracted from the input to simplify forecasting for subsequent blocks. The final forecast is the sum of all forecasts from each block across all stacks. Notably, the experiments in our work use the generic, non-interpretable version of N-BEATS to prioritise forecasting performance over interpretability, as interpretable constraints can slightly reduce accuracy. The architecture diagram is depicted in Figure 4-6.

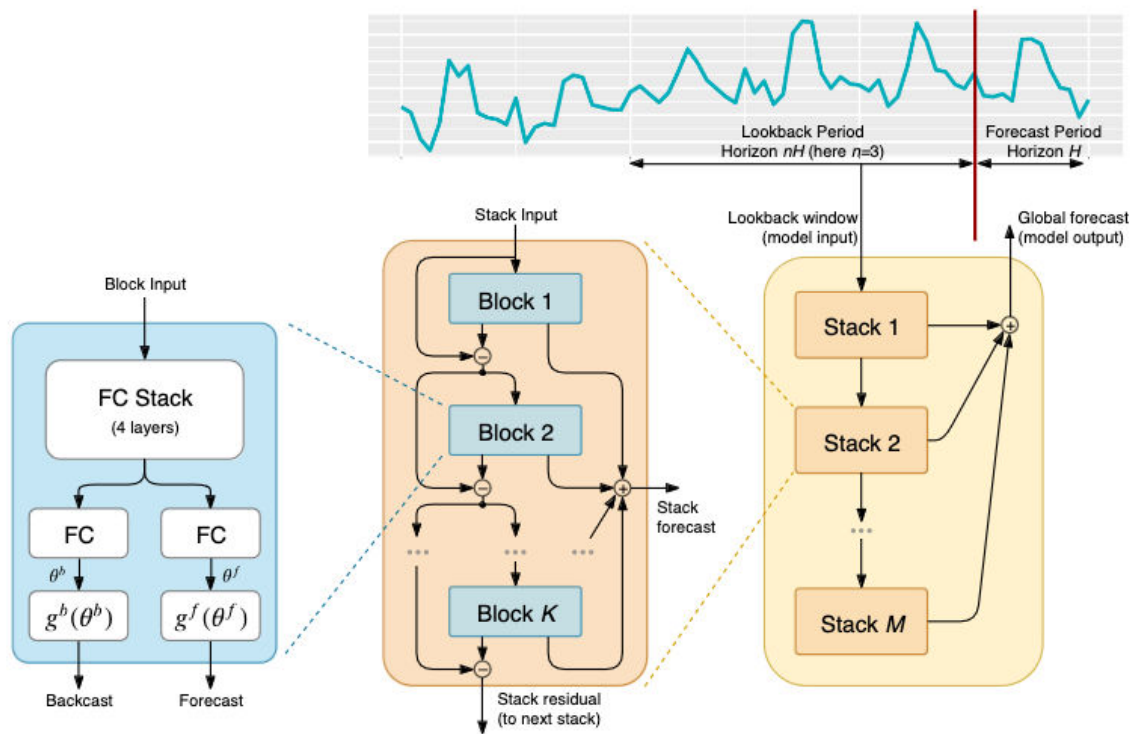


Figure 4-6: Architecture of the N-BEATS model.

4.2.5.3 Prediction Task

The primary objective is to reliably predict link quality KPMs, specifically SNR and Delay, for flights not included in the model training phase. Accurate forecasting of these KPMs enables proactive identification of potential link degradation, facilitating timely network adjustments to maintain optimal performance. The prediction task is formulated as a one-step-ahead forecast, utilising a 4-step input window.

An initial analysis of the KPM data revealed significant differences in the value ranges across the three station types (LEO, HAP, and ATG).

For the SNR KPM, HAP and ATG stations exhibit considerably larger value ranges compared to LEO stations. Conversely, for the Delay KPM, LEO stations demonstrate the most extensive value ranges.

The distributions of SNR and Delay across station types are visualised in Figure 4-7 and Figure 4-8, respectively.

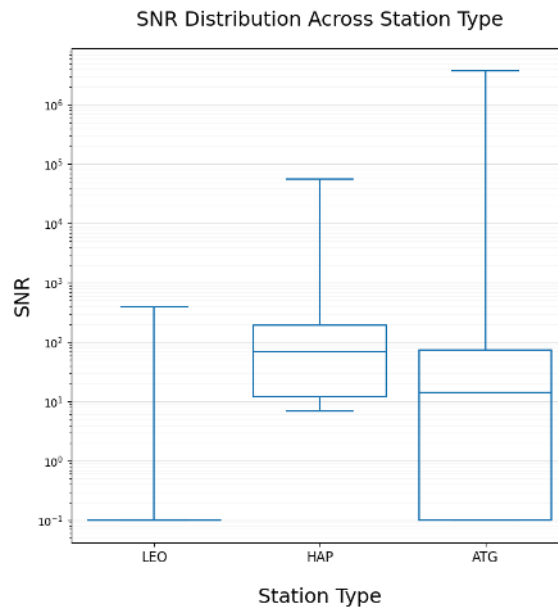


Figure 4-7: SNR distribution across station types.

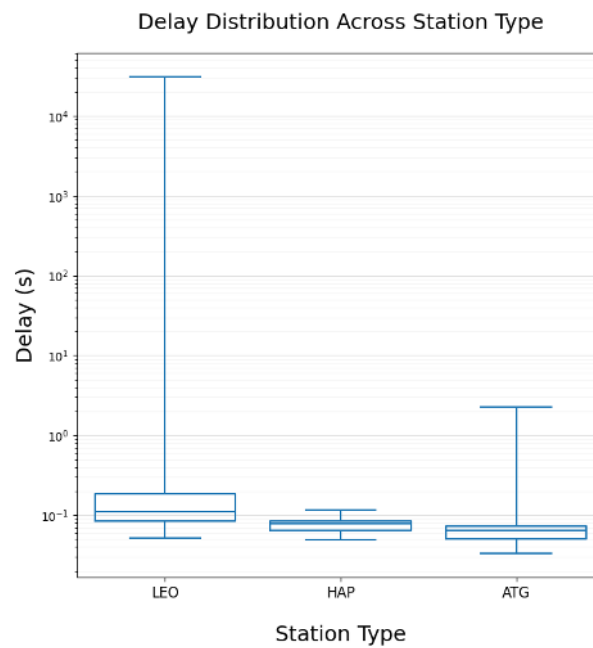


Figure 4-8: Delay distribution across station types.

To mitigate the impact of large value ranges on model performance, a log transformation was applied to the KPM data. This technique is commonly employed to compress the scale of data and reduce the influence of outliers. The effect of log normalisation on prediction errors for the SNR KPM is shown in Figure 4-9.

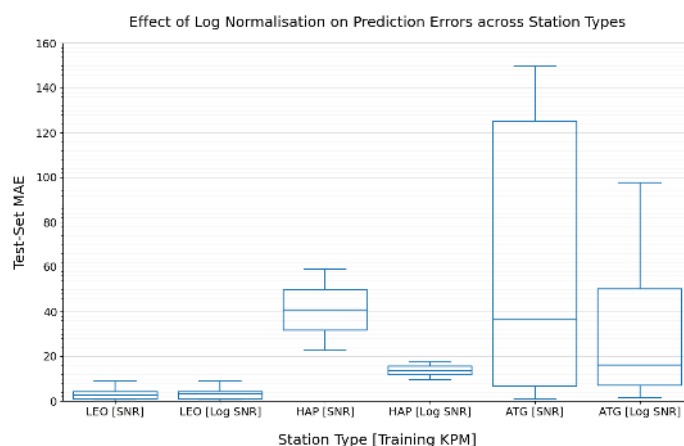


Figure 4-9: Effect of log normalisation on prediction errors for the SNR KPM.

The results demonstrate that log normalisation leads to a substantial reduction in prediction errors for HAP and ATG stations, which exhibit large SNR ranges. For LEO stations, where the SNR range is smaller, the impact of log normalisation on prediction error is negligible.

4.2.5.4 Model Results and Predictions

Following appropriate log normalisation of the SNR data, the performance of the TSMixer and NBEATS models was evaluated. Figure 4-10 presents a comparison of the prediction errors for the SNR KPM across station types and models.

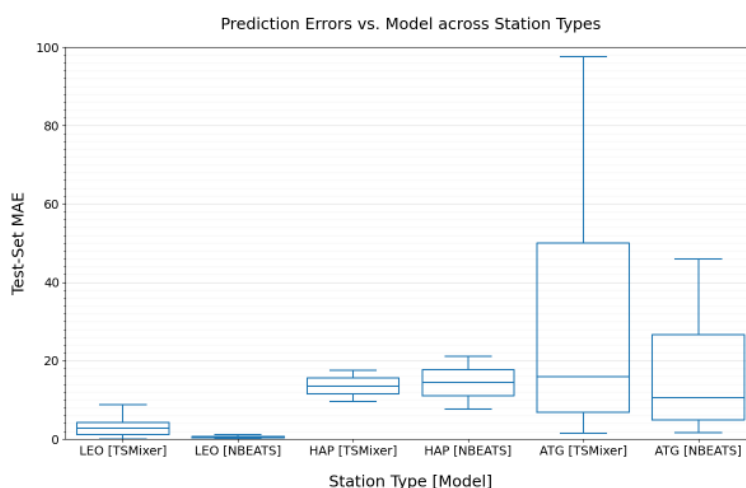


Figure 4-10: Prediction errors for the SNR KPM across station types and models.

The results indicate that NBEATS generally outperforms TSMixer for SNR prediction. Given that LEO and ATG stations constitute the majority of stations in the dataset, this suggests that NBEATS is better suited for overall SNR forecasting in this scenario.

A detailed analysis of LEO station 119, where TSMixer exhibits particularly poor performance, reveals that NBEATS is more effective at capturing the timing of transient spikes in the SNR. Visualisations of the predictions for LEO station 119 generated by TSMixer and NBEATS are provided in Figure 4-11 and Figure 4-12, respectively.

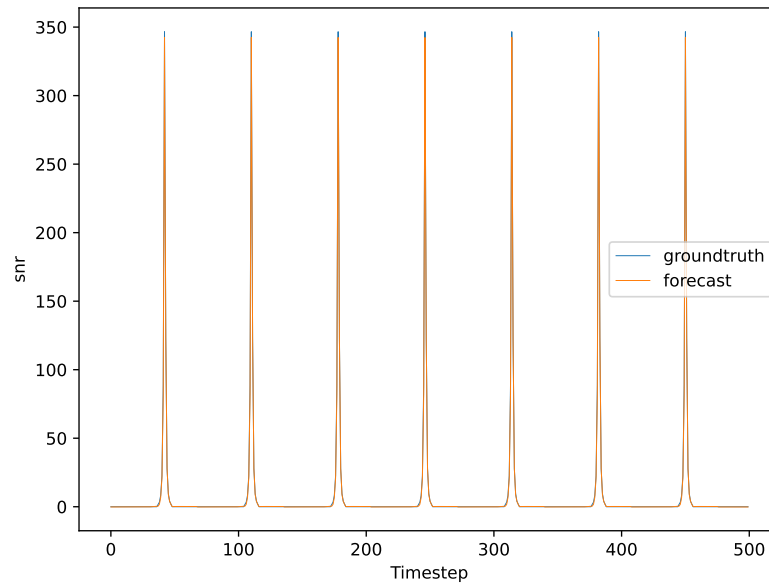


Figure 4-11: NBEATS LEO 119 predictions.

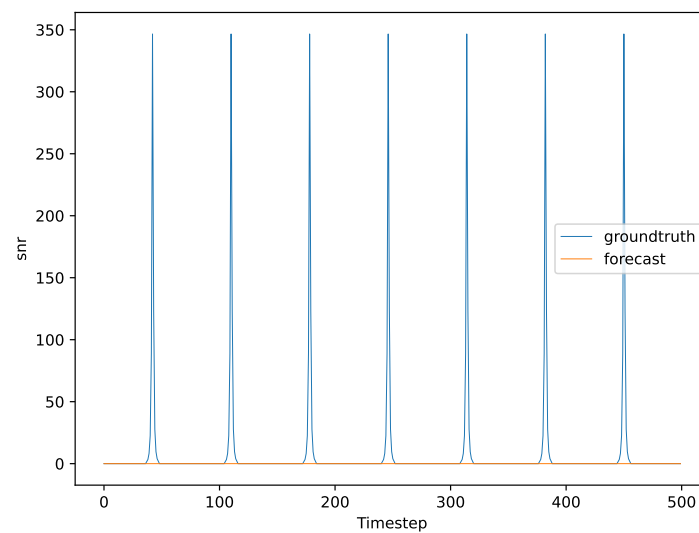


Figure 4-12: TSMixer LEO 119 predictions.

The performance of the models in forecasting the Delay KPM was also assessed. Figure 4-13 illustrates the prediction errors for Delay across station types.

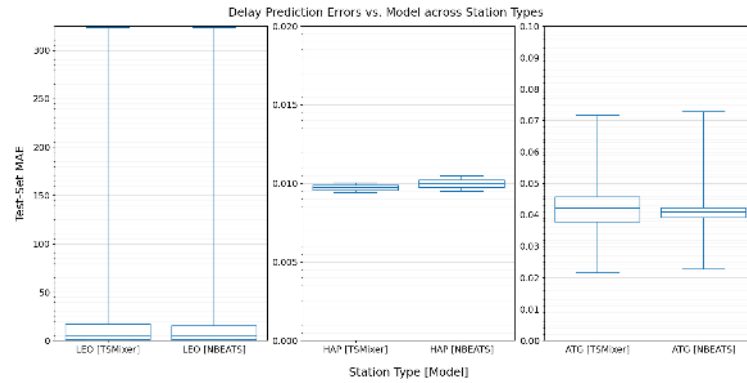


Figure 4-13: Delay Prediction Errors vs. Model across station types.

The results show that TSMixer and NBEATS exhibit comparable performance in Delay forecasting. However, both models struggle to accurately predict large, transient spikes in delay values observed in some stations. Notably, the timing of these spikes is consistent across the training and test datasets. Predictions for LEO station 10, demonstrating this limitation, are provided in Figure 4-14 and Figure 4-15 for TSMixer and NBEATS, respectively.

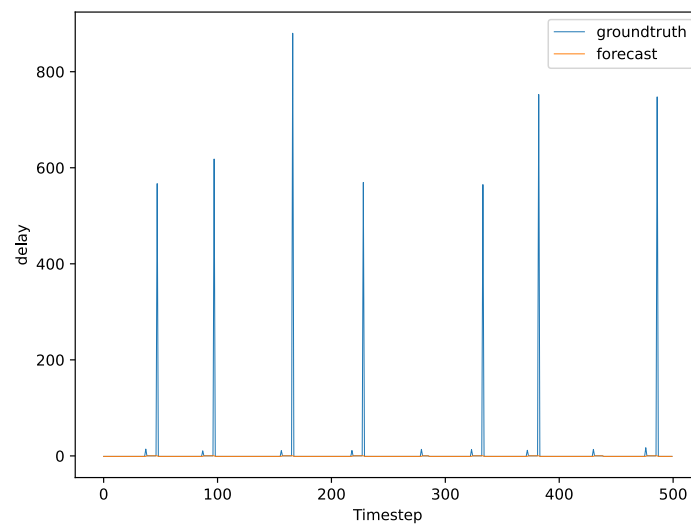


Figure 4-14: TSMixer LEO 10 predictions.

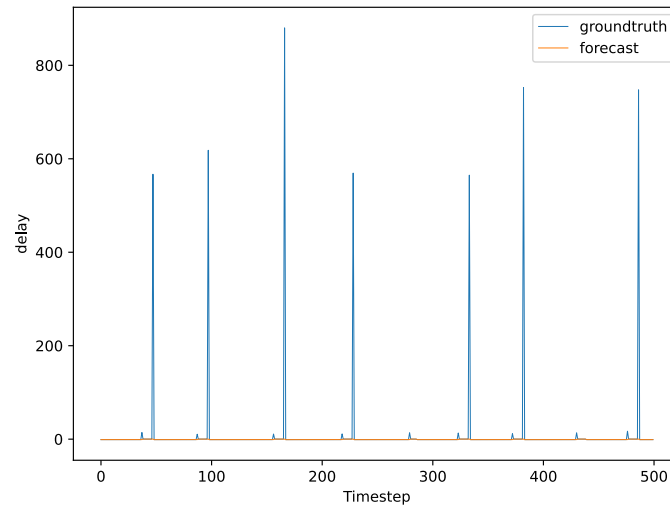


Figure 4-15: NBEATS LEO 10 predictions.

Figure 4-16 provides a comparative view of the delay dynamics in the training and test sets for LEO station 10, highlighting the temporal consistency of the spikes.

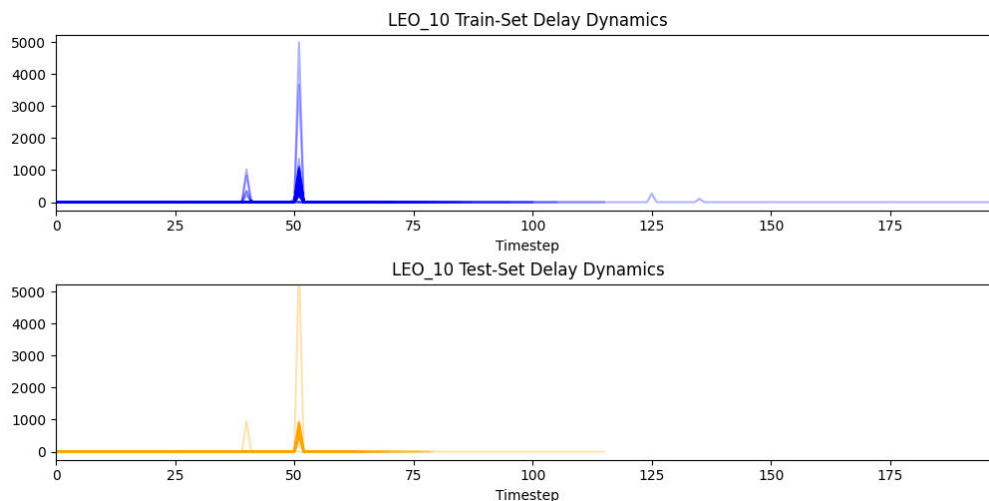


Figure 4-16: LEO 10 Train vs. Test Set Dynamics (all flights plotted with start timestep aligned).

In contrast, for stations where the delay dynamics are smoother and lack prominent spikes, both models demonstrate good tracking performance. However, the models do not fully capture the inherent noisiness of the observed data. Predictions for HAP station 1, illustrating this behavior, are shown in Figure 4-17 and Figure 4-18.

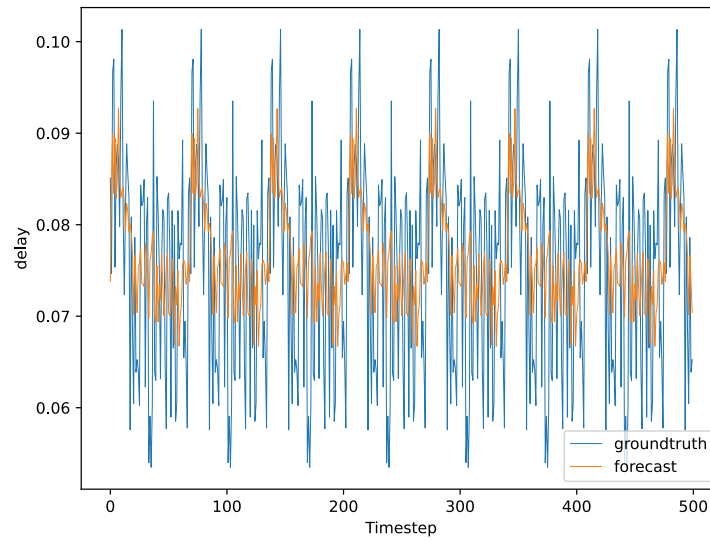


Figure 4-17: TSMixer HAP 1 predictions.

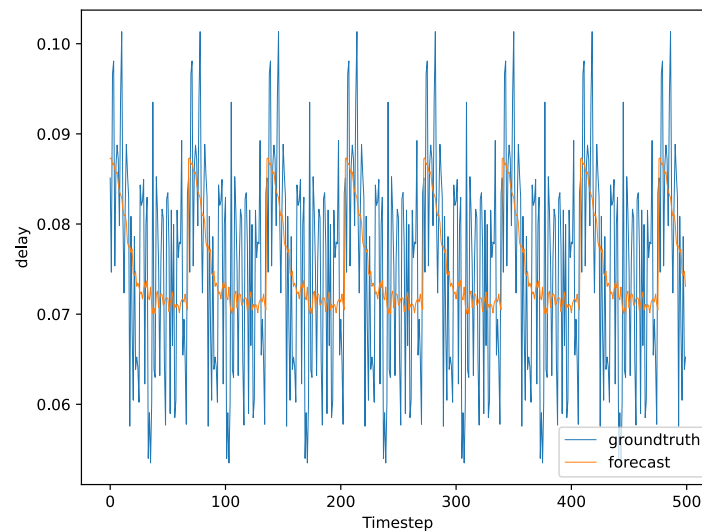


Figure 4-18: NBEATS HAP 1 predictions.

