

# 5G ALLSTAR



Document Number: H2020-EUK-815323/5G-ALLSTAR/D4.3

Project Name:

5G AgiLe and fLexible integration of SaTellite And cellulaR (5G-ALLSTAR)

## Deliverable D4.3

Implementation of the multi-RAT load balancing algorithms and technical specifications of the relevant interfaces

## Deliverable D4.3

# Implementation of the multi-RAT load balancing algorithms and technical specifications of the relevant interfaces

<b>Project Number:</b>	H2020-EUK-815323
<b>Project Name:</b>	5G AgiLe and fLexible integration of SaTellite And cellu-laR

<b>Document Number:</b>	H2020-EUK-815323/5G-ALLSTAR/D4.3
<b>Document Title:</b>	Implementation of the multi-RAT load balancing algorithms and technical specifications of the relevant interfaces
<b>Editor(s):</b>	Alessandro Giuseppe (CRAT)
<b>Authors:</b>	Francesco Delli Priscoli (CRAT), Antonio Ornatelli (CRAT), Antonio Pietrabissa (CRAT), Alessandro Giuseppe (CRAT), Vito Andrea Racanelli (CRAT), Emanuele De Santis (CRAT); Seok Ho Won (ETRI), Taesang Choi (ETRI); You Jun Choi (KATECH), Jung Ho Kim (SNETICT), Quavlive (CRAT subcontractor) team Additional authors from University of Ulsan under sub-contract agreement with ETRI: Sungoh Kwon, Syed Maaz Shahid, Seyoum Yemane Teklay
<b>Dissemination Level:</b>	PU
<b>Security:</b>	Public
<b>Status:</b>	Final
<b>Version:</b>	1.0
<b>File Name:</b>	5G-ALLSTAR_D4.3_v5.docx3

## Abstract

This deliverable is the final output of WP4. The purpose of this document is to detail the multi-connectivity algorithm choice and implementation for the final Proof of Concept of the project. This document reports the testing environments developed by both the EU and the KR partners, presenting also the rationale behind certain design choices in their corresponding selected algorithms. Full detail on the functionalities implemented in the software components developed for the project are discussed, together with the main software interfaces and architectures.

This deliverable will lead the final integration activities in WP5 regarding the multi-connectivity capabilities that will be demonstrated at the end of the project.

## Keywords

*Multi-connectivity, traffic steering, network selection, network control algorithms*

## Acknowledgements

We would like to acknowledge the following people for the valuable reviews to the deliverable:

Seok Ho Won (ETRI)  
You-Jun (KATECH)  
Marjorie Thary (TAS)





## Executive Summary

This deliverable, issued at M34 reports the activities completed in the scope of the entire WP4, and in particular in tasks T4.2 and T4.3. This document is the final output produced by WP4, representing the final deliverable for the multi-connectivity aspects of the project.

The main objective of the deliverable is to detail the implementation and the selection process of the algorithms that will be demonstrated in the project Proof of Concept, detailing all the developed software and functionalities.

The deliverable presents the testing environments for both the EU and the KR sides of the consortium.

The 5G-ALLSTAR consortium published in, or submitted for, publication to international conferences and journals part of the results and discussions of this deliverable, e.g., [16]:

S. M. Shahid, Y. T. Seyoum, S. H. Won and S. Kwon, "Load Balancing for 5G Integrated Satellite-Terrestrial Networks," in IEEE Access, vol. 8, pp. 132144-132156, 2020, doi: 10.1109/ACCESS.2020.3010059.