4.2.5.5 Inference Pass Speed

The computational efficiency of the models is an important consideration for real-time deployment. Table 4-1 presents the inference times for a full inference pass, encompassing all 428 models (214 stations x 2 KPMs) and including input normalisation and output denormalisation.

Table 4-1: Inference times for a full inference pass.

Machine	vCPUs / Cores	Memory	Full Pass Inference Time
M1 Mac	8	16GB	0.546s
GCloud n1-standard-16	16	60GB	1.476s

The results indicate a substantial difference in inference speed between the hardware platforms. The M1 Mac demonstrates a significantly faster inference time compared to the Google Cloud n1-standard-16 instance.

4.2.5.6 Discussion

The findings of this evaluation provide valuable insights into the challenges and opportunities in forecasting flight link quality. The importance of data pre-processing, particularly log normalisation, for handling KPMs with wide dynamic ranges has been clearly demonstrated. The performance differences observed between TSMixer and NBEATS highlight the significance of model selection based on the specific characteristics of the data and the prediction task. The limitations of both models in accurately capturing transient spikes in delay warrant further investigation, potentially involving the incorporation of exogenous variables or the exploration of alternative model architectures. Finally, the observed differences in inference speed underscore the importance of considering computational resources when deploying these models in real-world applications.

4.3 REAL-TIME ENERGY-EFFICIENT RESOURCE ALLOCATION AND ROUTING

4.3.1 Background

The forthcoming 6G mobile networks, which are anticipated to integrate TN and NTN, including satellite and aerial platforms, will deliver seamless and ubiquitous connectivity. This integration is crucial to meet the escalating demands for high data throughput, ultra-low latency, extensive connectivity, and enhanced energy efficiency. The realisation of such capabilities necessitates advanced network architectures capable of dynamic resource management, intelligent orchestration, and elastic function placement across a highly heterogeneous infrastructure.

A central challenge in 6G network management lies in the efficient joint management of user association, traffic routing, and the placement of VNFs, or more broadly, xNFs, across a distributed computing continuum encompassing edge, fog, and cloud resources. Traditional approaches that decouple these issues often fail to attain optimal network-wide performance, particularly under the constraints of energy efficiency, processing capacity, and delay sensitivity in hybrid TN-NTN environments.

Recent research has explored a variety of strategies for resource allocation and Service Function Chain (SFC) placement in multi-layered networks, including heuristic methods, optimization techniques, and increasingly, AI-based approaches. However, many existing solutions either disregard the impact of the joint study of the problem [74] [47] or not considering all the domains of the integrated network, i.e., terrestrial, aerial and space [75] [76] [77]. Authors in [74] focus on terrestrial data collection and transmission using HAPs and

LEO satellites in 6G aerial and space access networks prioritising throughput maximisation, not addressing user association and direct handheld access to the space domain, or xNF placement, focusing solely on data flow and resource allocation. In [47], authors propose a constrained multi-agent reinforcement learning algorithm (CMADR) for dynamic routing in integrated satellite-terrestrial networks, excluding aerial platforms and not addressing xNF placement or user association.

The work in [69] jointly optimizes user association, traffic routing, and xNF placement in an integrated terrestrial-satellite 6G network, aiming to maximize energy efficiency and user acceptance ratio. However, it does not incorporate the aerial domain. Similarly, [70] addresses the joint optimization of routing, bandwidth allocation, user association, channel allocation, and power allocation in satellite-terrestrial integrated networks, with the goal of minimizing satellite power consumption while meeting QoS requirements. Yet, it also omits the aerial domain, falling short of offering a comprehensive TN-NTN solution. In [71], the authors present a joint framework for optimizing gateway placement, dynamic routing, and flow management in satellite networks, taking into account inter-satellite links and terrestrial gateways. Nevertheless, aerial platforms such as LAPs or HAPs are not considered, and the work does not address xNF placement or explicitly prioritize energy efficiency, instead focusing on latency and flow management.

4.3.2 Motivation and Contribution

The motivation behind this work is the urgent need to develop an integrated, energy-efficient, and scalable orchestration mechanism specifically designed for heterogeneous 6G environments that encompass terrestrial, aerial, and satellite domains. The challenges posed by joint user association, multi-hop traffic routing, and xNF placement, particularly under real-time constraints, necessitate a novel solution that can strike a balance between computational efficiency and service quality. This work builds upon D4.1 [1] by proposing PETA (Power-Efficient Terrestrial and non-terrestrial resource Allocation), an enhanced version of the online algorithm in [1]. PETA is designed to comprehensively address the interconnected challenges of resource allocation in integrated networks.

Key extensions include incorporating Low Altitude Platform Systems (LAPSSs), commonly known as UAVs, as a new type of aerial base station and introducing a new backhaul category connecting the aerial domain. Additionally, PETA now accounts for interference among nearby base stations operating on the same frequency band in both the aerial and space domains.

PETA intelligently manages service requests by dynamically selecting user association points, identifying the most power-efficient routing paths, and strategically placing xNFs based on node capabilities and the underlying network topology.

One of the core contributions of this work is the joint optimization approach adopted by PETA. This approach contrasts sharply with the disjoint treatment often seen in SotA solutions. Instead of handling user association, traffic routing, and xNF placement in isolation, PETA treats these elements as interdependent components of a single optimization problem. This results in enhanced overall system efficiency, particularly under fluctuating load and network topology conditions.

PETA introduces a power-aware path selection mechanism that considers both transmission and computational power requirements. By constructing a weighted graph for each service request and analysing all feasible paths, the algorithm selects those routes that not only meet the required quality of service constraints, such as delay and capacity, but also minimize overall power consumption. This capability is crucial in hybrid TN-NTN environments where power resources may be scarce or unevenly distributed across network domains.

The algorithm's innovative xNF placement strategy, guided by a custom-designed node ranking metric called Ω , is of great significance. This metric evaluates several key attributes of candidate computational nodes, such as their closeness centrality within the network, available CPU capacity, and operational status. By prioritizing nodes based on these factors, the algorithm ensures efficient and context-aware function placement. This means that each xNF is hosted in a node that optimally balances performance, power, and availability.

The effectiveness of this approach is validated through extensive simulations in realistic 6G network scenarios. These scenarios include diverse TN and NTN configurations, satellite and aerial node integration, and heterogeneous traffic demands. The results demonstrate that PETA achieves a significant improvement in energy efficiency compared to representative SotA algorithms like VNFPAR and BCSP. At the same time, it maintains low execution times and consistently high service acceptance ratios. These findings highlight the practical relevance and superiority of our joint optimization approach, particularly for future networks that require intelligent, power-aware orchestration across vastly distributed infrastructures.

4.3.3 Proposed algorithm

In this section, a heuristic algorithm, named PETA, is proposed to address the joint user association, traffic routing, and xNF placement problem in the 3D ETHER architecture. The detailed system model, as well as the problem formulation, can be found D4.1 [1].

The algorithm's development is closely aligned with the use cases of Demo 2 and Demo 3. In Demo 2, an energy-efficiency handover policy is proposed, that is built upon PETA's logic, enabling the UEs to efficiently transition between terrestrial and non-terrestrial BSs. In Demo 3, PETA is utilised to identify the best-case scenario for user association in a real-time architecture demonstration, focusing on air-space safety operations that demand extremely low latency, while prioritising energy saving. Both Demos necessitate the algorithm's capabilities to ensure the optimal functioning of the multi-layered integrated network, in terms of real-time capabilities and minimising the energy consumption.

Delving into the algorithm itself, the primary objective of PETA is to maximise user acceptance ratio and enhance the network energy efficiency. As depicted in Figure 4-19, PETA comprises two distinct phases: a) in the initial phase, the algorithm determines the user association and the traffic routing path, and b) in the second phase, the algorithm places the xNFs of the SFC required by each UE in the precise order specified by the SFC.

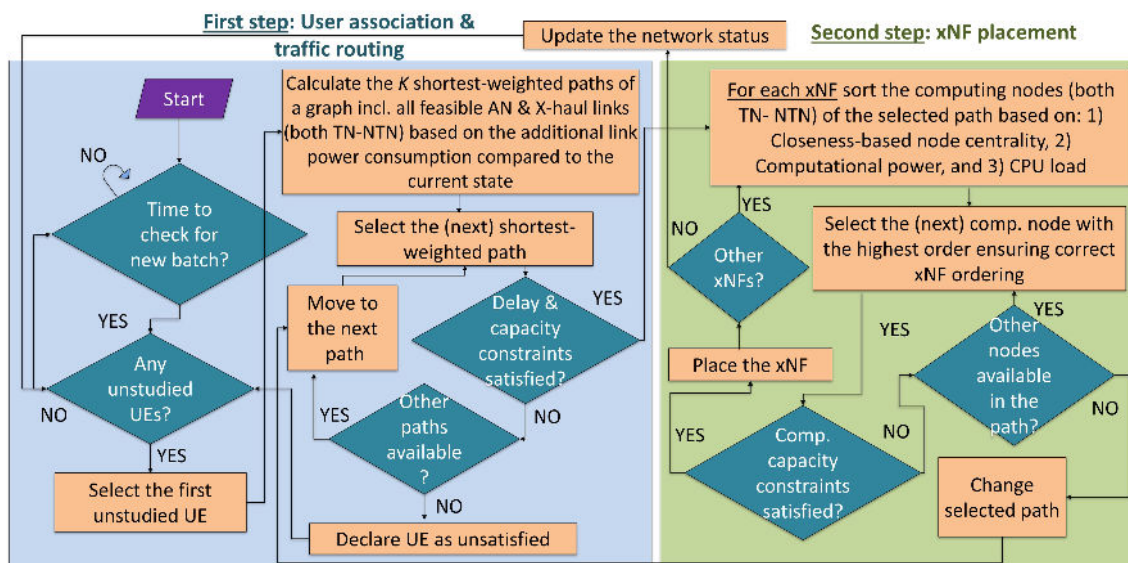


Figure 4-19: Operation flowchart of PETA.

PETA periodically monitors new batches of SRs. Upon identifying a non-empty batch, PETA constructs a weighted graph for each SR, analysing all available paths from the source to the destination, i.e., the UE (the downlink is studied), based on their power consumption. Each path incorporates feasible wireless and fibre X-haul links, i.e., front-/mid-/back-haul links, as well as the AN link between the serving BS and the UE. The shortest-weighted path, with the additional power consumption being set as weight for each link, is subsequently selected to fulfil the UE's demands, provided that capacity and delay constraints are not violated. In the event of a constraint violation, PETA selects the next available shortest path, ensuring no constraints are breached. If no alternative path is available, PETA discards the SR. Subsequently, the algorithm checks for new SRs in the batch. If none are detected, it waits until the next batch arrival time. Otherwise, it proceeds with the next SR of the batch.

Once a path has been selected for the SR under study, PETA proceeds to the next step, which is the placement of the xNFs. To determine the optimal placement of each requested SFC xNF in the available nodes specified by the selected path, PETA sorts the nodes of the path based on a parameter denoted by Ω . This parameter considers the node's closeness centrality, maximum computational capacity, and CPU load. Regarding the CPU load, four values are permitted: a) 1 (high priority) when the node possesses sufficient computational capacity and can accommodate the xNF without the need for a new instance initiation, b) 0.5 (medium priority) when the node is active and has adequate computational capacity but requires a new instance initiation to host the xNF, c) 0.1 (low priority) when the node is inactive and a new xNF instance is necessary, and d) 0 (no priority) when the node is unable to host the xNF. After sorting the nodes, PETA selects the highest-ranked node and places the xNF, provided that the computational constraints are met. If no suitable node is found, PETA returns to the initial step and selects the next shortest path, repeating the process in the second step until all xNFs are placed or there are no other available paths. In such a case, PETA blocks the SR. If all xNFs from the SFC are successfully placed, PETA updates the network state and proceeds to the next SR, if any. Otherwise, it waits until the appropriate time to check for new batches, repeating the aforementioned steps.

4.3.4 Performance evaluation

Extensive simulations have been conducted (20 BS distribution scenarios with 20 UE hotspot traffic distribution snapshots each) in MATLAB R2024b for both the proposed heuristic and the reference algorithms, utilising an Apple M4 Pro (12-core CPU) processor with 24 GB of RAM. For the TN, a gNB sector area is considered, located at Thessaloniki, Greece (geodetic coordinates: [40.63, 22.94]) with a 500 m radius. This sector is overlaid with two SC clusters, each consisting of four SCs uniformly distributed within a 100 m radius from the cluster center [78]. Millimeter Wave (mmWave) backhaul links are also assumed among TN BSs that are located less than 200 m apart. The gNB and a randomly chosen SC per cluster have fibre connections to the transport network, which is divided into two levels, i.e., fog and cloud, based on their distance from the UE. Each level has four randomly chosen nodes that are fibre-connected to each other. Figure 4-20 depicts the different BSs' coordinates. For simplicity, only the LEO and GEO satellites that have Line of Sight (LoS) during the entire UE service duration are considered, thereby avoiding handovers for the time period studied. To this end, 1 Iridium, 18 OneWeb, 52 Starlink, and 2 GEO satellites (HellasSat 3 and HellasSat 4) are considered, whose orbital elements were retrieved from [79]. For the aerial domain, a HAPS is employed, centrally located above Thessaloniki at an altitude of 20 km, and 2 LAPs around the surroundings at an altitude of 5 km (geodetic coordinates: [40.6365602718026, 22.937585787631857] and [40.630327767771206, 22.949539421793666]). For the TN AN, orthogonal channels are employed between the gNB and the SCs (100 Physical Resource Blocks (PRBs) for each TN BS). However, SCs belonging to different clusters as well as aerial and satellite nodes that reuse the same frequencies may interfere with each other. For the ISLs, the interference is considered negligible due to the high frequencies used, which significantly attenuate in other directions beyond the direct LoS path.

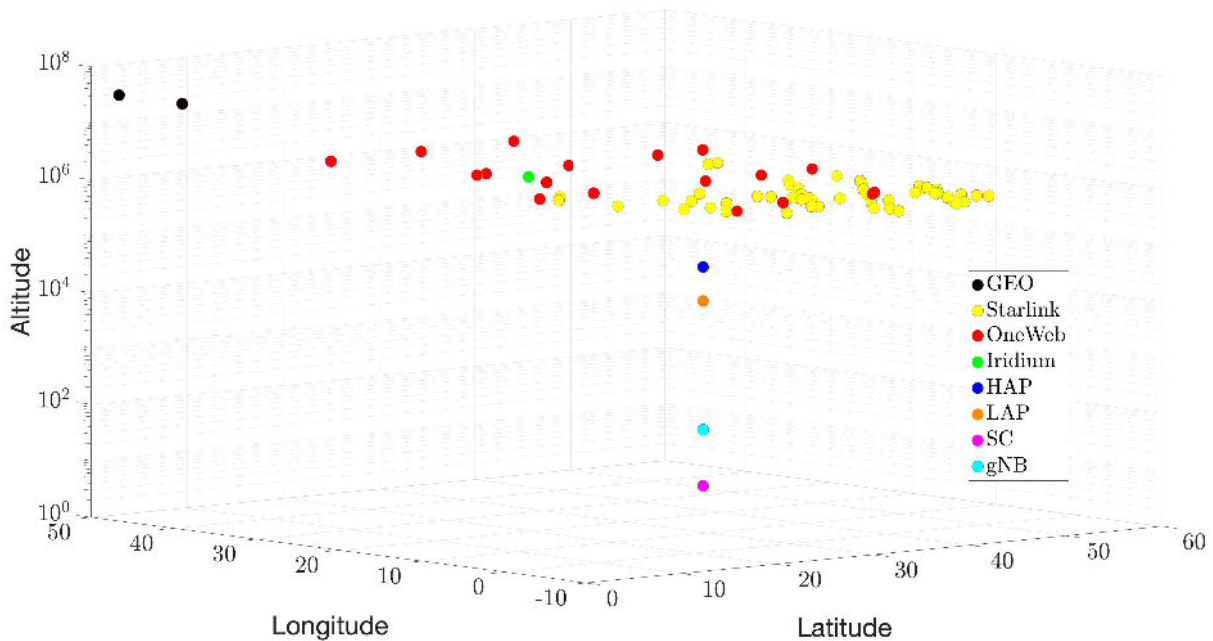


Figure 4-20: Simulation setup example depicting both the terrestrial and non-terrestrial base station positions.

For illustration purposes, 5 distinct SFCs are considered, each one being an ordered sequence of VNFs (although any xNF could be employed) with specific data processing capacities and requirements, as explained in [1]. The rest of the simulation parameters, for the access and the X-haul network, such as the number of transceiver chains (N_{TRX}), the idle power consumption (P_0), the load-dependent slope parameter depending mainly on the link's

transmitter amplifier performance (Δ_p), are summarised in Table 4-2, while the CPU cores and power consumption values for the computational nodes of [78] are employed for the edge, fog and cloud, both in the TN and NTN domain. For the fibre links, the power consumption of a switch in idle state is $P_{idle}^{(sw)} = 315 W$, while the consumed power per port is $P_{port} = 7 W$ and the fibre link rate equals to 10 Gbps. Finally, a packet length of 1.5 KB is assumed.

Regarding the X-haul links in Table 4-2, five different types are considered that connect the different layers (inter-layer links) or the BSs in each layer (intra-layer links). The intra-layer links are: a) TN BH connects the terrestrial domain (gNb and SCs) via mmWave links, b) air-to-air (A2A) BH connects the aerial domain (LAPS and HAPS), also via mmWave links, and c) ISL BH connects the space domain (LEOs, MEOs, GEOs) via Free Space Optical (FSO) links operating in the THz frequencies. As for the inter-layer links, the following are considered: a) TN-NTN BH, which connects the terrestrial domain with the aerial and the space domain (gNb/SC to LAPS/HAPS/LEO/MEO/GEO or vice versa), and b) HAPS-to-space (H2S) BH, which connects the HAPS from the aerial domain with the space domain (HAPS to LEO/MEO/GEO or vice versa). Fibre X-haul links are also considered, allowing fibre optical connectivity between the terrestrial domain and the 1st and 2nd aggregation layers, as explained in [1]. It should be noted that flexible functional split is beyond the scope of this work. Accordingly, in our results all nodes in both terrestrial and non-terrestrial domains are treated as full gNBs. Nevertheless, the proposed framework can be readily extended to support flexible functional split. For this reason, we use the term X-haul to collectively refer to transport links across all split options, i.e., front-, mid-, and backhaul.

The proposed heuristic solution is compared with two SotA algorithms, which have been properly adapted to the studied scenarios for a fair comparison. The first algorithm is named VNFPAR [80] and performs first the traffic routing (criterion: lowest additional Bandwidth (BW) needed, i.e., throughput of UE/max capacity) and then the xNF placement (criterion: lowest additional GFLOPs needed) so that the E2E latency constraint is not violated. The second algorithm is named BCSP [81] and assigns the VNFs to the computational node with the highest betweenness centrality while selecting the lowest delay route that meets the E2E latency requirement of the SR. As both algorithms do not take into account the user association problem, the default criterion is applied, i.e., the UEs are connected to the BS from which they receive the highest signal power (highest SINR).

Table 4-2 Simulation Parameters [81], [78], [82], [83].

Part of the network	AN					BH Network				
	TN		LAPS	HAPS	SN	TN	TN-NTN	H2S	A2A	ISL
	gNB	SC								
F (GHz)	2		28	28	28	60	28	28	28	200000
BW (GHz)	0.02		0.2	0.2	0.4	0.2	0.6	0.6	0.6	20
N_{TRX}	128	32	32	64	128	32	32	64	64	128
P_0 W	145	6.8	1	1	1	3.9	3.9	3.9	3.9	3.9
Δ_p	4.2	4	2	2	1	5	4	4	4	10

4.3.5 Results

In Figure 4-21, the energy efficiency (bits/Joule) is illustrated for the proposed work in comparison to the VNFPAR and BCSP algorithms under various traffic load conditions, including the number of simultaneous UEs in the system and arrival rates ($1/\lambda$). In this work, arrival rates follow Poisson distribution and there are two different cases: a) a new user enters the system approximately every second ($1/\lambda=1$), and b) approximately every 10 seconds ($1/\lambda=10$). The system checks for new SRs in terms of batches, which in each case is selected equal to twice the arrival rate. As observed, for high arrival rates ($1/\lambda=1$), both the proposed algorithm and the SotA algorithms exhibit an increase in energy efficiency. This occurs as the number of concurrent active UEs in the system rises, leading to a corresponding increase in the total data rate. It is important to note that the total throughput increases at a faster rate than power consumption leading to increased energy efficiency. On the contrary, for low arrival rates ($1/\lambda=10$), the prolonged inter-arrival times result in some UEs exiting the system upon completion of their service time, a phenomenon that becomes more pronounced in scenarios with a larger number of users. This process leads to a decrease in the total data rate and, consequently, a lower energy efficiency. Notably, PETA achieves an energy efficiency that is up to 5.9 and 5.8 times superior to the VNFPAR and BCSP, respectively.

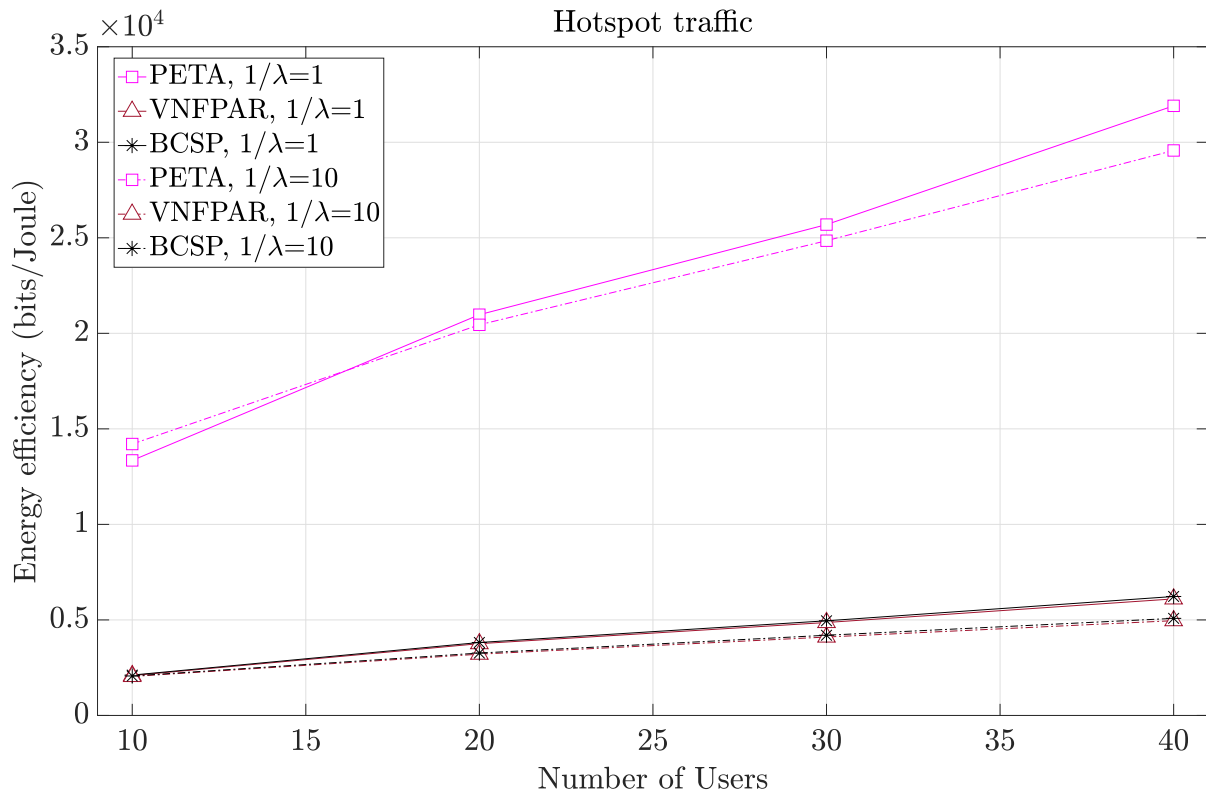


Figure 4-21: Energy efficiency (bits/Joule) of all algorithms for different traffic load conditions and SR arrival rates ($1/\lambda$).

In Figure 4-22, the execution time increases with higher arrival rates for all algorithms, as anticipated. This is because, despite the arrival of SRs in batches when the arrival rate is high, the number of concurrent SRs in the system increases, leading to a reduction in the number of feasible solutions and, consequently, to higher computational time for all algorithms. PETA, however, strikes a satisfactory balance between energy efficiency and complexity, significantly outperforming both SotA algorithms in terms of energy efficiency, while keeping the complexity in acceptable levels. This is attributed to the fact that, unlike the SotA, PETA concurrently

addresses the user association problem, resulting in enhanced flexibility. Notably, in the SotA, the serving BSs are already determined based on the optimal SINR. Additionally, execution time can be significantly reduced by leveraging hardware acceleration techniques and deploying more powerful dedicated hardware, thereby enhancing the practicality of the proposed algorithm in demanding real-time 6G scenarios. In addition, in all cases, the SR acceptance ratio is 1 for all algorithms, indicating that no SR is being blocked, as illustrated in Figure 4-23.

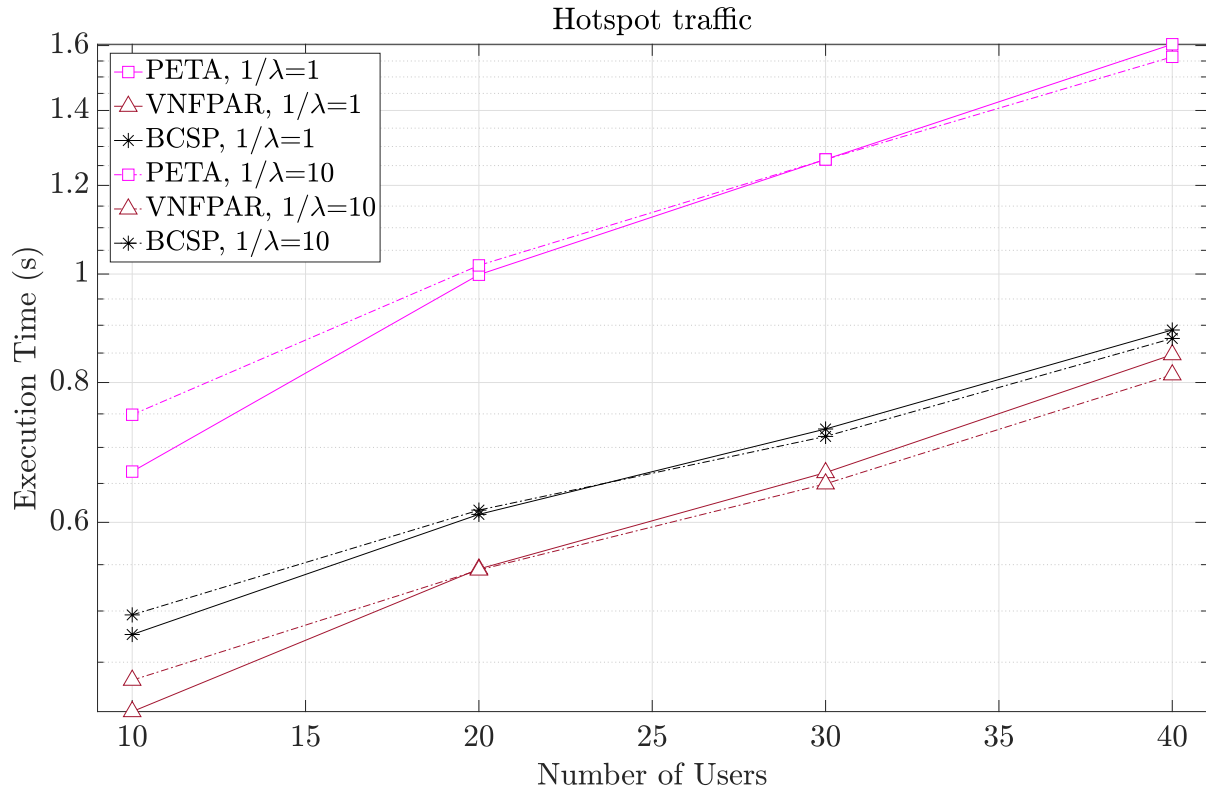


Figure 4-22: Execution time in sec of all algorithms for different user traffic load conditions and SR arrival rates ($1/\lambda$).

This is validated by Figure 4-24 and Figure 4-25, which present the power break-down of all algorithms under low ($N=10$) and high ($N=40$) traffic conditions for both cases, $1/\lambda=1$ and $1/\lambda=10$. As evident, PETA favors the selection of SCs, LAPSSs, and HAPSSs as serving BSs, rather than the gNB (contrary to the VNFPAR and BCSP), achieving significantly lower power consumption. Furthermore, it can be observed that the PETA power consumption scales more effectively with increasing load than the SotA, further supporting the need of jointly solving user association, traffic routing, and xNF placement problems to ensure truly optimal E2E real-time network performance in integrated 6G networks.

Considering the results and the aforementioned observations, the primary conclusion drawn from the simulations is the clear performance superiority of the proposed heuristic algorithm compared to the SotA. Specifically, PETA achieves substantial improvements in both evaluated scenarios, demonstrating up to 5.9 times higher energy efficiency. These findings highlight PETA's effectiveness in real-time resource allocation and its strong potential to support truly sustainable 6G 3D networks.

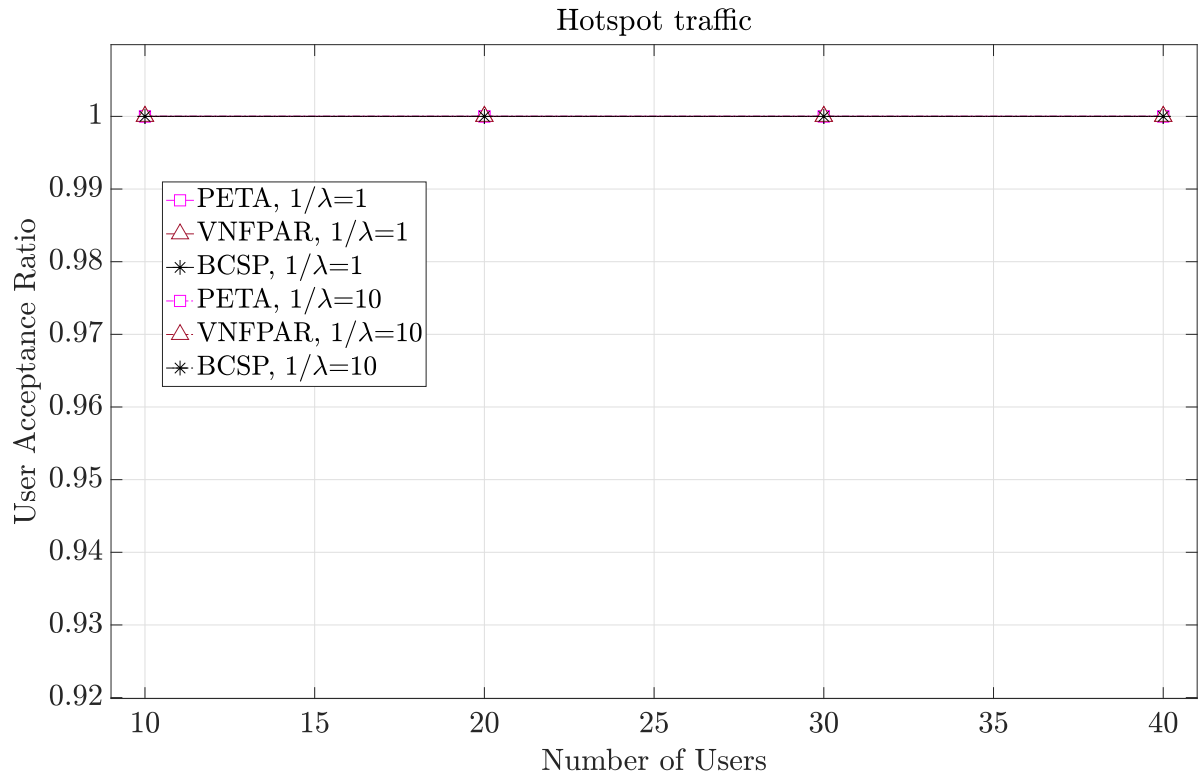


Figure 4-23: Acceptance ratio of all algorithms for different user traffic load conditions and SR arrival rates ($1/\lambda$).

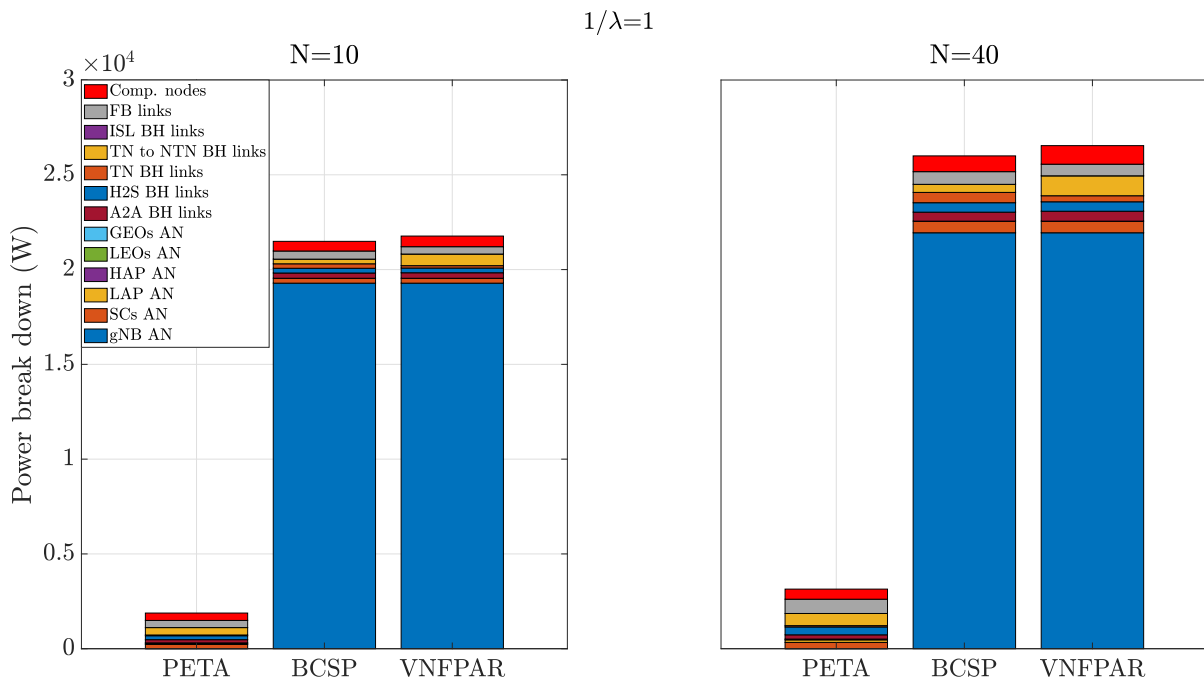


Figure 4-24: Power break-down in W of all algorithms for low and high traffic ($N=10$ and $N=40$, respectively) as well as high arrival rate ($1/\lambda=1$).

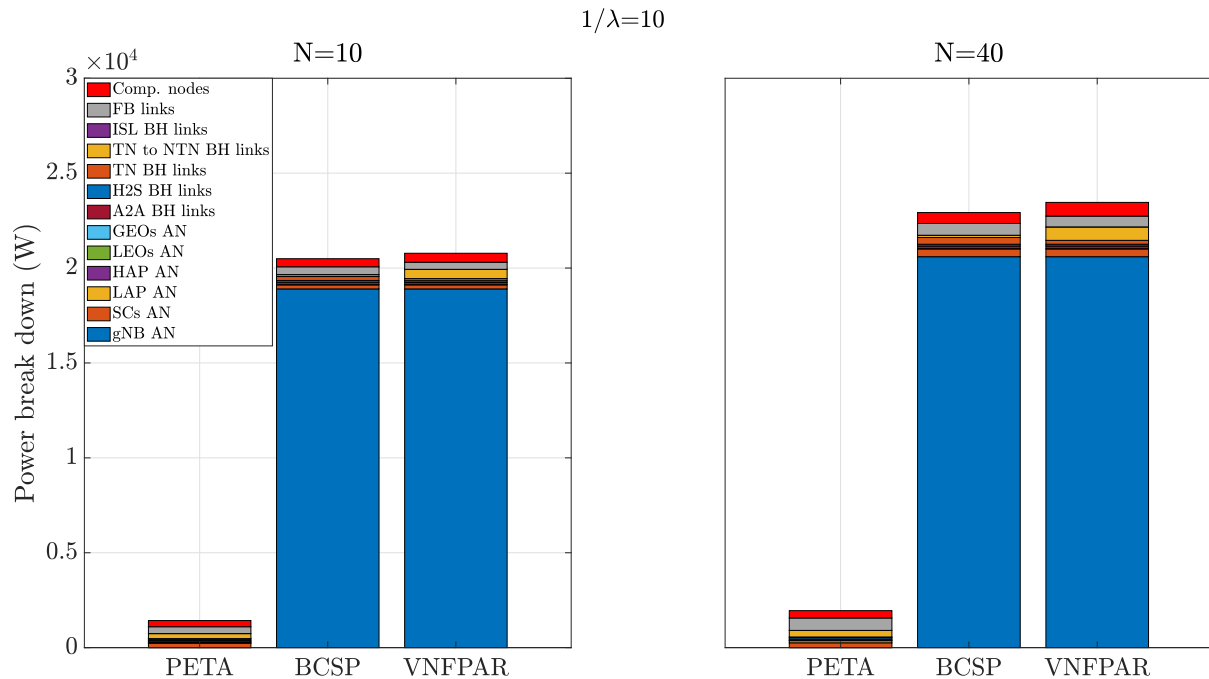


Figure 4-25: Power break-down in W of all algorithms for low and high traffic ($N=10$ and $N=40$, respectively) as well as low arrival rate ($1/\lambda=10$).

4.4 AI-BASED OFFLINE USER ASSOCIATION FOR THE 3D SUSTAINABLE ETHER NETWORK

4.4.1 Background

As already explained, user association and resource allocation have become increasingly complex problems, as aerial and space base stations (BSs) demand greater spectral resources than their terrestrial counterparts. UE needs to be assigned to BSs while considering capacity and power constraints, ensuring that Service Function Chains (SFCs) requirements are simultaneously met. Furthermore, the backhaul communication spectrum will be strategically divided: terrestrial mmWave bands, and Intersatellite Links (ISLs) that utilize higher frequencies in the terahertz (THz) range will serve wireless links.

The deployment of denser network infrastructure serves as a fundamental means for increasing system capacity. Yet this strategy introduces significant energy and economic considerations, with the access network emerging as the principal energy limitation in next-generation 6G architectures [84]. Therefore, achieving optimal energy efficiency through network densification is imperative from two primary objectives: a) minimizing operational costs (OPEX) and b) establishing environmentally sustainable network infrastructures through reduced power consumption. These evolving requirements necessitate the development of sophisticated offline resource allocation frameworks that address several key challenges.

First, they need to accommodate emerging 6G technologies like terahertz frequency bands and their inherent limitations. Second, they must ensure comprehensive end-to-end performance optimization by meeting strict latency and throughput requirements. Finally, these solutions need to prioritize network-wide energy efficiency. Therefore, the convergence of terrestrial, aerial, and space infrastructure in 6G networks demands innovative, energy-conscious approaches to user connectivity that are able to simultaneously maintain QoS standards for all users and preserve SFC integrity.

4.4.2 Motivation and Contribution

To that end, the research on terrestrial, aerial, and space integration has been extensive over the last few years, as efficient user association will be a crucial factor in ensuring reliability in 6G networks. In [85]- [86], the proposed user association solutions may suffice for the studied use cases, but they only address a subset of the TN-NTN network. In [85], the problem concerns aerial UEs that need to be accommodated by part of the TN and a single LEO satellite for Urban Air Mobility (UAM) services. Similarly, in [87], the authors focus on the aerial segment, specifically HAP-TN connectivity, where their goal is to optimize spectral efficiency while reducing interference throughout the vertical network. With the same objective, the authors in [88] propose an offloading method using Cloud-enabled HAPs capable of servicing both aerial and terrestrial users, while considering transmitted power and finite fronthaul capacity constraints. In [89], the solution for the association problem in disaster-stricken areas involves deploying not only HAPs but also UAVs as inter-mediaries between the HAPs and the terrestrial network. The research in [90] addresses maritime user services using UAVs and the LEO satellite network to meet requirements over large bodies of water, ranging from high-end users to simple data buoys, while focusing on energy and spectral efficiency. In [91], the authors employ a deep deterministic policy gradient (DDPG) to solve the data service maximization problem in a space-air-ground integrated network (SAGIN), considering LEO trajectory and power consumption. The authors of [86] also utilize a Deep Neural Network (DNN) that converges to a policy capable of handling terrestrial, aerial, and GEO satellite user association under power constraints.