## Contents

1	Introduction .....	13
1.1	Scope of the document .....	13
2	Structure of the document .....	14
3	Opensource Multi-RAT network EU simulator description .....	15
3.1	Environment.....	16
3.2	User Equipment .....	17
3.3	Base Stations.....	18
3.3.1	4G LTE Base Station .....	18
3.3.2	5G NR Base Station.....	20
3.3.3	Satellite Base Station .....	21
3.4	Pathloss Models.....	23
3.4.1	Cost Hata Pathloss model.....	23
3.4.2	Free Space Pathloss model .....	24
4	Description of KR simulator .....	25
4.1	Background.....	25
4.2	Network Architecture .....	25
4.3	Measurement Report Triggering.....	27
4.4	Flow Classification in 5G .....	29
4.5	Load Measurement in 5G Multi-RATs .....	29
4.6	Problem Formulation .....	31
4.7	Main Algorithm for Load Balancing.....	33
4.7.1	Information Gathering .....	34
4.7.2	Intra-RAT Load Balancing .....	34
4.7.3	Inter-RAT Load Balancing .....	36
5	Tial multi-connectivity scenarios for PoC .....	39
5.1	EU side scenarios .....	39
5.2	KR side scenarios .....	41
6	Selection of multi-connectivity algorithm for the demonstrator .....	42
6.1	EU side Algorithm Selection Rationale .....	42
6.1.1	Customisation of the selected algorithm for the scenarios of the PoC .....	44
6.1.2	Overview on the 5G-ALLSTAR suite of algorithms for multi-connectivity (from D4.2v2) .....	47
6.1.2.1	Traffic steering and network selection in 5G networks based on reinforcement learning .....	47
6.1.2.2	Wardrop Equilibrium based control (now selected for final PoC) .....	47
6.1.2.3	Friend or Foe Q-Learning.....	48
6.1.2.4	Maximal Load Balancing (developed by the KR partners) .....	48
6.1.2.5	Reinforcement learning resource optimization for mobile relays (developed by the KR partners).....	49
6.1.2.6	Analytic Hierarchy Process and Cooperative Differential Games for Multiconnectivity .....	49
6.1.2.7	ZeroSum games and Minimax for Multi-Connectivity.....	50
6.1.3	GAP analysis and comparison .....	51
6.2	KR side Simulator Performance Evaluation.....	52
6.2.1	Simulation Environment .....	52
6.2.2	Impact of the Proposed Algorithm on Load Distribution .....	53
6.2.3	Impact of the MLB Algorithm on Network Throughput and QoS .....	55
6.2.4	Impact of Various Numbers of Users.....	56
6.2.5	Impact of Different Channel Bandwidth .....	57
6.2.6	Impact of Delay-Tolerant Flows with Different Network Load .....	58
7	Description of developed SW components .....	60
7.1	SW Components of EU side.....	60
7.1.1	The Video Server architecture.....	60

7.1.2	The Adaptive Video Player algorithm .....	61
7.1.3	The encoding settings .....	65
7.1.4	Testing setup and tools .....	66
7.1.5	Custom Multipath TCP .....	69
7.1.5.1	The modified Round-Robin (mRR) scheduler .....	71
7.1.6	Video Server and Video Client Virtual Machines .....	73
7.1.7	OpenBACH architecture and integration .....	76
7.1.7.1	The OpenBach custom jobs .....	77
7.1.8	The Grafana “Live scenario” dashboard .....	81
7.1.8.1	mRR test scenarios .....	81
7.1.8.1.1	Test scenario 1 .....	82
7.1.8.1.2	Test scenario 2 .....	83
7.1.9	HTTP Proxy configuration .....	83
7.2	KR side Traffic Controller description .....	85
8	Conclusions .....	90
9	References.....	91



## List of Figures

Figure 1-1 Work package 4 activities.....	13
Figure 3-1 Multi-RAT network simulator functional architecture .....	16
Figure 3-2 Multi-RAT simulator implemented renderer .....	17
Figure 3-3 Frame structure of satellite telecommunication system . ....	22
Figure 4-1 Network architecture for KR simulator. ....	26
Figure 4-2 Radio protocol architecture for multi-connectivity [23]. ....	26
Figure 4-3 Events A3 and A4 for the algorithm: (a) A3 event parameter settings for load balancing, and (b) getting candidate edge UEs and target cell information using A4 event parameters. ....	29
Figure 4-4 5G QoS model .....	30
Figure 5-1 KR side multi-connectivity scenario .....	41
Figure 6-1 Latest EU PoC functional architecture from WP5 .....	42
Figure 6-2 Algorithm testing on real video stream .....	46
Figure 6-3 RRUR of terrestrial cells in the network (a) without the MLB algorithm (b) with the adaptive intra-RAT MLB algorithm, and (c) with the adaptive multi-RAT MLB algorithm. ....	54
Figure 6-4 RRUR of the satellite.....	54
Figure 6-5 8. Standard deviation of RRUR among the cells of the 5G RAT .....	55
Figure 6-6 (a) The number of satisfied UEs in the network (b) Average throughput of the network. ....	55
Figure 6-7 Performance of the load balancing algorithms with different numbers of UEs: (a) average throughput of the network, and (b) standard deviation of RRUR among the cells of 5G RAT.....	56
Figure 6-8 11. RRUR of the satellite cell.....	56
Figure 6-9 Standard deviation of RRURs among the terrestrial cells with varied 5G cell bandwidths.....	57
Figure 6-10 Average throughput of the network with varied terrestrial RAT bandwidths. ....	58
Figure 6-11 (a) Average throughput of the network, and (b) the number of satisfied UEs in the network with different delay-tolerant and delay-sensitive traffic ratios.....	58
Figure 7-1 The Video Server architecture.....	61
Figure 7-2 The ELASTIC controller block diagram.....	62
Figure 7-3 : Video Player screenshot.....	65
Figure 7-4 Video Player screenshot with quality levels detail.....	66
Figure 7-5 The testing architecture.....	67
Figure 7-6 Single path vs Multipath TCP. ....	69
Figure 7-7 Multipath TCP protocol stack.....	70
Figure 7-8 Virtual machines network configuration. ....	74
Figure 7-9 Test scenario 1 results .....	82
Figure 7-10 Test scenario 2 results .....	83