In the previously mentioned works, energy efficiency cannot be guaranteed as they focus only on parts of the 3D network, having less service types and a less dynamic network that considerably simplifies the problem. In this paper, a comprehensive framework is presented that encompasses satellites at various orbital heights, HAPs, and a terrestrial urban network. In this work the pressing challenge of efficiently associating numerous base stations with diverse user needs is addressed, optimizing for energy use across the full 3D network landscape. The proposed solution is a low-complexity DNN, capable of associating UEs with the access network based on their service requirements, while minimizing network power consumption as well as avoiding violating the capacity constraints.

4.4.3 System Model and Problem Formulation

An integrated 3D TN-NTN, as depicted in Figure 4-26, is considered, consisting of the terrestrial, aerial and space nodes. In the terrestrial domain, a set of macro gNBs is considered, denoted by G , which is overlaid by a set S of Small Cells (SCs). In the aerial domain, a set of HAPs, denoted by H , is considered. In the space domain, the sets S_{LEO} , S_{MEO} and S_{GEO} for the LEO, MEO and GEO satellites are considered, respectively. A set of users, denoted by U , is also considered with Guaranteed Bit Rate (GBR) requirements equal to d_u for user u .

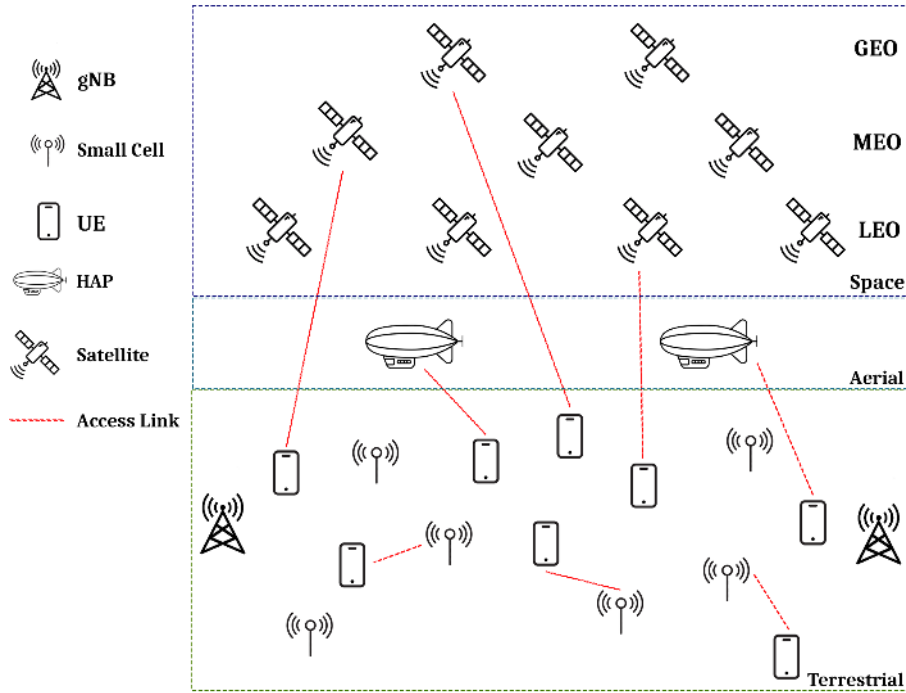


Figure 4-26: System model of the integrated terrestrial, aerial, and satellite network.

Given that the main goal of the studied problem is the minimization of total power expenditure in the Access Network, each action needs to target high energy-efficiency in the network while satisfying the user service requirements. The metric, which is measured in bits/Joule, is defined as the total throughput of the network divided by the total power consumption, and so the mixed integer linear programming (MILP) may be written as

$$\operatorname{argmin} \sum_{i \in I} p_i \quad \text{s. t. (4-5)-(4-7)} \quad (4-2)$$

where p_i is the total power consumption of each BS i (terrestrial, aerial, space) belonging to the set $I := G \cup S \cup H \cup S_{LEO} \cup S_{MEO} \cup S_{GEO}$.

For the power model of BS i , the linear approximation in [92] is considered for which both the variable RF output power p_{out_i} and the power consumption at cell i are nearly linear as

$$p_i = N_{TRX_i} (p_{0_i} + \Delta p_i \cdot p_{out_i}), \quad \forall i \in I \quad (4-3)$$

where N_{TRX_i} is the number of transceiver chains of cell i , and p_{0_i} represents the minimum non-zero output power of the transceiver of cell i ; Δp_i is the load-dependent power consumption

slope, which receives different values based on the antenna type used [92]; p_{out_i} is the transmitted power consumption of the cell i , given by

$$p_{out_i} = \frac{p_{max_i}}{c_{i_{max}}} \sum_{u \in U} (x_{(i,u)} c_{(i,u)}) \quad (4-4)$$

where p_{max_i} is the maximum transmit power of the cell i and $c_{i_{max}}$ is the maximum number of Physical Resource Blocks (PRBs) of cell i . The parameter $x_{(i,u)}$ is a binary link vector of the Access Network link (AL) between BS i and user u , which takes the value 1 when user u is connected to BS i and 0 otherwise, that is

$$x_{(i,u)} \in \{0,1\}, \quad \forall u \in U, \quad \forall i \in I \quad (4-5)$$

In addition, the parameter $c_{(i,u)}$ is the number of PRBs needed to satisfy the QoS of user u when connected to BS i . The problem is subject to additional constraints. Specifically, the capacity constraint of BS i can be expressed as

$$\sum_{u \in U} (x_{(i,u)} c_{(i,u)}) \leq c_{i_{max}}, \quad \forall i \in I \quad (4-6)$$

Furthermore, each UE u may connect to a single cell only, which is expressed as follows

$$\sum_{i \in I} x_{(i,u)} = 1, \quad \forall u \in U \quad (4-7)$$

Each UE u associated with any given cell i will be assigned the appropriate amount of PRBs, $c_{(i,u)}$, according to user demand and link state, which is calculated as

$$c_{(i,u)} = \left\lceil \frac{d_u}{BW_{PRB} SE_{(i,u)}} \right\rceil \quad (4-8)$$

where BWP RB denotes the bandwidth of a PRB and $\lceil \cdot \rceil$ is the ceiling operator, $SE_{(i,u)}$ is the maximum achievable spectrum efficiency with effective signal-to-interference-plus-noise ratio ($SINR_{(i,u)}$), given by [93], when user u connects to BS i .

4.4.4 Proposed RL-algorithm

The algorithm used to train the DNN is Proximal Policy Optimization (PPO), used for environments with discrete or continuous action spaces. The algorithm alternates between sampling data through the features fed through the environment and optimizing a clipped surrogate objective function using stochastic gradient descent, while the actor estimates a

stochastic policy, the critic estimates the value of the policy. The objective function clips the policy values to improve training stability. Based on [94]

$$L^{CLIP}(\theta) = \hat{E}[\min(r_t(\theta)\widehat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\widehat{A}_t)] \quad (4-9)$$

in which θ are the parameters of the actor that are randomly initialized and tuned during training thus improving the policy. $\hat{E}$ are the experiences over multiple episodes following the current policy, specifically S is the current state observation from which an action A is chosen by the actor using the policy $\pi_\theta(a_t|s_t)$. The reward R is observed and the next observation S' , along with the action and the last observation, is stored as an experience $E = (S, A, R, S')$.

The value $r_t(\theta)$ is the new policy $\pi_\theta(a_t|s_t)$ over the old policy $\pi_{\theta_{old}}(a_t|s_t)$ that is multiplied by the advantage function estimator $\widehat{A}_t$, which in this problem is the Generalized Advantage Estimator (GAE) [95]. GAE computes the advantage function D_t (discounted sum of temporal difference errors)

$$D_t = \sum_{k=t}^{t+N-1} (\gamma\lambda)^{k-t} \delta_k \quad (4-10)$$

where t is a single step, γ is the discount factor, λ is the GAE factor, and δ_k is the temporal difference error. Finally, ϵ is the clip factor which clips the gradients to improve stability during training.

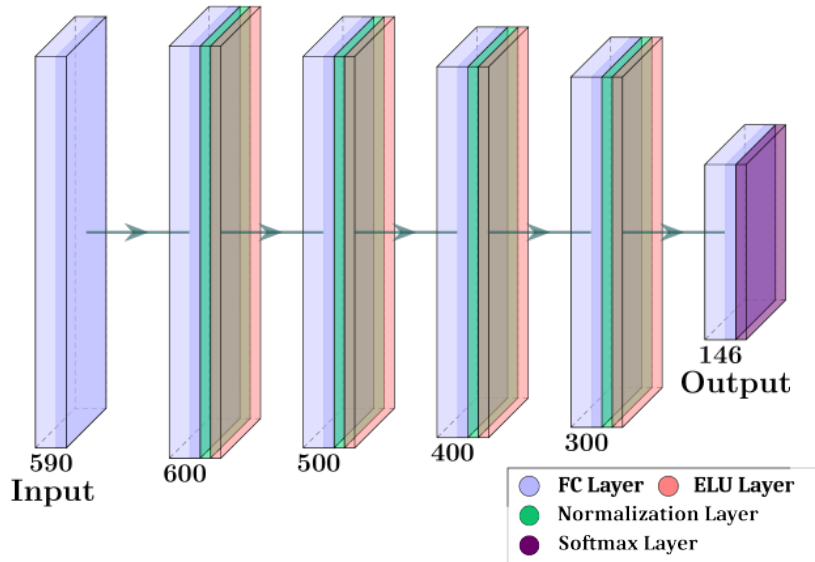


Figure 4-27: Actor network layer visualization.

The DNN architecture that is selected to solve the aforementioned problem is shown in Figure 4-27. The selected neuron count is shown to sufficiently capture the full space of observations and actions within the network. As for the DNN input, certain observations have been picked that proved sufficient enough to accomplish the energy efficiency goal, in a network of 146

cells, without bloating the input layer with unneeded information. The following 590 features were selected as input of the network:

- 146 binary flags to track which base station is currently active or not.
- 146 calculations of the needed PRBs for a certain user to connect to a certain cell, normalized and bounded between 0 and 1.
- 146 calculations of the remaining PRBs on the selected cell, normalized between -1 and 0.
- 146 calculations of the power per resource block multiplied by Δp_i and N_{TRX_i} for each base station, based on its type (terrestrial, aerial, space), and normalized between 0 and 1.
- 6 idle power calculations for each type of base station, normalized between 0 and 1.

Feature normalization is used for training stability and uniform feature importance, as non-normalized values would have biased the agent's decision-making. The only non-normalized values are the 146 binary flags, which do not require normalization due to their binary nature.

During training, three different activation functions are tested, ReLU, Swish and ELU; ReLU provided stability but converged more slowly than the other two, and Swish is able to converge faster than ReLU but created non-stable rewards, effectively never reaching the potential of the network. Among them, ELU has the best results, as it converged quickly to the best rewards across different scenarios while keeping the learning stable enough to avoid dips during training. Finally, having a bottleneck architecture (lessening neurons in each layer), helps the DNN have a higher understanding of the problem at hand, as it needs to mix the results of earlier layer neurons to the later layer neurons. This architecture forces the DNN to have a more abstract understanding of the problem and its solutions.

Hard constraints are implemented to enable the DNN to reach its energy efficiency objective while preserving resource availability in each cell. During each training step, once a cell is selected, the step function verified whether this action would result in resource depletion. If the selected cell retains zero or more PRBs after the action, the step function proceeds normally to the reward function. Otherwise, a flag is triggered, and a standard penalty of -1 is returned as an experience alongside the state-action-next state tuple (S, A, S') .

The Reward function is structured so that the agent reaches the energy efficiency goal while minimizing the power consumption of the cell network, as follows

$$R_s = 1 - \frac{\Delta P_s - P_{i_{min}}}{P_{i_{max}} - P_{i_{min}}}, \quad R_s \in [0,1] \quad (4-11)$$

where R_s is the step reward, ΔP_s is the power increase of the step, $P_{i_{min}}$ and $P_{i_{max}}$ are the minimum and maximum power consumption values, respectively. Furthermore, R_s is bound between 0 and 1, so that the rewards are also normalized.

4.4.5 Performance Evaluation

In the current section, the results of the agent during a multi-phase training period are presented, along with the performance evaluation across a 100-scenario simulation.

Table 4-3 Parameter values used in all training phases.

Hyperparameter	Value
Discount Factor (γ)	0.99
Experience Horizon	250
Batch Size	65
Entropy Loss Weight	0.01
Actor Learn Rate	10^{-5}
Critic Learn Rate	$2 \cdot 10^{-5}$
GAE Factor	0.95
Clip Factor	0.2

The training and simulation results are created in MATLAB R2021a, leveraging its Reinforcement Learning Toolbox. For the DNN's convergence, the technique of curriculum learning is employed, which uses a three-phase training. The first training is employed for the model to learn the basic principles of user association in a network, the second is to create UE diversity and start learning that different sets of services need different solutions, and the third and final training is to see as many different states of the network as possible.

Table 4-3 shows the hyperparameter values that were used during the multi-phase training. The discount factor γ is a scalar value between 0 and 1 that determines how much the actor will prioritize future rewards, with values near 0 prioritizing present rewards, while values near 1 prioritizing future rewards. The experience horizon is the maximum number of time steps that an agent can collect experience for during a single episode, while the batch size is a random set of 65 experiences from the total of 250 that will be used to update the agent's policy. Entropy loss weight, which is specified as a scalar value between 0 and 1, is a parameter that promotes agent exploration by applying a penalty for being too certain about which action to take. Doing so can help the agent move out of local optima. Learning rates determine how much the actor and critic networks' parameters are adjusted based on the computed gradients, higher rates tend to converge faster, but may provide suboptimal results. GAE factor controls the bias-variance tradeoff in advantage estimation by smoothing advantage estimates across different time horizons. A value of 0.95 heavily weights longer-term information while incorporating shorter-term corrections, creating more stable learning signals than using immediate or fully discounted rewards alone. The clip factor ϵ is used in PPO to limit how much the policy can change in a single update, preventing large policy changes that could destabilize training.

Table 4-4 Simulation parameters for different access network parts.

Part of the network	Macro gNB	SC	HAP	SN
f (GHz)	2	2	28	28
BW (GHz)	0.02	0.02	0.2	0.4
N_{TRX_i}	128	32	64	128
P_{0_i} (W)	145	6.8	1	1
Δ_{p_i}	4.2	4	2	1

P_t (dBm)	46	38	40	60
L_{Tx} (dB)	1	1	1	1
G_{Tx} (dBi)	23	20	32	36
N_F (dB)	7	7	7	7
G/T (dB/K)	8.5	8.5	8.5	8.5

Table 4-4 presents the simulation parameters used to model different components of the heterogeneous access network architecture. The network consists of four distinct parts: a macro gNB, Small Cells, High Altitude Platforms, and a Satellite Network (SN). The table shows significant variation in system characteristics across the access network. Each component operates at different frequency bands, with the macro gNB and the SCs at 2 GHz and the other 3D network components at 28 GHz, with a bandwidth of 0.02 GHz, 0.2 GHz, and 0.4 GHz for the Terrestrial, Aerial, and Satellite parts respectively. P_t is the transmission power in dBm, L_{Tx} and G_{Tx} are the losses and gains of the antenna chains measured in dB and dBi, respectively. N_F is the noise figure of the receivers measured in dB, and G/T is the antenna gain-to-noise-temperature ratio, which is measured in dB/K.

During the first training (Figure 4-28), the model is subjected to 4000 episodes of the same scenario: 146 cells, specifically 1 gNB, 8 SCs (2 clusters of 4), 1 HAP, 134 LEO satellites, and 2 GEO satellites, while 40 UEs with random topology and services were created. The rewards show a small upward trend, especially after 3500 episodes, where a tendency toward convergence is indicated. However, due to the limited diversity in the dataset, comprising only 40 UEs with a fixed deployment scenario and services, the learning process is inherently constrained, preventing faster or more robust generalization.

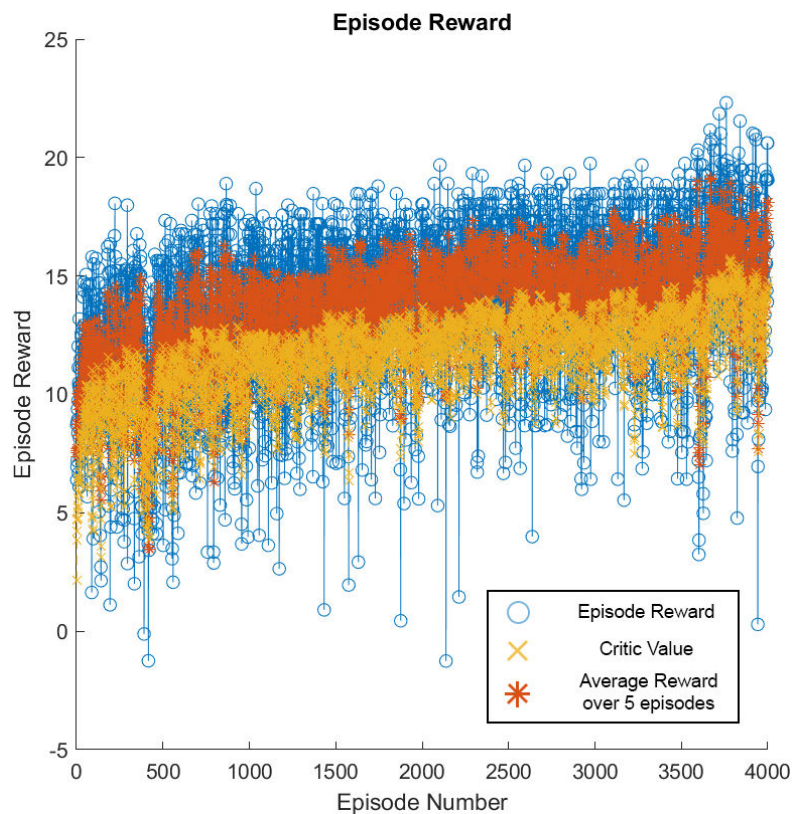


Figure 4-28: First DNN training over 4000 same iteration scenarios.

In the second training phase (Figure 4-29), it maintains the same base station topology and functionality, however, the 40 UEs vary in both position and service requirements across these 10 scenarios. To ensure effective training, the scenarios were replicated 1000 times and were shuffled, which prevents the agent from overfitting and promotes convergence to a more generalized policy. Compared to the first training phase, the reward gains exhibit substantial improvement with an almost linear upward trajectory. There is a substantial reduction in power consumption across the episodes of this training phase, as shown in Figure 4-30.

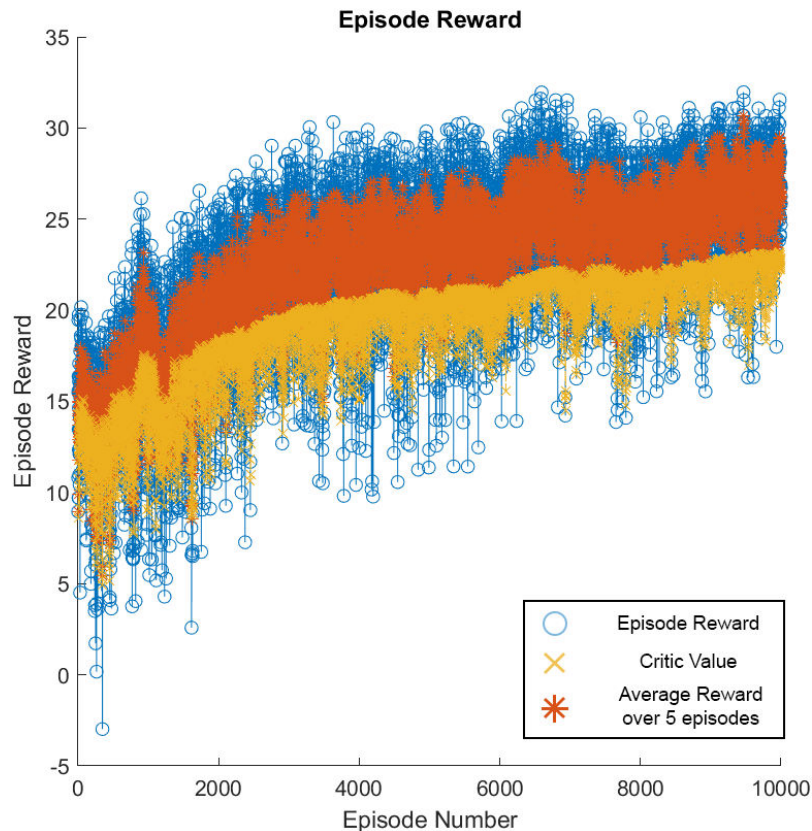


Figure 4-29: Second DNN training over 10 different scenarios. Each scenario has different UE services and topologies. The scenarios are multiplied by 1000 and shuffled to avoid overfitting.

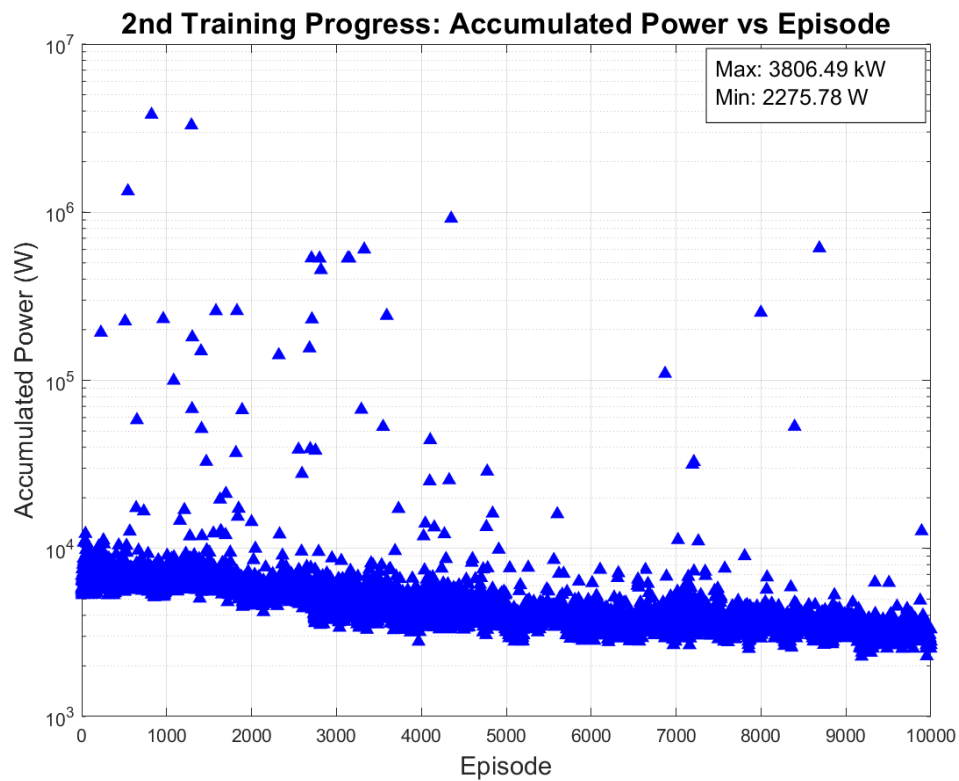


Figure 4-30: Power consumption reduction across 2nd training's episodes.

For the third and final training phase (Figure 4-31), the same network architecture and cell configuration remained unchanged. The 40 user equipment units are assigned varying locations and service requirements across 9900 distinct episodes. These scenarios are then tripled and randomly reordered to ensure a more uniform distribution of learning experiences. The purpose of this training approach is to enable the agent to preserve the strong performance achieved in the initial 10 scenarios while developing a broadly generalized policy capable of handling 9900 diverse scenarios.

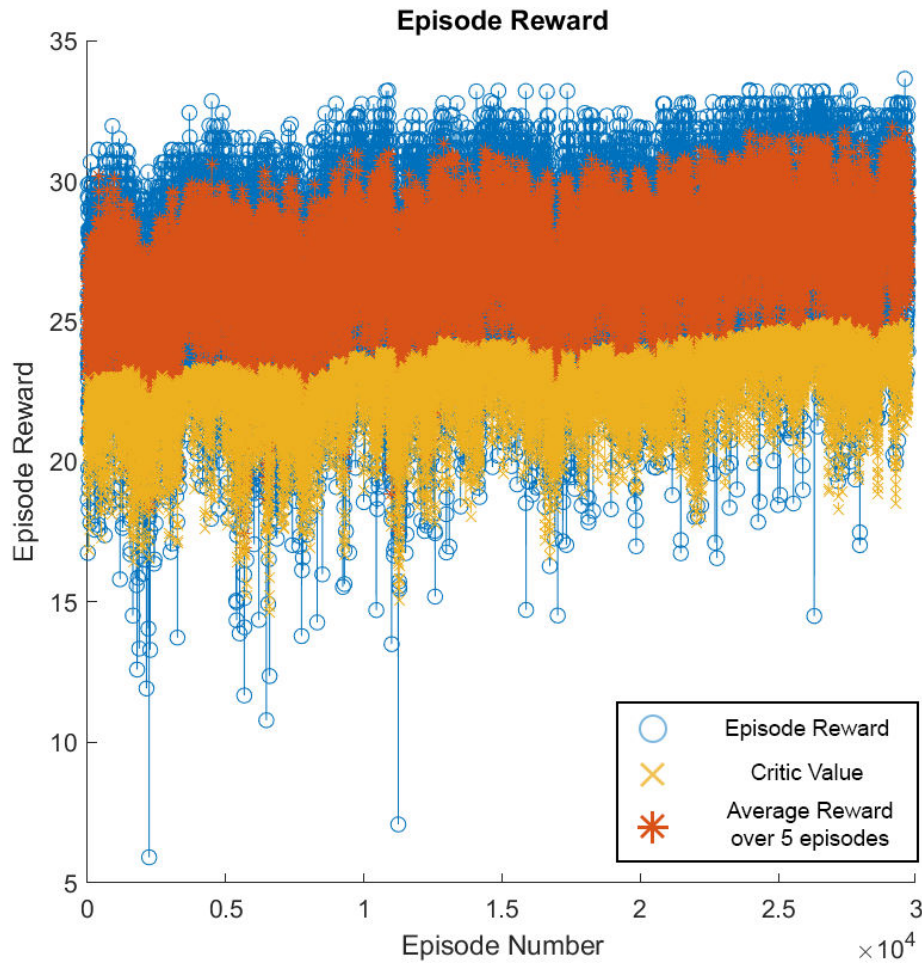


Figure 4-31: Third DNN training over 9900 different scenarios. Each scenario has different UE services and topologies. The scenarios are multiplied by 3 and shuffled.

4.4.6 Simulation Results

The effectiveness of the selected agent through 100 scenarios (Figure 4-32) of different UE positions and services is evaluated, while keeping the network topology unchanged. The mean reward returned is 29.9414 while the deviation is only ± 1.1467 . The current algorithm is compared against two SotA solutions: i) the Reference Signal Received Power (RSRP) algorithm [96], which is the default user association criterion that associates the user based on the highest received signal strength, and ii) the TERA algorithm that is developed in [75], which is a heuristic algorithm that associates the users while minimizing the incremental power consumption in the network. The mean power consumption over these 100 scenarios is $2.5707 \cdot 10^3 W$, significantly lower than the RSRP solution, which, using the same scenarios, resulted in a mean power consumption of $4.9198 \cdot 10^3 W$. At the same time, the TERA heuristic algorithm scored higher than the RSRP, but still lower than the DNN, having a mean of $3.7779 \cdot 10^3 W$, as shown in Figure 4-33. As a result, the proposed solution has been proven to be up to 47.7% more energy efficient than the RSRP algorithm, and 31.9% more than the TERA heuristic algorithm.

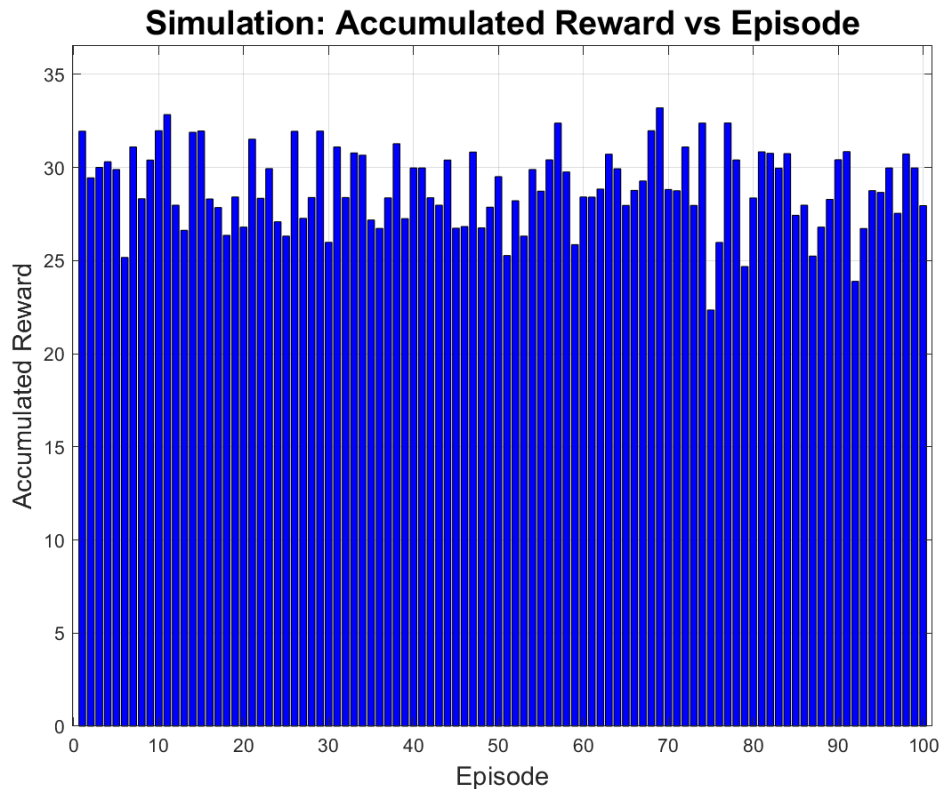


Figure 4-32: Simulation over 100 different scenarios. The 40 UEs have different topologies and services, while the cell network stays the same. The agent isn't trained on these specific 100 scenarios.

Concluding, a reinforcement learning-based approach for energy-efficient user association in integrated terrestrial and non-terrestrial 6G networks was presented. By developing a Deep Neural Network trained with Proximal Policy Optimization, the complex challenge of maximizing energy efficiency of a heterogeneous network was tackled, comprising terrestrial base stations, HAPs, and LEO satellites at multiple orbital heights. The proposed approach demonstrates significant improvements over current SotA methods, achieving significant improvements in energy efficiency while meeting user service requirements. The three-phase curriculum learning strategy proved essential for the DNN to effectively learn the intricate relationships between PRB allocation and energy consumption, ultimately reducing power consumption by up to 47.7% compared to the SotA approaches.

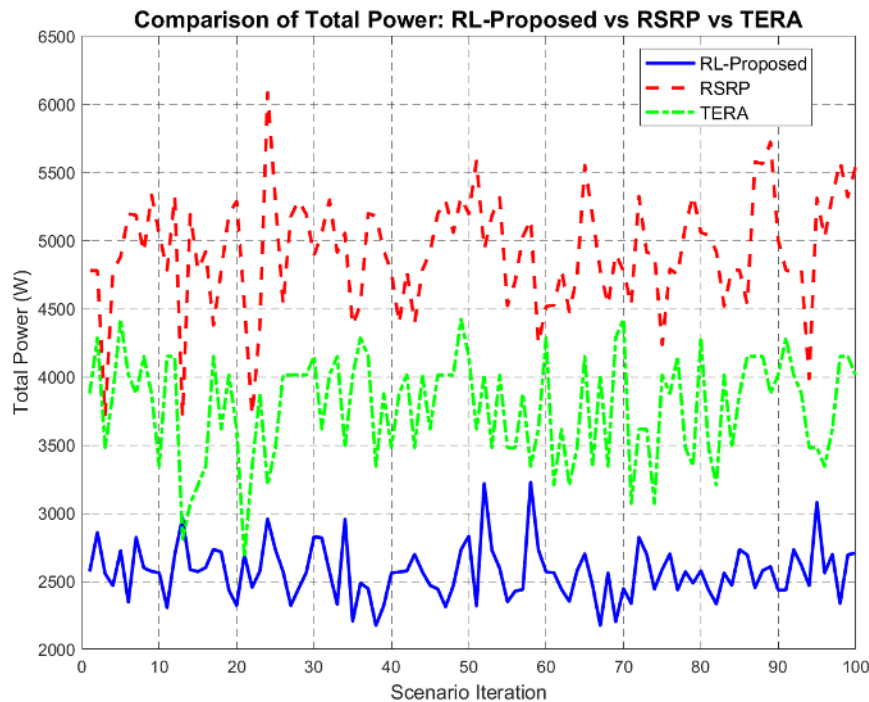


Figure 4-33: Power consumption comparison among RL-Proposed, RSRP and TERA algorithms.

4.5 AI-DRIVEN BEAMFORMING FOR INTEGRATED SENSING AND COMMUNICATION IN TERRESTRIAL-SATELLITE NETWORKS

4.5.1 Background

The exponential growth in wireless data traffic, driven by the proliferation of multimedia services and bandwidth-intensive applications, has led to severe spectrum scarcity. At the same time, the ambition to achieve global wireless coverage is hindered by the economic and logistical challenges of deploying terrestrial infrastructure in remote and underprivileged regions. These limitations necessitate innovative solutions that can simultaneously address spectrum efficiency and coverage gaps.

Integrated terrestrial and satellite networks (ITSNs) have emerged as a promising architecture to tackle these challenges. By enabling spectral coexistence between terrestrial and satellite systems, ITSNs can enhance both spectral and energy efficiency while ensuring seamless connectivity across diverse geographical areas [97]. Their performance can be further enhanced with distributed MIMO in the terrestrial network, which mitigates inter-cell interference through distributed access points that cooperatively serve users, improving service uniformity and system capacity [98].

The integration of ISAC into this framework adds another layer of functionality and efficiency. ISAC enables the dual use of wireless signals for both communication and sensing, allowing for cost-effective deployment of applications such as environmental monitoring, drone and aircraft tracking, and remote sensing [99]. This convergence of communication and sensing not only optimises spectrum usage but also supports a wide range of mission-critical and data-driven services in next-generation networks.

Complementing these technologies is RSMA, a novel multiple access scheme that has demonstrated superior performance in ISAC-enabled systems. RSMA enhances spectral and

energy efficiency by flexibly managing interference and exploiting the common message component for sensing purposes, thereby eliminating the need for dedicated radar signals [100]. This makes RSMA particularly suitable for ITSNs, where efficient resource use and interference mitigation are critical.

4.5.2 Motivation and Contribution

The practical deployment of joint communication and sensing systems faces significant challenges due to the complexity of their joint optimization. Traditional optimization techniques often suffer from high computational complexity and are generally unable to respond in real time to rapidly changing channel conditions. Furthermore, these methods face significant limitations in dynamically managing the intricate trade-offs among spectral efficiency, sensing accuracy, and power consumption.

To address these challenges, AI-based beamforming design has emerged as a transformative approach. By leveraging deep learning models, AI can effectively learn and model complex channel dynamics, significantly reducing the computational burden associated with traditional optimization techniques. This enables fast, real-time adaptation of beamforming and resource allocation strategies in response to dynamic channel conditions and system requirements. This data-driven approach improves system responsiveness and energy efficiency, supporting the development of intelligent and globally connected wireless networks.

Motivated by these insights, this work investigates an AI-driven beamforming design is investigated in this work for ITSN with ISAC. Specifically, a LEO satellite interacting with terrestrial distributed MIMO system is considered to perform simultaneous sensing and communication. The system operates under a cognitive radio paradigm, allowing spectrum sharing in the same frequency band, and employs RSMA to enhance spectral efficiency and interference management. The objective is to maximise the system sum rate while satisfying power, sensing and interference constraints. The main contributions can be summarised as follows:

- A cognitive satellite–terrestrial network architecture is proposed, in which a multi-antenna LEO satellite shares the licensed spectrum of terrestrial distributed MIMO systems to simultaneously perform target sensing and provide communication services to satellite users using RSMA. The spectrum sharing is designed to ensure that satellite transmissions do not exceed an interference threshold for terrestrial operations.
- A weighted sum-rate maximisation problem is formulated by jointly optimizing the common precoder vector, common rate variables, and power allocation to satellite users subject to constraints on the power budget of the satellite, a minimum beam pattern gain in the direction of the sensing target, and a maximum interference threshold for terrestrial users. To solve the resulting non-convex problem, a hybrid solution is proposed, in which a deep CNN-based model is used for power allocation, and a semidefinite relaxation (SDR)-based method is applied for common precoding and rate optimization.
- Numerical simulations are conducted to demonstrate that all imposed constraints are satisfied by the proposed scheme, which achieves performance close to that of a near-optimal successive convex approximation (SCA)-based solution developed for comparison purposes. In addition to maintaining high performance, the proposed AI-based approach significantly reduces computational time compared to the SCA-based benchmark. Furthermore, the RSMA-based approach is compared with conventional techniques employing maximum ratio transmission (MRT) and zero-forcing (ZF) precoding. The results show that the RSMA-based solution substantially improves the sum rate compared to these baseline methods.

4.5.3 System Model

A LEO satellite equipped with N antennas is considered to provide service to K satellite users (SUs) and perform sensing to a target, as illustrated in Figure 4-34: ISAC-enabled integrated satellite and terrestrial distributed MIMO system.. The terrestrial network consists of B BSs, each equipped with N_B antennas, and M terrestrial users (TUs). Each BS serves all TUs within the same time-frequency resource block, coordinated by a CPU via backhaul links. The underlying cognitive radio is applied to enable the satellite system to share the licensed spectrum of terrestrial networks while ensuring that interference to terrestrial operations does not exceed a defined threshold.

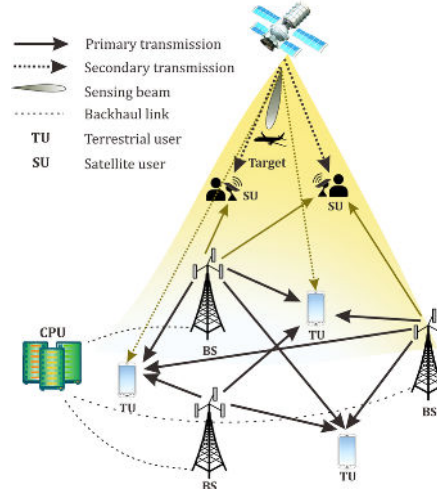


Figure 4-34: ISAC-enabled integrated satellite and terrestrial distributed MIMO system.

At the satellite, RSMA is employed to enable simultaneous sensing and communication. Under the one-layer RSMA scheme, each user message m_k is divided into common part m_k^c and a private part m_k^p . The common parts are encoded together into a common stream s_0 to be decoded by all users, while the private parts are encoded independently into private streams s_k , as illustrated in Figure 4-35. Then, linear precoding is applied to all the streams, where $\mathbf{p}_0$ is common precoder for s_0 , and $\mathbf{p}_k$ is private precoder for s_k .

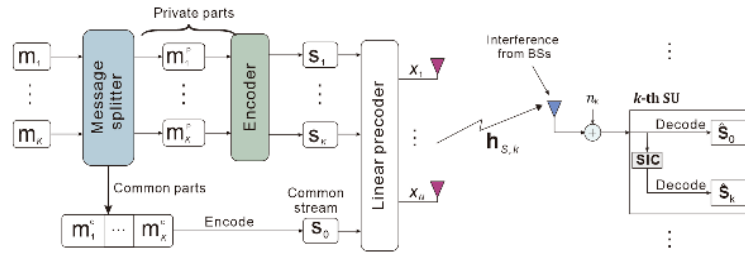


Figure 4-35: Illustration of the RSMA framework.