Figure 7-11 Virtual machines network configuration with HTTP proxy. ....	84
Figure 7-12 TC hardware .....	85
Figure 7-13 Test-bed for multi-connectivity functionality testing.....	86
Figure 7-14 Snapshot of link status monitored by TC. ....	87
Figure 7-15 Link failure test setup .....	87
Figure 7-16 Status of the links when a failover from terrestrial to satellite links occurred .....	88
Figure 7-17 Satellite traffic status when a failover occurred.....	88
Figure 7-18 Test result of the failover case i traffic congestion .....	88
Figure 7-19 Failover decision testing.....	89
Figure 7-20 Failback functionality testing.....	89

## List of Tables

Table 3-3-1: Available Resource Blocks (RB) for 4G LTE bandwidths and frequency bands (in bold the most used in EU) .....	18
Table 4-1: Simulation parameters.....	27
Table 4-2: Simulation parameters.....	33
Table 6-1: GAP analysis of the proposed algorithms for traffic flow control .....	51
Table 6-2: Simulation parameter values .....	52
Table 7-1 The encoding settings. ....	65
Table 7-2 The Video Player QoE metrics. ....	68
Table 7-3 The server side metrics. ....	68
Table 7-4 VM instances configuration.....	73
Table 7-5 Video Server Setup .....	75

## List of Abbreviations

<b>2D</b>	2 Dimensions / 2-Dimensional
<b>3D</b>	3 Dimensions / 3-Dimensional
<b>3GPP</b>	3 <sup>rd</sup> Generation Partnership Project
<b>5GPPP</b>	5G Public-Private Partnership
<b>5QI</b>	5G QoS Identifier
<b>ADP</b>	Approximated Dynamic Program- ming
<b>AHP</b>	Analytic hierarchy process
<b>AI</b>	Artificial Intelligence
<b>AIV</b>	Air Interface Variants
<b>AMF</b>	Access and Mobility Management Function
<b>AP</b>	Access Point
<b>API</b>	Application Programming Inter- face
<b>BS</b>	Base Station
<b>CN</b>	Core Network
<b>CP</b>	Control Plane
<b>CQI</b>	Channel Quality Indicator
<b>C-RAN</b>	Central Radio Access Network
<b>cRRM</b>	Central Radio Resource Manage- ment
<b>CSI</b>	Channel State Information
<b>CSI-RSRP</b>	CSI Reference Signal Received Power
<b>CSI-RSRQ</b>	CSI Reference Signal Received Quality
<b>CSI-SINR</b>	CSI Signal-to-Noise and In- terference Ratio
<b>CU</b>	Centralized Unit
<b>DASH</b>	Dynamic Adaptive Streaming over HTTP
<b>DL</b>	Downlink
<b>D2C</b>	Device to cellular
<b>D2D</b>	Device to Device
<b>DN</b>	Data Network
<b>D-RAN</b>	Distributed Radio Access Net- work