It has been shown that the common signal in RSMA can be exploited for sensing purposes and it is able to replace the dedicated radar signal [100]. Then, the transmit signal at the satellite is given by

$$\mathbf{x}_S = \mathbf{p}_0 s_0 + \sum_{k=1}^K \mathbf{p}_k s_k. \quad (4-12)$$

In the terrestrial network, local zero-forcing precoding is employed by the BSs to transmit the stream s_m^{TU} to the m -th TU. Hence, the transmit signal at the b -th BS is denoted as

$$\mathbf{x}_{BS,b} = \sum_{m=1}^M \sqrt{q_{b,m}} \mathbf{w}_{T,b,m} s_m^{TU}, \quad (4-13)$$

where $q_{b,m}$ denotes the power allocated to the m -th TU at the b -th BS, and $\mathbf{w}_{T,b,m}$ is the corresponding precoding vector. The precoding matrix at the b -th BS is given by $\mathbf{W}_{T,b} = [\mathbf{w}_{T,b,1}, \dots, \mathbf{w}_{T,b,M}] = \left[\frac{\tilde{\mathbf{w}}_{T,b,1}}{\|\tilde{\mathbf{w}}_{T,b,1}\|}, \dots, \frac{\tilde{\mathbf{w}}_{T,b,M}}{\|\tilde{\mathbf{w}}_{T,b,M}\|} \right]$, where $\tilde{\mathbf{W}}_{T,b} = [\tilde{\mathbf{w}}_{T,b,1}, \dots, \tilde{\mathbf{w}}_{T,b,M}] = \mathbf{G}_{T,b} (\mathbf{G}_{T,b}^H \mathbf{G}_{T,b})^{-1}$ and $\mathbf{G}_{T,b} = [\mathbf{g}_{T,b,1}, \dots, \mathbf{g}_{T,b,M}] \in \mathbb{C}^{N_B \times M}$ with $\mathbf{g}_{T,b,m} \in \mathbb{C}^{N_B \times 1}$ representing the channel vector from the b -th BS to the m -th TU.

In the satellite network, the received signal at the k -th SU is given by

$$y_k = \mathbf{h}_{S,k}^H \mathbf{p}_0 s_0 + \sum_{k'=1}^K \mathbf{h}_{S,k}^H \mathbf{p}_{k'} s_{k'} + \sum_{b=1}^B \sum_{m=1}^M \mathbf{g}_{S,b,k}^H \sqrt{q_{b,m}} \mathbf{w}_{T,b,m} s_m^{TU} + n_k, \quad (4-14)$$

where $\mathbf{h}_{S,k} \in \mathbb{C}^{N \times 1}$ is the channel vector from the satellite to the k -th SU, $\mathbf{g}_{S,b,k} \in \mathbb{C}^{N_B \times 1}$ is the channel from the b -th BS to the k -th SU, and n_k is the additive white Gaussian noise with zero mean and variance σ_k^2 . First, each k -th SU decodes the common stream s_0 by treating all the private streams as noise, where the achievable rate at the k -th SU is given by

$$R_k^c = \log_2 \left(1 + \frac{|\mathbf{h}_{S,k}^H \mathbf{p}_0|^2}{\sum_{k'=1}^K |\mathbf{h}_{S,k}^H \mathbf{p}_{k'}|^2 + \sum_{b=1}^B \sum_{m=1}^M q_{b,m} |\mathbf{g}_{S,b,k}^H \mathbf{w}_{T,b,m}|^2 + \sigma_k^2} \right). \quad (4-15)$$

Next, the k -th SU uses successive interference cancellation (SIC) to remove the interference of the common message from the received signal and decodes its private stream with an achievable rate denoted as

$$R_k^p = \log_2 \left(1 + \frac{|\mathbf{h}_{S,k}^H \mathbf{p}_k|^2}{\sum_{k'=1, k' \neq k}^K |\mathbf{h}_{S,k}^H \mathbf{p}_{k'}|^2 + \sum_{b=1}^B \sum_{m=1}^M q_{b,m} |\mathbf{g}_{S,b,k}^H \mathbf{w}_{T,b,m}|^2 + \sigma_k^2} \right). \quad (4-16)$$

To ensure that the common stream s_0 can be successful decoded by all the SUs, its rate must not exceed $R_0 = \min(R_1^c, \dots, R_K^c)$. Since R_0 consists of the common parts of the K SU's messages, it can be expressed as $R_0 = \sum_{k=1}^K c_k$, where c_k denotes the common rate associated with the k -th user's common message part m_k^c . Hence the total achievable rate of the k -th SU is defined by $R_k = c_k + R_k^p$.

The interference from the satellite to each m -th TU can be expressed as

$$I_m = |\mathbf{h}_{T,m}^H \mathbf{p}_0|^2 + \sum_{k=1}^K |\mathbf{h}_{T,m}^H \mathbf{p}_k|^2, \quad (4-17)$$

where $\mathbf{h}_{T,m} \in \mathbb{C}^{N \times 1}$ is the channel vector from the satellite to the m -th TU.

For radar sensing at the satellite, both common and private precoding vectors are utilized to simultaneously perform target detection and data transmission. The radar sensing

performance is evaluated using the transmit beampattern metric, defined as

$$P(\theta) = \mathbf{a}^H(\theta) \left(\mathbf{p}_0 \mathbf{p}_0^H + \sum_{k=1}^K \mathbf{p}_k \mathbf{p}_k^H \right) \mathbf{a}(\theta), \quad (4-18)$$

where $\mathbf{a}(\theta) = [1, e^{j2\pi d_\Delta \sin(\theta)}, \dots, e^{j2\pi(N-1)d_\Delta \sin(\theta)}]^T$ is the steering vector of direction θ , with d_Δ denoting the normalized spacing between adjacent antenna elements relative to the carrier wavelength. The desired beam pattern is designed to maximise the minimum beampattern gain in the direction of the target, where θ represents the potential target location [101].

4.5.4 Problem Formulation

The objective is to maximize the weighted sum rate of the SUs, subject to a power budget constraint at the satellite, a minimum beampattern gain requirement for target sensing, and a maximum allowable interference toward the TUs. The proposed optimization problem is formulated as follows:

$$\max_{\mathbf{p}_0, \mathbf{p}_k, c_k} \sum_{k=1}^K u_k (R_k^p + c_k) \quad (4-19)$$

$$\text{s.t. C1: } \sum_{k'=1}^K c_{k'} \leq R_{k'}^c, \forall k \quad (4-20)$$

$$\text{C2: } P(\theta) \geq G_{\min} \quad (4-21)$$

$$\text{C3: } \|\mathbf{p}_0\|^2 + \sum_{k=1}^K \|\mathbf{p}_k\|^2 \leq P_{\max} \quad (4-22)$$

$$\text{C4: } I_m \leq I_{\max}, \forall m \quad (4-23)$$

$$\text{C5: } c_k \geq 0, \forall k, \quad (4-24)$$

where u_k denotes the rate weight for the k -th SU, $G_{\min}$ represents the minimum beampattern gain in the direction of the target, $P_{\max}$ is the satellite's total power budget, and $I_{\max}$ is the maximum allowed interference to TUs. Constraint C1 ensures that s_0 can be successfully decoded by all users. Constraint C2 enforces the radar requirement by guaranteeing a minimum beampattern gain, while Constraint C4 limits the interference introduced to the terrestrial network. The proposed problem (4-19) is non-convex. To address this, a hybrid solution combining a CNN-based model with an SDR-based method is proposed.

4.5.5 Proposed Solution

The design for the private precoding vectors $\{\mathbf{p}_1, \dots, \mathbf{p}_K\}$ is based on ZF. Then, the private precoding is defined as $\mathbf{p}_k = \alpha_k \frac{\tilde{\mathbf{p}}_k}{\|\tilde{\mathbf{p}}_k\|}$, where α_k^2 represents the transmit power and for ZF, $\tilde{\mathbf{p}}_k$ is the k -th column of $\tilde{\mathbf{P}}_S = \tilde{\mathbf{H}}_S (\tilde{\mathbf{H}}_S^H \tilde{\mathbf{H}}_S)^{-1} = [\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_K] \in \mathbb{C}^{N \times K}$ with $\tilde{\mathbf{H}}_S = [\mathbf{h}_{S,1}, \dots, \mathbf{h}_{S,K}] \in \mathbb{C}^{N \times K}$. Next, the non-convex problem (4-19) is divided into two subproblems. First, a CNN-based scheme is developed to optimise the power allocation variables $\{\alpha_k\}$. Subsequently, given $\{\alpha_k\}$, an SDR-based method is proposed to optimise the common precoder $\mathbf{p}_0$ and the common rate variables $\{c_k\}$.

A. CNN-based model to predict $\{\alpha_k\}$

The proposed CNN-based model is composed of five inputs and one output representing the power allocation variables $\{\alpha_k\}$, as illustrated in Figure 4-36. The first inputs are the ZF based matrix $\tilde{\mathbf{P}}_S$ which consists of real and imaginary components. These components are treated as separate image channels, allowing the matrix to be represented as an image-like input. The second input is the channel matrix $\tilde{\mathbf{H}}_S$ from the satellite to the SUs, which can also be considered as an image with two image channels. The third input is the correlation between private precoders and the channels associated to TUs, defined by $\mathbf{C}_{\mathbf{P}_S, \mathbf{H}_T} = |\mathbf{P}_S^H \mathbf{H}_T| \in \mathbb{R}^{K \times M}$, where $\mathbf{H}_T = \begin{bmatrix} \mathbf{h}_{T,1} \\ \vdots \\ \mathbf{h}_{T,M} \end{bmatrix}$ and $\mathbf{P}_S = \begin{bmatrix} \tilde{\mathbf{p}}_1 \\ \vdots \\ \tilde{\mathbf{p}}_K \end{bmatrix}$. The fourth input is the correlation between private precoders, and the steering vector of the angle associated with the direction of the target, which can be represented as $\mathbf{c}_{\mathbf{P}_S, \theta} = |\mathbf{P}_S^H \mathbf{a}(\theta)| \in \mathbb{R}^{K \times 1}$. The last input is the power budget of the satellite.

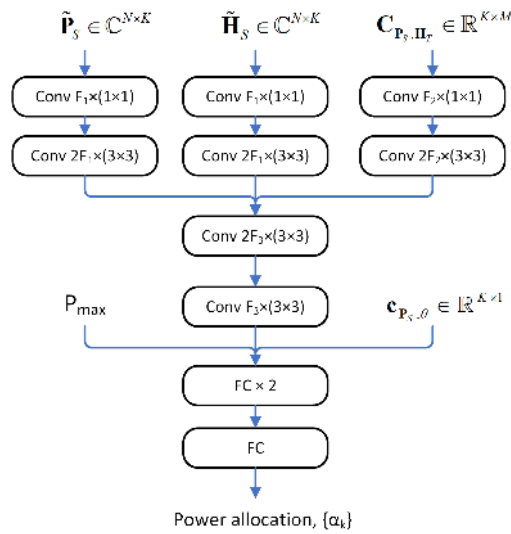


Figure 4-36: Proposed CNN-based model for power allocation.

In Figure 4-36, the Conv $w \times (b \times b)$ module is composed three sequential layers. First, a convolutional layer with w filters and a kernel size of $b \times b$ is applied. This is followed by a batch normalisation layer, which accelerates training and enhances stability. Finally, a Parametric ReLU (PReLU) activation function is used to introduce non-linearity, allowing for adaptive learning of the negative slope. The model is trained using the Adam optimiser and compiled with mean squared error (MSE) as both the loss function and evaluation metric, making it well-suited for regression tasks. The Conv module enables feature extraction from complex-valued inputs, effectively capturing and consolidating this information in a high-dimensional feature space. These extracted features are then merged with real-valued inputs, such as power levels and directional correlation toward the target. The combined representation is subsequently passed through two fully connected (FC) layers to generate the power allocation variables $\{\alpha_k\}$.

B. SDR-based method to optimise $\mathbf{p}_0$ and $\{c_k\}$.

In this subsection, the SDR-based method is described, given the power allocation variables $\{\alpha_k\}$, to optimise $\mathbf{p}_0$ and $\{c_k\}$. First, $\mathbf{P}_0 = \mathbf{p}_0 \mathbf{p}_0^H$, $\mathbf{H}_{S,k} = \mathbf{h}_{S,k} \mathbf{h}_{S,k}^H$, $\mathbf{H}_{T,m} = \mathbf{h}_{T,m} \mathbf{h}_{T,m}^H$, and $\eta_{k,k'} = \frac{\mathbf{h}_{S,k}^H \tilde{\mathbf{p}}_{k'}}{\|\tilde{\mathbf{p}}_{k'}\|}$ are defined. Then, it implies that the matrix $\mathbf{P}_0$ is rank-one symmetric positive semidefinite, i.e., $\mathbf{P}_0 \succeq 0$ and $\text{rank}(\mathbf{P}_0) = 1$. Moreover, the following equivalences hold:

$\|\mathbf{p}_0\|^2 = \text{Tr}(\mathbf{P}_0)$, $|\mathbf{h}_{S,k}^H \mathbf{p}_0|^2 = \text{Tr}(\mathbf{H}_{S,k} \mathbf{P}_0)$, and $|\mathbf{h}_{T,m}^H \mathbf{p}_0|^2 = \text{Tr}(\mathbf{H}_{T,m} \mathbf{P}_0)$. Then, by relaxing the rank-one constraint on $\mathbf{P}_0$, the original non-convex problem (4-19), given $\{\alpha_k\}$, can be reformulated into a convex optimisation problem as follows:

$$\max_{\mathbf{P}_0, c_k} \sum_{k=1}^K u_k c_k \quad (4-25)$$

subject to

$$-\text{Tr}(\mathbf{H}_{S,k} \mathbf{P}_0) + \left(2^{\sum_{k'=1}^K c_{k'}} - 1\right) \left(\sum_{k'=1}^K \alpha_{k'}^2 |\eta_{k,k'}|^2 + \sum_{b=1}^B \sum_{m=1}^M q_{b,m} |\mathbf{g}_{S,b,k}^H \mathbf{w}_{T,b,m}|^2 + \sigma_k^2 \right) \leq 0, \forall k \quad (4-26)$$

$$\mathbf{a}^H(\theta)(\mathbf{P}_0)\mathbf{a}(\theta) + \mathbf{a}^H(\theta) \left(\sum_{k=1}^K \alpha_k^2 \tilde{\mathbf{p}}_k \tilde{\mathbf{p}}_k^H \right) \mathbf{a}(\theta) \geq G_{\min} \quad (4-27)$$

$$\text{Tr}(\mathbf{P}_0) + \sum_{k=1}^K \alpha_k^2 \leq P_{\max} \quad (4-28)$$

$$\text{Tr}(\mathbf{H}_{T,m} \mathbf{P}_0) + \sum_{k=1}^K \alpha_k^2 |\mathbf{h}_{T,m}^H \tilde{\mathbf{p}}_k|^2 \leq I_{\max}, \forall m \quad (4-29)$$

$$\mathbf{P}_0 \succeq 0 \quad (4-30)$$

$$c_k \geq 0, \forall k \quad (4-31)$$

Problem (4-25) is convex and the solution can be obtained by the CVX toolbox in MATLAB. If the predicted values of $\{\alpha_k\}$ result in an infeasible problem, a slack variable s can be introduced into constraint (4-28) and incorporated as a penalty term in the objective function to ensure feasibility. Subsequently, the values of $\{\alpha_k\}$ are updated based on constraint (4-28) using $\mathbf{P}_0^*, s^*$. Moreover, if the optimal $\mathbf{P}_0^*$ is rank-one, the common precoder vector can be obtained by $\mathbf{p}_0 = \sqrt{\lambda_{\max}(\mathbf{P}_0)} \mathbf{v}_{\max, \mathbf{P}_0}$, where $\lambda_{\max}(\mathbf{P}_0)$ denotes the largest eigenvalue of the matrix $\mathbf{P}_0$, and $\mathbf{v}_{\max, \mathbf{P}_0}$ represents the corresponding eigenvector associated with this maximum eigenvalue. If the solution $\mathbf{P}_0^*$ is not rank-one, penalty-based optimisation approach [102] can be employed to approximate rank-one matrix constraints, enabling the derivation of near-optimal solutions by progressively enforcing rank-one through penalisation.

4.5.6 Results

Numerical simulations are presented to validate the effectiveness of the proposed scheme. The terrestrial channels from the b -th BS to k -th SU and to m -th TUs are modelled based on [98] and are defined as $\mathbf{g}_{S(T),b,k(m)} = \sqrt{\alpha_{S(T),b,k(m)}} \tilde{\mathbf{g}}_{S(T),b,k(m)}$, where $\tilde{\mathbf{g}}_{S(T),b,k(m)}$ represents the small-scale fading with elements following $CN(0,1)$, and $\alpha_{S(T),b,k(m)}$ denotes the large-scale fading coefficient, which accounts for both path loss and shadow fading, as described in [98].

The channel vector from the satellite to k -th SU follows the model described in [103], [104], and is given by $\mathbf{h}_{S,k} = \frac{\sqrt{G_u}}{4\pi \frac{d_{S,k}}{\lambda} \sqrt{\kappa T B_W}} \mathbf{b}_{S,k}^{1/2} \odot \mathbf{r}_{S,k}^{1/2} \odot \exp(j\theta_{S,k})$, where G_u accounts for the antenna gain of the SU, $d_{S,k}$ denotes the distance from the satellite to the k -th SU, λ represents the wavelength, $\kappa = 1.38 \times 10^{-23}/K$ is the Boltzmann's constant, $T = 300$ K is the system noise temperature, and B_W is the bandwidth. The beam gain component is defined as $b_{S,k,n} =$

$G_{max} \left[\frac{J_1(\delta_{S,k,n})}{2\delta_{S,k,n}} + 36 \frac{J_3(\delta_{S,k,n})}{\delta_{S,k,n}^3} \right]^2$, where G_{max} denotes the maximum beam gain, and $J_1(\cdot)$ and $J_3(\cdot)$ are the first- and third-order Bessel functions of the first kind, respectively. $\delta_{S,k,n} = 2.07123(\sin(\vartheta_{S,k,n}) / \sin(\vartheta_{3dB}))$ and $\vartheta_{S,k,n}$ denotes the angle between the center of the n -th beam and the k -th SU, and ϑ_{3dB} represents the angle at which the antenna gain is reduced by 3 dB. $\mathbf{r}_{S,k}$ is an N -dimensional vector representing rain attenuation coefficients, and $\boldsymbol{\theta}_{S,k}$ is an N -dimensional channel phase vector with elements independently and uniformly distributed over the interval $[0, 2\pi]$. The channel vector from the satellite to the m -th TU follows the same model.

The LEO satellite is positioned at an altitude of 600 km, located at coordinates (0,0). SUs are randomly distributed within a circular area of 10 km radius centered at (0,0). The BSs and TUs are deployed within a 1 km radius area centered at (12 km,0). The BSs are assumed to be mounted at a height of 15 m, while the TUs are positioned at 1.65 m [98]. The SUs are equipped with receiver dishes mounted at a height of 10 m [105]. The satellite is equipped with a directional antenna providing a maximum gain of 35 dBi and one-half the 3 dB beamwidth of 1.6° [106]. The satellite receiver achieves a maximum gain of 39.74 dBi [107].

The system operates at a carrier frequency of 20 GHz with a total bandwidth of 500 MHz. $B=2$ BSs are considered, each transmitting at 25 dBm with $N_B = 4$ antennas, serving $M=2$ TUs. The satellite is equipped with $N=4$ antennas, serving $K=2$ SUs with $u_k = 1$, and operates with a power budget ranging from 30 dBm to 36 dBm. An interference-to-noise ratio threshold of -10 dB [105] is considered at the TUs. The direction of the target is randomly selected from $\theta^* \in [-45^\circ, 45^\circ]$, and the minimum beampattern gain G_{min} is set to 50% of the gain achieved by the radar-only scheme described in [101].

The proposed hybrid CNN-SDR scheme, referred to as "RSMA-CNN", is compared with the near-optimal SCA-based method denoted as "RSMA-SCA", which approximates the non-convex rate and radar constraints using convex lower bounds in an iterative manner. Additionally, the RSMA method without radar constraints, denoted as 'RSMA w/o radar', is considered, along with two baseline schemes based on ZF and MRT. These baseline methods optimize the power allocation for the users' precoding vectors and precoding vector of the dedicated radar signal.

The RSMA-SCA method was employed to generate the dataset used for training the deep CNN model, comprising 32,000 training samples and 8,000 validation samples. After fine-tuning, the selected model parameters include $F_1 = 8$, $F_2 = 4$, $F_3 = 64$, 1024 hidden neurons in the FC layers, a batch size of 128 for training, and an initial learning rate of lr with an exponential decay factor of 0.95 applied every 1,000 mini-batch updates. Figure 4-37 illustrates the convergence behaviour of the proposed deep CNN-based scheme under different learning rates. It can be observed that relatively high learning rate values lead to unstable MSE behaviour, characterized by significant fluctuations across epochs. In contrast, lower learning rates result higher final MSE values. Therefore, a learning rate of $lr = 0.005$ is selected for training the final model, as it provides a good balance between stability and performance.

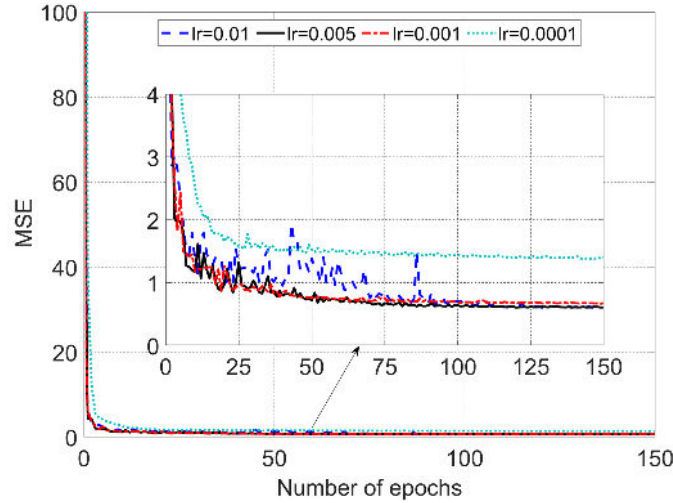


Figure 4-37: Convergence behaviour of the CNN-based model.

The sum-rate performance as a function of the power budget at the LEO satellite is presented in Figure 4-38. It is observed that the RSMA-based schemes outperform the baseline methods of ZF and MRT, owing to the ability of RSMA to partially decode both radar and inter-user interference through the common stream. A slight improvement in sum-rate is achieved by RSMA without radar compared to the proposed RSMA with radar, due to the additional radar constraints that limit communication performance. Furthermore, performance comparable to the RSMA-SCA method is achieved by the proposed hybrid CNN-SDR scheme, while a significant reduction in computational complexity is also attained.

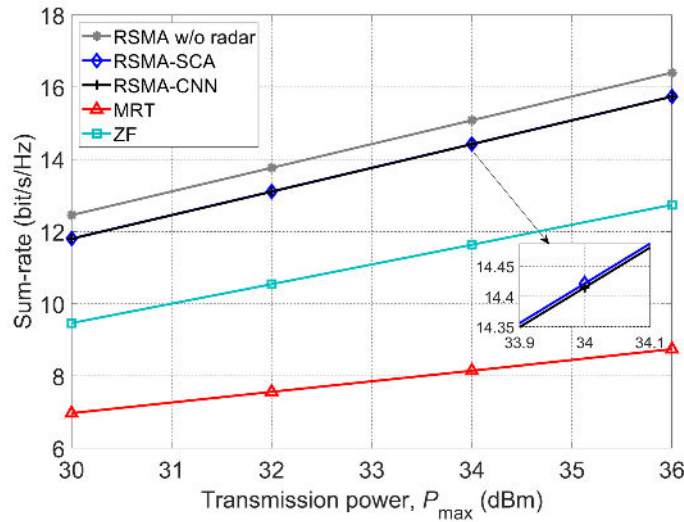


Figure 4-38: Sum-rate versus the transmission power.

In Figure 4-39, the cumulative distribution function (CDF) of the sum-rate over multiple channel realizations is illustrated, considering power budgets of $P_{max} = 32$ dBm and $P_{max} = 36$ dBm. It is shown that the proposed hybrid CNN-SDR scheme closely matches the performance of the RSMA-SCA method, thereby demonstrating its effectiveness in achieving near-optimal power allocation and precoding. Moreover, consistent outperformance over the baseline methods of ZF and MRT is achieved by the RSMA-based schemes.

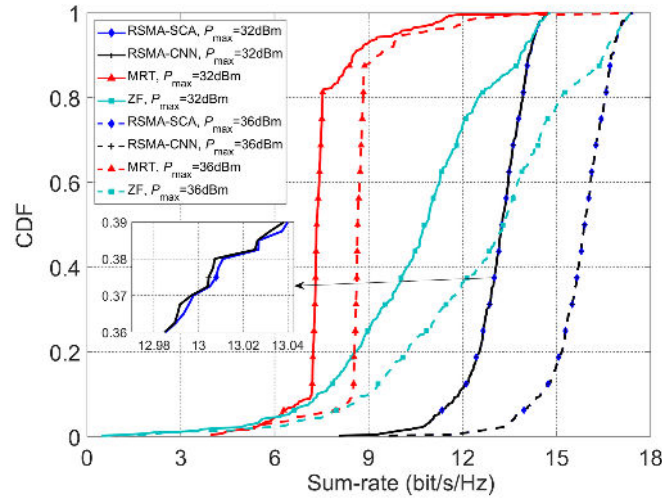


Figure 4-39: CDF of the sum-rate.

In Figure 4-40, the normalized transmit beampattern obtained for the target at $P_{max} = 30$ dBm is presented, where the angle $\theta^* = 0^\circ$ is selected to enhance visualization. The transmit beampattern is normalized with respect to the minimum required beampattern gain, G_{min} . It is observed that comparable performance in satisfying the radar requirements is yielded by the RSMA-based schemes. Additionally, similar beampattern characteristics to the RSMA-based schemes are exhibited by the baseline methods of ZF and MRT. This is attributed to the radar constraint (4-21), by which a minimum beampattern gain is enforced while the remaining resources are allocated for maximizing the users' sum-rate.

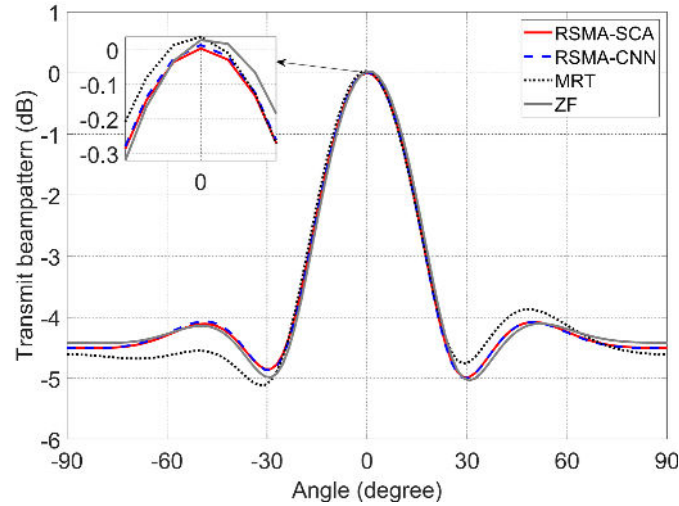


Figure 4-40: Normalized transmit beampattern for a target with $\theta^* = 0^\circ$.

In Figure 4-41, the CDF of the computational time required by each comparative scheme is shown. All simulations were conducted on a workstation equipped with an Intel Core i9-14900K processor and 64 GB of RAM. The computational time for the proposed method is measured by including both the CNN-based model and the SDR-based technique, with an average inference time of 4.3 ms recorded for the CNN when executed solely on the CPU to ensure a fair comparison with the baseline methods. The iterative baseline schemes are considered converged when the relative change in the objective function between successive iterations falls below 10^{-4} . A significantly lower computational cost is achieved by the proposed hybrid

CNN-SDR scheme, requiring on average only 2.7% of the time compared to the RSMA-SCA method. Additionally, superior performance is attained over the ZF and MRT baselines, which involve radar precoding vector optimization. These results highlight the computational efficiency and overall performance achieved by the proposed approach.

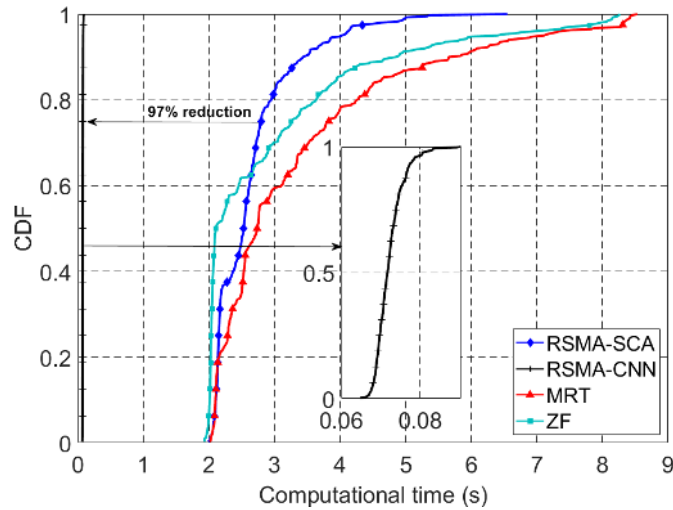


Figure 4-41: CDF of the computational time.

5 CONCLUSION

This chapter acts as the conclusion for ETHER D4.2, which is the final deliverable of WP4, providing a summary of the results related to the zero-touch data-driven network and service orchestration within the 3D ETHER architecture. WP4 is founded on three key developments: 1) The zero-touch management, orchestration, and monitoring of the virtualised ETHER network and cloud continuum resources; 2) Unified RAN and transport management leveraging distributed network control across the ETHER layers; and 3) AI-enabled E2E network optimisation. Each of these developments corresponds to a specific chapter in this document.

The first development is concerned with Zero-touch management, orchestration, and monitoring of the virtualised ETHER network and cloud continuum resources. It aims to create a framework that enables seamless service provisioning, enhances network resilience, and facilitates real-time management across a geographically varied and heterogeneous network infrastructure in an integrated terrestrial and non-terrestrial network scenario. This has been achieved through the execution of two main activities (as described in chapter 2): AI-driven zero-touch closed-loop orchestration (A1-1) and Mobility and geo-localisation management (A1-2).

- AI-driven zero-touch closed-loop orchestration (A1-1): This activity focussed on developing a fully automated, AI-driven closed-loop orchestration framework that facilitates continuous monitoring, analysis, and adaptation of both service and infrastructure components across the integrated TN and NTN. By replacing error-prone, human-centric approaches, this self-managing system leverages historical and real-time data to make proactive, intelligent orchestration decisions suited for the complex integrated network architecture targeted by ETHER. This deliverable has enhanced the framework by extending the Northbound Interface with infrastructure-related endpoints for real-time visibility into node availability and state and by introducing a modular hybrid observability stack that ensures scalable, efficient metric collection across heterogeneous environments. These developments directly support the needs of UC3, where safety-critical operations demand rapid, context-aware orchestration.
- Mobility and geo-localisation management (A1-2): This activity addressed the challenges of mobility and service placement by developing a spatial-aware orchestration capability. The solution is based on the integration of a GIS within the MANO framework, enhanced by a plugin that utilizes OSM data. This plugin enriches the orchestrator with geographical intelligence, allowing it to map network resources and services to precise physical locations and trajectories. Consequently, the system can deploy services to a specified target area. An operator can define a geographical zone of interest, and the orchestrator automatically identifies and allocates the optimal network assets, such as virtual functions on LEO satellites, to serve that region effectively. This functionality enables geo-aware service provisioning, which is critical for ensuring service continuity and efficient resource management in dynamic integrated network environments

The second development is focused on the unified RAN and transport management through distributed network control across several layers, which includes six main activities (as described in chapter 3): SDN-based WIM for integrated terrestrial TN-NTN (A2-1), QoS-aware multi-domain 3D routing (SDN-App) (A2-2), Reinforcement learning-based data compression and routing design for multi-satellite systems (A2-3), Model-based 3D network management and SDN integration (A2-4), Distributed SDN controller placement (A2-5), and SDN controller for the 3D Ether architecture (A2-6).

- **SDN-based WIM for integrated terrestrial TN-NTN (A2-1):** This activity was focused on the development and validation of an SDN-based WIM designed to address the dynamic challenges of integrated TN-NTN environments. The developed WIM leverages an SDN controller, which ingests time-varying topology data from a satellite network simulator (Hypatia), to automate the provisioning of E2E transport paths across a virtualised testbed. The performance evaluation confirmed the WIM's effectiveness, demonstrating stable throughput, low average RTT, and minimal jitter under conditions of frequent topological changes that emulate LEO satellite handovers. The observed transient disruptions during path recalculation were minimal, validating the WIM's capability for rapid adaptation. The results confirm that the developed WIM is a robust solution for ensuring seamless service continuity and automating network operations in dynamic, integrated network architectures.
- **QoS-aware multi-domain 3D routing (SDN-App) (A2-2):** This activity was devoted to the development of QoS-aware routing algorithm (3DQR) for 3D multi-controller SDN environment (compliant with ETHER general architecture). The proposed approach addresses key issues of SDN and optimised routing in integrated 3D environments: CP scalability, interoperability across controllers, QoS-driven communication and optimised network resources utilization. While provisioning QoS paths, 3DQR effectively reduces flow rejection and rerouting rates as well as improves load distribution on both domain and E2E levels. The introduced GNN-based agent architecture enables transferring the agents into different network setups without major performance losses, which makes them suitable for optimising time-varying network environments such as NTN. The evaluation results of 3DQR confirm substantial performance gains compared to the baseline routing methods, effective prioritisation of traffic, and stable performance improvements regardless of network topology. Altogether, the proposed routing approach can be effectively used to provisioning QoS-driven services including critical ones such as aviation services.
- **Reinforcement learning-based data compression and routing design for multi-satellite systems (A2-3):** This activity has proposed: the JCR strategy tailored for multi-orbit LEONets. Aiming to optimize large-scale LEOSat networks, a distributed JCR method based on MADDQN, namely MADDQN-JCR, has been developed to minimize the total delay, where each LEOSat acts as a learning agent. In this approach, the source LEOSat learns to optimize its CR and routing policy, while other LEOSat agents determine the most efficient routing through the learning process to forward data to the GW. Simulation results have revealed that the proposed MADDQN-JCR method outperforms the benchmark scheme WoC, significantly reducing E2E delay and data loss rate. This highlights the proposed method's effectiveness for latency-sensitive applications within LEOSat networks.
- **Model-based 3D network management and SDN integration (A2-4):** Cost-driven aircraft communications and message routing methods have been proposed and evaluated for integrated 3D networks management, considering the context of future 6G aeronautical networks. The proposed 3D network routing and handover methods consider the costs of message transmission and routing through the different network layer base stations, following a schedule established for handover decisions that minimise the overall costs. The proposed cost-driven handover method is built on top of a graph-based framework, where the service time of the base stations is predicted. Hence, the method can proactively schedule a handover when the base station becomes visible with a good link quality. The proposed method is evaluated and compared to the standard 5G measurement-based method and to our previous non-cost-driven method. The results demonstrate that the proposed cost-driven method can achieve cost-effective operations while meeting service performance requirements. The developed methods enable data-driven decisions for efficient message routing and handover management across the integrated 3D network.
- **Distributed SDN controller placement (A2-5):** This activity proposed a hierarchical controller placement scheme, CoMOSat, tailored for multi-orbit satellite networks. CoMOSat partitions the set of LEO and ground nodes into domains and employs a GA-based algorithm to elect

domain controllers for each domain and a master controller to manage the domain controllers. Additionally, CoMOSat incorporates an update strategy designed to minimize overhead during domain and control plane reconfigurations. Simulation results demonstrate that CoMOSat effectively reduces the average number of controllers by 3.54% and the flow setup delay by 41.79% compared to FP, showcasing its efficiency in optimizing resource utilization and enhancing system stability.

- SDN controller for the 3D Ether architecture (A2-6): This activity explored a technical specification for an SDN controller implementation through ONOS, harnessing all TN and NTN network components to provide a seamless connection between different users. Using ONOS built-in capabilities of reactive forwarding, host location and OpenFlow providers, it is possible to provide on-demand connectivity between different hosts without any necessary configuration. ONOS also allows the development of applications on top of its baseline deployment, which broadens the number of devices that can be connected to its network, enabling the connection between the different layers of the ETHER environment, the Ground (gNBs, TN backhaul, edge and IoT nodes), Aerial (UAVs and HAPS) and Satellite (LEO and MEO), enhancing service continuity and improving data link resiliency for the use case of air-space safety critical operations.

The third development is related to the AI-enabled E2E network optimisation, which includes four key activities (as described in chapter 4): AI-enabled 3D network performance and topology forecasting (A3-1), real-time energy-efficient resource allocation and routing (A3-2), AI-based offline user association for the 3D sustainable ETHER network (A3-3) and AI-Driven Beamforming for Integrated Sensing and Communication in Terrestrial-Satellite Networks (A3-4).

- AI-enabled 3D network performance and topology forecasting (A3-1): Satellite-based network KPIs are inherently susceptible to noise, significantly complicating accurate forecasting. To address this challenge, A3-1 activity introduces a specialised machine learning pipeline from ETHER, designed for scalable and accurate forecasting despite noisy KPI data. This pipeline integrates a robust data preprocessing module, an optimised deep learning model, and custom loss functions. Building on the work presented in D4.1 we developed a finely tuned model to predict key performance metrics (KPMs) of flight link quality, like Signal-to-Noise Ratio (SNR) and Delay, leveraging a dataset of 223 flight simulations provided by Collins Aerospace. This model supports both univariate and multivariate UCs, leveraging intra-KPI dynamics to enhance forecasting accuracy. Our findings clearly demonstrate the critical role of data preprocessing, particularly log normalisation, in effectively managing KPMs that exhibit wide dynamic ranges. We observed notable performance differences between TSMixer and NBEATS models, underscoring the importance of model selection tailored to the specific characteristics of the data and the prediction task at hand. However, a limitation shared by both models was their struggle to accurately capture transient spikes in delay. This suggests a need for further investigation, potentially through incorporating exogenous variables or exploring alternative model architectures. Finally, the observed variations in inference speed highlight the crucial need to consider computational resources when deploying these models in real-world applications.
- Real-time energy-efficient resource allocation and routing (A3-2): This activity is focused on the joint network, computational and storage resource allocation problem so as to achieve high energy efficiency. Both an online algorithm, which offers self-configuration at run time, as well as an optimal solution, to be used as a benchmark were developed. The simulation results demonstrated the clear performance superiority of the proposed heuristic algorithm compared to the SotA. Specifically, PETA achieved substantial improvements in all evaluated scenarios, demonstrating up to 5.9 times higher energy efficiency. These findings

highlight PETA's effectiveness in real-time resource allocation and its strong potential to support truly sustainable 6G 3D networks.

- **AI-based offline user association for the 3D sustainable ETHER network (A3-3):** this activity leveraged RL to associate the UEs in the 3D ETHER network in an energy efficient way. In particular, a low-complexity DNN trained with Proximal Policy Optimization was proposed, capable of associating UEs with the access network based on their service requirements, while minimizing network power consumption as well as avoiding violating the capacity constraints. A three-phase curriculum learning strategy was employed for the DNN to effectively learn the intricate relationships between PRB allocation and energy consumption. The proposed approach was shown to reduce power consumption by up to 47.7%, while meeting user service requirements.
- **AI-Driven Beamforming for Integrated Sensing and Communication in Terrestrial-Satellite Networks (A3-4):** This activity focused on the development of a RSMA-based cognitive satellite–terrestrial network architecture, enabling a multi-antenna LEO satellite to perform joint communication and sensing while sharing spectrum with a terrestrial distributed MIMO system. The weighted sum-rate maximization problem was addressed under constraints related to power, sensing, and interference. A hybrid CNN-SDR solution was proposed, combining deep learning for power allocation with SDR-based common precoding and rate optimization. Evaluation results confirmed that the proposed scheme achieved performance close to a near-optimal SCA-based benchmark, while satisfying all system constraints and significantly reducing computational complexity. Moreover, the RSMA-based method consistently outperformed conventional baseline schemes, demonstrating its potential as a practical and efficient solution for future integrated satellite–terrestrial networks.