<b>DRB</b>	Data Radio Bearer
<b>dRRM</b>	Distributed Radio Resource Man- agement
<b>DP</b>	Dynamic Programming
<b>DU</b>	Distributed Unit
<b>FDD</b>	Frequency Division Duplexing
<b>FLC</b>	Fuzzy Logic Controller
<b>FS</b>	Fast Switch
<b>GEO</b>	Geostationary Earth Orbit
<b>gNB-CU</b>	next Generation Node B Central Unit
<b>gNB-DU</b>	next Generation Node B Distrib- uted Unit
<b>GRA</b>	Grey Relational Analysis
<b>HAS</b>	HTTP Adaptive Streaming
<b>HH</b>	Hard Handover
<b>HLS</b>	HTTP Live Streaming
<b>KPI</b>	Key Performance Indicator
<b>LEO</b>	Low Earth Orbit
<b>LTE</b>	Long Term Evolution
<b>MAC</b>	Medium Access Control
<b>MADM</b>	Multiple attribute decision making
<b>MCG</b>	Master Cell Group
<b>MLB</b>	Mobility Load Balancing
<b>MPTCP</b>	Multi Path Transmission Control Protocol
<b>MDP</b>	Markov Decision Process
<b>MN</b>	Master Node
<b>NE</b>	Nash Equilibrium
<b>NG-RAN</b>	New Generation RAN
<b>NR</b>	New Radio
<b>NTN</b>	Non-Terrestrial Network
<b>PDB</b>	Packet Delay Budget
<b>PDCP</b>	Packet Data Convergence Proto- col
<b>PDR</b>	Packet Detection Rule
<b>PDU</b>	Protocol Data Unit

<b>PHY</b>	Physical / physical layer
<b>PoC</b>	Proof of Concept
<b>PRB</b>	Physical Resource Block
<b>QFI</b>	QoS Flow Identifier
<b>QoE</b>	Quality of Experience
<b>QoS</b>	Quality of Service
<b>RAT</b>	Radio Access Technology
<b>RBUR</b>	Resource Block Utilization Ratio
<b>RL</b>	Reinforcement Learning
<b>RLC</b>	Radio Link Control
<b>RRC</b>	Radio Resource Control
<b>RRM</b>	Radio Resource Management
<b>RRUR</b>	Radio Resource Usage Ratio
<b>RSRP</b>	Reference Signal Received Power
<b>RTT</b>	Round Trip Time
<b>SCG</b>	Secondary Cell Group
<b>SDAP</b>	Service Data Adaptation Protocol

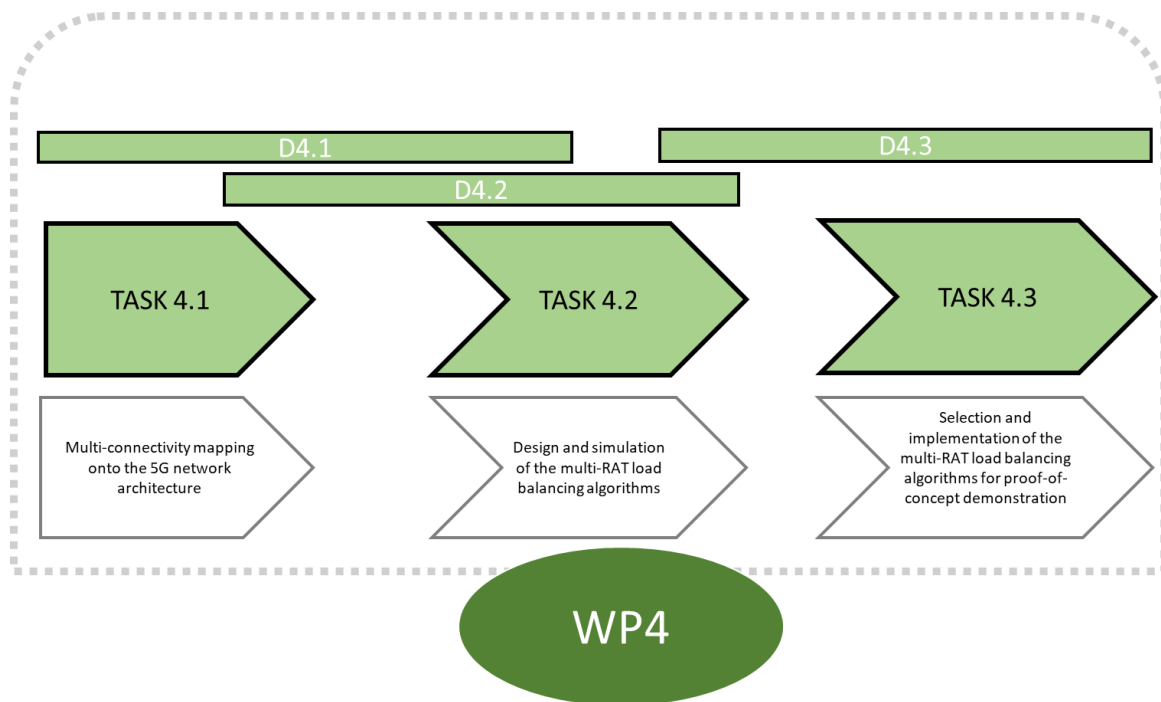
<b>SDF</b>	Service Data Flow
<b>SINR</b>	Signal over Interference plus Noise Ratio
<b>SMF</b>	Session Management Function
<b>SN</b>	Secondary Node
<b>SNR</b>	Signal to Noise Ratio
<b>TCP</b>	Transmission Control Protocol
<b>TDD</b>	Time Division Duplexing
<b>TDM</b>	Time Division Multiplexing
<b>TDMA</b>	Time Division Multiple Access
<b>TN</b>	Terrestrial Network
<b>UE</b>	User Equipment
<b>UL</b>	Uplink
<b>UP</b>	User Plane
<b>UPF</b>	User Plane Function
<b>VOD</b>	Video On Demand
<b>WLAN</b>	Wireless Local Area Network

# 1 Introduction

This deliverable provides an overview of the final activities conducted in the scope of WP4, and in particular in the two active tasks T4.1 and T4.2. The main goal of the deliverable is to detail the selection and the implementation of the multi-connectivity algorithms to be demonstrated in the PoC of the project. This document follows the second issue of D4.2, where in the light of the second project review, some advancements in the activities of WP4 were already introduced.

## 1.1 Scope of the document

This document is organized to report the completion and refinements of the final activities performed in T4.3 and “Selection and implementation of the multi-RAT load balancing algorithms for proof-of-concept demonstration”, as well as T4.2 “Design and simulation of the multi-RAT load balancing algorithms”. The logical relation among the deliverables of WP4 and their scopes is depicted in Figure 1-1, in compliance with the grant agreement.



**Figure 1-1 Work package 4 activities**

## 2 Structure of the document

The document is structured so that it first describes the numerical simulators and environments on which the algorithms developed in the scope of T4.2 have been validated and tested.

The PoC scenarios identified in WP2 are briefly discussed, and their requirements are mapped onto functionalities and characteristics used for the selection of the algorithms for the demonstration of the project.

The final part of the document presents an overview of the software components developed in the scope of WP4 for multi-connectivity, detailing the traffic flow controllers implementation and the custom developed adaptive 8k video streaming platform. Such components were used in WP4 to further test the developed algorithms in a more realistic communication setting, involving real data flows, and will be integrated in the scope of WP5 with all the other software components developed in the project.

In the light of the discussion above, the rest of the document is organised as follows:

- Chapter 3 details the implementation details of the opensource network simulator developed by the EU partners to test their algorithms (developed for multi-connectivity, network selection and traffic steering and detailed in D4.2v2) and to validate them with some preliminary numerical simulations (task in the scope of T4.2). Starting from this preliminary evaluation, the consortium was able to select and tailor an algorithm that meets the final PoC requirements and provides all the needed features.  
We mention that the developed simulator allows the demonstration and the study of multi-connectivity functionalities and algorithms beyond the scope of the project PoC, as it simulates a “*target scenario*” (as defined in the original proposal) in which all 5G technologies are deployed and available.  
This simulator has been opensourced and given a doi [1], becoming one of the output of the work package that is envisaged to be further developed in the upcoming months and years.
- Chapter 4 describes the simulator developed by the KR partners designed to allow the testing of load balancing algorithms in multi-RAT networks. The simulator developed allows for intra-RAT and inter-RAT traffic offloading and 5QI-aware traffic steering.
- Chapter 5 briefly reports the scenarios and use cases identified in the scope of WP2 to demonstrate in the PoC the multi-connectivity functionalities of WP4, for both the EU and the KR partners. This suite of “*trial scenarios*” (defined in the project proposal as the ones used for the PoC) is then used to identify the main functional requirements that the selected algorithms will need to satisfy in order to fully demonstrate in the final PoC the capabilities of the 5G-ALLSTAR multi-connectivity approach.
- Chapter 6 describes the process of selecting and customising the specific algorithms to be demonstrated in the project PoCs. Note that this chapter in some instances refers to the second issue of D4.2 when describing the algorithms customization aspects and their main characteristics.
- Chapter 7 presents an overview of all developed software components and functionalities related to WP4 activities, which will be used in the framework of the project PoC, starting from the EU 8k adaptive video streaming platform and following with the KR traffic controller. The software here described will then be integrated with the rest of the components developed in the project in the scope of the activities of WP5, for the overall project demonstration.