6 REFERENCES

- [1] ETHER, "Deliverable D4.1: Initial report on zero-touch data-driven network and service orchestration in the 3D ETHER architecture," 2024.
- [2] Nearby Computing, "NearbyOne Edge Orchestrator - Product Description," 2021. [Online]. Available: <https://www.nearbycomputing.com/wp-content/uploads/2021/08/NearbyOne-Product-Data-Sheet-v2.0.pdf>. [Accessed 23 July 2025].
- [3] MLflow Project, "MLflow – Mastering the ML lifecycle," 2025. [Online]. Available: <https://mlflow.org/classical-ml>. [Accessed 23 July 2025].
- [4] ETHER, "Deliverable D2.1: Initial report on ETHER network architecture, interfaces and architecture evaluation," 2023.
- [5] ETHER, "Deliverable D2.2: Use cases and KPIs," 2023.
- [6] ETHER, "Deliverable D2.4: Final report on ETHER network architecture, interfaces and architecture evaluation".
- [7] ETHER, "Deliverable D3.1: Initial report on key technological enablers for the seamless and energy efficient ETHER network operation," 2024.
- [8] ETHER, "Deliverable D5.1: Report on the testing methodologies and testbed setup," 2025.
- [9] ETSI, "European Telecommunications Standards Institute (ETSI) GS NFV 006 V4.4.1: Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Architectural Framework Specification," 2022.
- [10] J. A. Ruiz-de-Azua, U. Lopez, J. Avila and D. Rincon, "Virtual Satellite Network Simulator (VSNeS): a simulation engine to virtualize non-terrestrial networks," in *74th International Astronautical Congress: IAC 2023*, Baku, Azerbaijan, 2023.
- [11] M. Giordani and M. Zorzi, "Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities," *IEEE Network*, vol. 35, no. 2, pp. 244-251, Mar. 2021.
- [12] A. Guidotti, A. Vanelli-Coralli, M. Conti, S. Andrenacci, S. Chatzinotas and N. Maturo, "Architectures and Key Technical Challenges for 5G Systems Incorporating Satellites," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2624-2639, March 2019.
- [13] D. Wang, M. Giordani, M.-S. Alouini and M. Zorzi, "The Potential of Multilayered Hierarchical Nonterrestrial Networks for 6G: A Comparative Analysis Among Networking Architectures," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 99-107, September 2021.
- [14] "Ryu SDN Framework," [Online]. Available: <https://ryu-sdn.org/>. [Accessed 25 April 2025].
- [15] D. B. A. B. Á. J. E. S. a. A. S. S. Kassing, "Exploring the 'Internet from space' with Hypatia," in *in Proc. ACM Internet Meas. Conf. (IMC '20), Virtual Event, Pittsburgh, Pennsylvania, USA, 2020*.
- [16] J. Hu, L. Cai, C. Zhao and J. Pan, "Directed Percolation Routing for Ultra-Reliable and Low-Latency Services in Low Earth Orbit (LEO) Satellite Networks," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, Victoria, BC, Canada, 2020.
- [17] P. A. Iannucci and T. E. Humphreys, "Fused Low-Earth-Orbit GNSS," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 4, pp. 3730-3749, August 2024.
- [18] "Hping3," Kali, 23 May 2024. [Online]. Available: <https://www.kali.org/tools/hping3/>. [Accessed 26 May 2025].
- [19] M. Mortimer, "iperf3 Documentation," 27 May 2018. [Online]. Available: <https://iperf.fr/>. [Accessed 26 May 2025].
- [20] A. Guidotti, A. Vanelli-Coralli, V. Schena, N. Chuberre, M. El Jaafari, J. Puttonen and S. Cioni, "The Path to 5G-Advanced and 6G Non-Terrestrial Network Systems.," in *11th Advanced Satellite Multimedia Systems Conference and the Multimedia Systems Conference and the 17th Signal Processing for Space Communications Workshop ASMS/SPSC*, Graz, Austria, 2022.
- [21] D. King and K. Shortt, "Time Variant Challenges for Non-Terrestrial Networks," Internet Engineering Task Force, Wilmington, USA, 2023.
- [22] G. Geraci, D. Lopez-Perez, M. Benzaghta and C. S., "Integrating Terrestrial and Non-terrestrial Networks: 3D Opportunities and Challenges.," *IEEE Communications Magazine*, vol. 61, p. 42–48, 2022.
- [23] L. Han, A. Retana, C. Westphal and R. Li, "Large Scale LEO Satellite Networks for the Future Internet: Challenges and Solutions to Addressing and Routing," *Computer Networks and Communications*, vol. 1(1), p. 30–57, 30 12 2022.
- [24] P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla and R. Gupta, "Multipath TCP in SDN-enabled LEO Satellite Networks," in *IEEE Military Communications Conference*, Baltimore, MD, USA, 2016.
- [25] V. Monzon Baeza, G. Rigazzi, S. Aguilar, R. Ferrus, J. Ferrer, S. Mhatre and M. Guadalupi, "IoT-NTN Communications via Store-and-Forward Core Network in Multi-LEO-satellite Deployments," in *IEEE 35th*

- International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Valencia, Spain, 2024.
- [26] 3rd Generation Partnership Project, "Study on Satellite Access-Phase 3, ver. 19.2.0. Technical Report TR 22.865," 2023.
 - [27] Orange, "Mobile Network Technology Evolutions Beyond 2030," Orange, Paris, France, 2024.
 - [28] Open Networking Foundation, "OpenFlow Switch Specification, Version 1.5.1 (Protocol Version 0x06); Specification ONF TS-025," Palo Alto, CA, USA, 2015.
 - [29] H. van Hasselt, A. Guez and D. Silver, "Deep Reinforcement Learning with Double Q-learning," in *30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, 2015.
 - [30] 3rd Generation Partnership Project, "System Architecture for the 5G System (5GS), ver. 19.2.1. Technical Standard TS 23.501," 2025.
 - [31] K. Phemius and M. Bouet, "Monitoring Latency with OpenFlow.," in *9th International Conference on Network and Service Management (CNSM 2013)*, Zurich, Switzerland, 2013.
 - [32] R. Kołakowski, S. Kukliński and L. Tomaszewski, "Hierarchical Deep Reinforcement Learning-Based Load Balancing Algorithm for Multi-Domain Software-Defined Networks," in *IFIP Networking Conference (IFIP Networking)*, Thessaloniki, Greece, 2024.
 - [33] H. van Hasselt, A. Guez and D. Silver, "Deep Reinforcement Learning with Double Q-learning," in *30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, 2015.
 - [34] J. Gilmer, S. Schoenholz, P. Riley, O. Vinyals and G. Dahl, "Neural Message Passing for Quantum Chemistry," in *34th International Conference on Machine Learning (ICML'17)*, Sydney, NSW, Australia, 2017.
 - [35] J. Lee, I. Lee and Kang, "Self-Attention Graph Pooling," 2019.
 - [36] Celestrak, "Current Supplemental GP Element Sets," [Online]. Available: <https://celestrak.org/NORAD/elements/supplemental/>.
 - [37] E. Lagunas, S. Chatzinotas and B. Ottersten, "Low-earth orbit satellite constellations for global communication network connectivity," *Nature Reviews Electrical Engineering*, 2024.
 - [38] R. Wang, M. A. Kishk and M.-S. Alouni, "Ultra reliable low latency routing in LEO satellite constellations: A stochastic geometry approach," *IEEE J. Sel. Areas Commun.*, 2024.
 - [39] X. Feng, Y. Sun and M. Peng, "Distributed Satellite-Terrestrial Cooperative Routing Strategy Based on Minimum Hop-Count Analysis in Mega LEO Satellite Constellation," *IEEE Trans. Mobile Computing*, 2024.
 - [40] I. Leyva-Mayorga, M. Martinez-Gost, M. Moretti, A. Pérez-Neira, M. Ángel Vázquez and P. Popovski, "Satellite Edge Computing for Real-Time and Very-High Resolution Earth Observation," *IEEE Trans. Commun.*, vol. 71, no. 10, pp. 6180-6194, 2023.
 - [41] T. T. Nguyen, N. V. Ha, B. L. Le and R. Schober, "Joint Data Compression and Computation Offloading in Hierarchical Fog-Cloud Systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 293-309, 2020.
 - [42] J. Song, Y. Ju, L. Liu, Q. Pei, C. Wu, M. A. Jan and S. Mumtaz, "Trustworthy and Load-Balancing Routing Scheme for Satellite Services with Multi-Agent DRL," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Hoboken, NJ, 2023.
 - [43] X. Shi, P. Ren and Q. & Du, "Heterogeneous Satellite Network Routing Algorithm Based on Reinforcement Learning and Mobile Agent," in *IEEE Globecom Workshops (GC Wkshps)*, 2020.
 - [44] B. Soret, I. Leyva-Mayorga, F. Lozano-Cuadra and M. D. Thorsager, "Q-learning for distributed routing in LEO satellite constellations," in *IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2024.
 - [45] F. Lozano-Cuadra, B. Soret, I. Leyva-Mayorga and P. & Popovski, "Continual Deep Reinforcement Learning for Decentralized Satellite Routing," *IEEE Trans. on Commun.*, 2025.
 - [46] P. Zuo, C. Wang, Z. Yao, S. Hou and H. Jiang, "An Intelligent Routing Algorithm for LEO Satellites Based on Deep Reinforcement Learning," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, 2021.
 - [47] Y. Lyu, H. Hu, R. Fan, Z. Liu, J. An and S. Mao, "Dynamic Routing for Integrated Satellite-Terrestrial Networks: A Constrained Multi-Agent Reinforcement Learning Approach," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1204-1218, 2024.
 - [48] T. Delamotte and A. Knopp, "Smart Diversity Through MIMO Satellite Q/V-Band Feeder Links," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 285-300, 2020.
 - [49] 3GPP, "NR and NG-RAN Overall Description, Technical Specification TS 38.300, V17.0.0," 3GPP, 2022.
 - [50] 3GPP, "Solutions for NR to Support Non-Terrestrial Networks (NTN), 3rd Generation Partnership Project (3GPP), Technical Report TR 38.821, V16.2.0," 3GPP, 2023.
 - [51] F. Fadhil, L. Chih-Kuang, A. Hamada and H. Phil, "Graph-Based Handover for Future Aeronautical 6G Integrated Terrestrial and Non-Terrestrial Networks," in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, San Diego, CA, USA, 2024.

- [52] W. Zhaofeng, J. Fenglin, L. Jianxin, F. Yinjin, S. Jinsong and H. Guyu, "A Graph-Based Satellite Handover Framework for LEO Satellite Communication Networks," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1547-1550, 2016.
- [53] FlightAware, "FlightAware, Live Flight Tracker," [Online]. Available: <https://www.flightaware.com/live>. [Accessed 21 05 2025].
- [54] J. Guo, L. Yang, D. Rincón, S. Sallent, Q. Chen and X. Liu, "Static Placement and Dynamic Assignment of SDN Controllers in LEO Satellite Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4975-4988, 2022.
- [55] X. Cao, Y. Li, X. Xiong and J. Wang, "Dynamic Routings in Satellite Networks: An Overview," *Sensors*, vol. 22, no. 12, p. 4552, 2022.
- [56] J. Zhou, Q. Yang, L. Zhao, H. Dai and F. Xiao, "Mobility-Aware Computation Offloading in Satellite Edge Computing Networks," *Sensors*, vol. 23, no. 10, pp. 9135-9149, 2024.
- [57] Z. Liu, Y. Jiang and J. Rong, "Resource Allocation Strategy for Satellite Edge Computing Based on Task Dependency," *Applied Sciences*, vol. 13, no. 18, p. 10027, 2023.
- [58] ITU, "Recommendation ITU-R M.2160-0: Framework and overall objectives of the future development of IMT for 2030 and beyond," Geneva, 2023.
- [59] GSMA, "Employing Non-Terrestrial Networks for IoT Connectivity," GSMA IoT Community, 2024.
- [60] J. W. Rabjerg, I. Leyva-Mayorg, B. Soret and P. Popovski, "Exploiting topology awareness for routing in LEO satellite constellations," in *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, 2021.
- [61] E. Baena, P. Testolina, M. Polese, D. Koutsonikolas, J. Jornet and T. Mel, "Space-O-RAN: Enabling Intelligent, Open, and Interoperable Non Terrestrial Networks in 6G," *arXiv preprint arXiv:2502.15936*, 2025.
- [62] ONF, "ONOS- Open Network Operating System," [Online]. Available: <https://opennetworking.org/onos/>. [Accessed 14 07 2025].
- [63] Wireshark, "Wireshark, The world's leading network protocol analyzer," Wireshark, [Online]. Available: <https://www.wireshark.org/>. [Accessed 14 07 2025].
- [64] T. O. Olwal, K. Djouani and A. M. Kurien, "A Survey of Resource Management Toward 5G Radio Access Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1656-1686, 3rd quarter 2016.
- [65] F. Rinaldi, H.-L. Maattanen, J. Torsner, S. Pizzi, S. Andreev, A. Iera and G. Araniti, "Non-Terrestrial Networks in 5G & Beyond: A Survey," *IEEE Access*, vol. 8, pp. 165178-165200, 2020.
- [66] J. Su, H. Cai, Z. Sheng, A. X. Liu and A. Baz, "Traffic prediction for 5G: A deep learning approach based on lightweight hybrid attention networks," *Digital Signal Processing*, vol. 146, no. 104359, March 2024.
- [67] C. Zhang, H. Zhang, J. Qiao, D. Yuan and M. Zhang, "Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1389-1401, June 2019.
- [68] Z. Luo and X. Dai, "Reinforcement learning-based computation offloading in edge computing: Principles, methods, challenges," *Alexandria Engineering Journal*, vol. 108, pp. 89-107, December 2024.
- [69] X. Ding, L. Feng, Y. Zou and G. Zhang, "Deep Learning Aided Spectrum Prediction for Satellite Communication Systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16314-16319, December 2020.
- [70] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *NIPS'14: Proceedings of the 28th International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014.
- [71] Y. Li, H. Zhang, L. Li, L. Shi, Y. Huang and S. Fu, "Multistep ahead atmospheric optical turbulence forecasting for free-space optical communication using empirical mode decomposition and LSTM-based sequence-to-sequence learning," *Frontiers in Physics*, vol. 11, January 2023.
- [72] H. Wu, J. Xu, J. Wang and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *NIPS'21: Proceedings of the 35th International Conference on Neural Information Processing Systems*, virtual conference, 2021.
- [73] B. N. Oreshkin, D. Carpo, N. Chapados and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," *arXiv*, February 2020. [Online]. Available: <https://arxiv.org/abs/1905.10437>. [Accessed November 2024].
- [74] Z. Jia, M. Sheng, J. Li, D. Zhou and Z. Han, "Joint Data Collection and Transmission in 6G Aerial Access Networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, 2021.
- [75] A. Mesodiakaki, M. Gatzianas, C. Bratsoudis, G. Kalfas, C. Vagionas, R. Maximidis and A. Miliou, "Online Energy-Efficient Resource Allocation in Integrated Terrestrial and Satellite 6G Networks," in *ICC 2024 - IEEE International Conference on Communications*, Denver, 2024.
- [76] Y. Yin, C. Huang, N. N. Xiong, D.-F. Wu and S. Huang, "Joint dynamic routing and resource allocation in satellite-terrestrial integrated networks," *Computer Networks*, vol. 231, 2023.

- [77] Y. Abe, F. Ortiz, E. Lagunas, V. M. Baeza, S. Chatzinotas and H. Tsuji, "Optimizing Satellite Network Infrastructure: A Joint Approach to Gateway Placement and Routing," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, Singapore, 2024.
- [78] A. Mesodiakaki, M. Gatzianas, G. Kalfas, C. Vagionas, R. Maximidis and N. Pleros, "ONE: Online Energy-efficient User Association, VNF Placement and Traffic Routing in 6G HetNets," in *2022 IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, 2022.
- [79] T. S. Kelso, "Celestrak," [Online]. Available: <https://celestrak.org/>. [Accessed May 2025].
- [80] J. Sun, F. Liu, H. Wang and D. O. Wu, "Joint VNF Placement, CPU Allocation, and Flow Routing for Traffic Changes," *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1208-1222, 2023.
- [81] A. Varasteh, B. Madiwalar, A. V. Bemten, W. Kellerer and C. Mas-Machuca, "Holu: Power-Aware and Delay-Constrained VNF Placement and Chaining," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1524-1539, 2021.
- [82] R. Stuhlfauth, "5G NTN takes flight: Technical Overview of 5G Non-Terrestrial Networks," Rohde & Schwarz, July 2022. [Online]. Available: https://www.rohde-schwarz.com/pl/solutions/satellite-testing/landingpages/white-paper-5g-ntn-takes-flight-technical-overview-of-5g-non-terrestrial-networks_255919.html. [Accessed May 2025].
- [83] MediaTek, "Satellite and Terrestrial Network Convergence," MediaTek 6G Technology White Paper, May 2023. [Online]. Available: https://728015.fs1.hubspotusercontent-na1.net/hubfs/728015/MediaTek_WhitePaper-6G-Satellite_EN0523F.pdf. [Accessed May 2025].
- [84] ITU-T, "Recommendation ITU-T L.1470: Greenhouse gas emissions trajectories for the information and communication technology sector compatible with the UNFCCC Paris Agreement," 2020.
- [85] H.-J. Moon and C.-B. Chae, "Cooperative ground-satellite scheduling and power allocation for urban air mobility networks," *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 1, pp. 218-233, 2024.
- [86] H. Dahrouj, S. Liu and M.-S. Alouini, "Machine Learning-Based User Scheduling in Integrated Satellite-HAPS-Ground Networks," *IEEE Network*, vol. 37, no. 2, pp. 102-109, 2023.
- [87] A. A. Shamsabadi, A. Yadav and H. Yanikomeroglu, "Enhancing Next-Generation Urban Connectivity: Is the Integrated HAPS-Terrestrial Network a Solution?," *IEEE Communications Letters*, vol. 28, no. 5, pp. 1112-1116, 2024.
- [88] R. Alghamdi, H. Dahrouj, T. Y. Al-Naffouri and M.-S. Alouini, "Equitable 6G Access Service Via Cloud-Enabled HAPS for Optimizing Hybrid Air-Ground Networks," *IEEE Transactions on Communications*, vol. 72, no. 5, pp. 2959-2973, 2024.
- [89] A. Alqasir and K. Aldubaikhy, "Optimizing Energy-Efficient Non-Terrestrial Networks with HAPs and UAVs for Enhanced Connectivity in Rural and Disaster-Affected Areas," in *2024 Conference on Renewable Energy Technologies and Modern Communications Systems: Future and Challenges*, 2024.
- [90] S. S. Hassan, D. H. Kim, Y. K. Tun, N. H. Tran, W. Saad and C. S. Hong, "Seamless and Energy-Efficient Maritime Coverage in Coordinated 6G Space-Air-Sea Non-Terrestrial Networks," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4749-4769, 2023.
- [91] N. Nway Ei, K. Kim, Y. Kyaw Tun, Z. Han and C. S. Hong, "Data Service Maximization in Space-Air-Ground Integrated 6G Networks," *IEEE Communications Letters*, vol. 28, no. 11, pp. 2598-2602, 2024.
- [92] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermarck, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume and A. Fehske, "How much energy is needed to run a wireless network?," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40-49, 2011.
- [93] E. Dahlman, S. Parkvall and J. Skold, 4G: LTE/LTE-advanced for mobile broadband, Academic press, 2013.
- [94] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.
- [95] J. Schulman, P. Moritz, S. Levine, M. Jordan and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," arXiv preprint arXiv:1506.02438, 2015.
- [96] A. Mesodiakaki, F. Adelantado, L. Alonso and C. Verikoukis, "Joint uplink and downlink cell selection in cognitive small cell heterogeneous networks," in *2014 IEEE Global Communications Conference*, 2014.
- [97] G. Geraci, D. López-Pérez, M. Benzaghta and S. Chatzinotas, "Integrating Terrestrial and Non-Terrestrial Networks: 3D Opportunities and Challenges," *IEEE Communications Magazine*, vol. 61, no. 4, p. 42-48, 2023.
- [98] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, p. 1834-1850, 2017.
- [99] F. Liu, C. Masouros, A. Li, H. Sun and L. Hanzo, "MU-MIMO Communications With MIMO Radar: From Co-Existence to Joint Transmission," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, p. 2755-2770, 2018.

- [100] C. Xu, B. Clerckx, S. Chen, Y. Mao and J. Zhang, "Rate-Splitting Multiple Access for Multi-Antenna Joint Radar and Communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 6, p. 1332–1347, 2021.
- [101] H. Hua, J. Xu and T. X. Han, "Optimal Transmit Beamforming for Integrated Sensing and Communication," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 8, p. 10588–10603, 2023.
- [102] A. H. Phan, H. D. Tuan, H. H. Kha and D. T. Ngo, "Nonsmooth Optimization for Efficient Beamforming in Cognitive Radio Multicast Transmission," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, p. 2941–2951, 2012.
- [103] J. Chu, X. Chen, C. Zhong and Z. Zhang, "Robust Design for NOMA-Based Multibeam LEO Satellite Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 3, p. 1959–1970, 2021.
- [104] S. Pala, K. Singh, C.-P. Li and O. A. Dobre, "Empowering ISAC Systems With Federated Learning: A Focus on Satellite and RIS-Enhanced Terrestrial Integrated Networks," *IEEE Transactions on Wireless Communications*, vol. 24, no. 1, p. 810–824, 2025.
- [105] E. Lagunas, C. G. Tsinos, S. K. Sharma and S. Chatzinotas, "5G Cellular and Fixed Satellite Service Spectrum Coexistence in C-Band," *IEEE Access*, vol. 8, p. 72078–72094, 2020.
- [106] ITU-R, "Satellite antenna radiation patterns for non-geostationary orbit satellite antennas operating in the fixed-satellite service below 30 GHz," ITU-R, Tech. Rep. S.1528, Geneva, Switzerland, 2001.
- [107] ITU-R, "Reference FSS Earth-station radiation patterns for use in interference assessment involving non-GSO satellites in frequency bands between 10.7 GHz and 30 GHz," ITU-R, Tech. Rep. S.1428-1, Geneva, Switzerland, 2001.