### 3 Opensource Multi-RAT network EU simulator description

This chapter describes the opensource network simulator [1] developed by the EU partners to test and validate their algorithms for multi-connectivity (traffic steering, network selection and resource management/load balancing) reported in D4.2. Thanks to such simulator the above-mentioned algorithms have been validated with some preliminary numerical simulations in the scope of T4.2 and the relevant results have been also reported in some scientific publications. The simulator in question has been essential in order to select and to customize the specific algorithms for both the target system (that envisages the complete deployment of all 5G functionalities) and the trial system (i.e. the system used for the PoC) considered in the project.

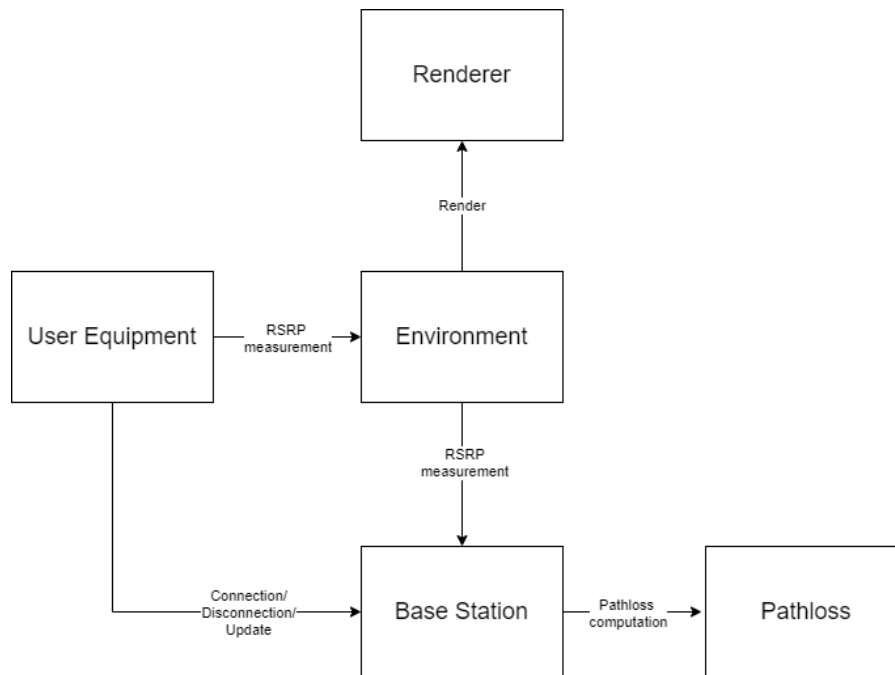
The four subsections of this chapter detail the three main software classes and the pathloss models developed for the functioning of the simulator. Regarding the software classes we start with the environment class, that models a RAN. We then proceed with the UE class and conclude with the base station class, that is used to represent any AP of various possible RATs. As mentioned, the final subsection of this chapter presents two pathloss models (for satellite and terrestrial communications)

The simulator considers three types of Radio Access Technologies (RATs): 4G LTE, 5G NR and Geostationary Satellite. The wireless network simulator developed aims at capturing the main dynamics of the radio access technologies that are taken into consideration, without addressing all the specifications required by the radio standards. In this way it is possible to maintain the complexity of the simulator low, and so being able to apply it to simulate (very) different scenarios, and at the same time to have significative results.

The simulator is written for Python 3, given its flexibility and portability. Moreover, the simulator has been designed to be as much modular as possible, so if a new module has to be added or an existing module has to be replaced, it is possible to change it without having to modify the remaining components. This facilitates a lot a developer in testing new scenarios, algorithms, and architectures.

The architecture of the simulator is presented in Figure 3-1, and it represents the fundamental interaction between the different functional blocks. From Figure 3-1 it is possible to see that no RAT is specified, and this is because all the technology-specific code has been packed inside the specific Base Station classes for each RAT. This aspect will be further analysed in the following sections.





**Figure 3-1 Multi-RAT network simulator functional architecture**

As mentioned, the simulator is completely opensource and it has been published on Zenodo [1] and Github [2].

The following sections deepen into the components of Figure 3-1 and the interactions between them.

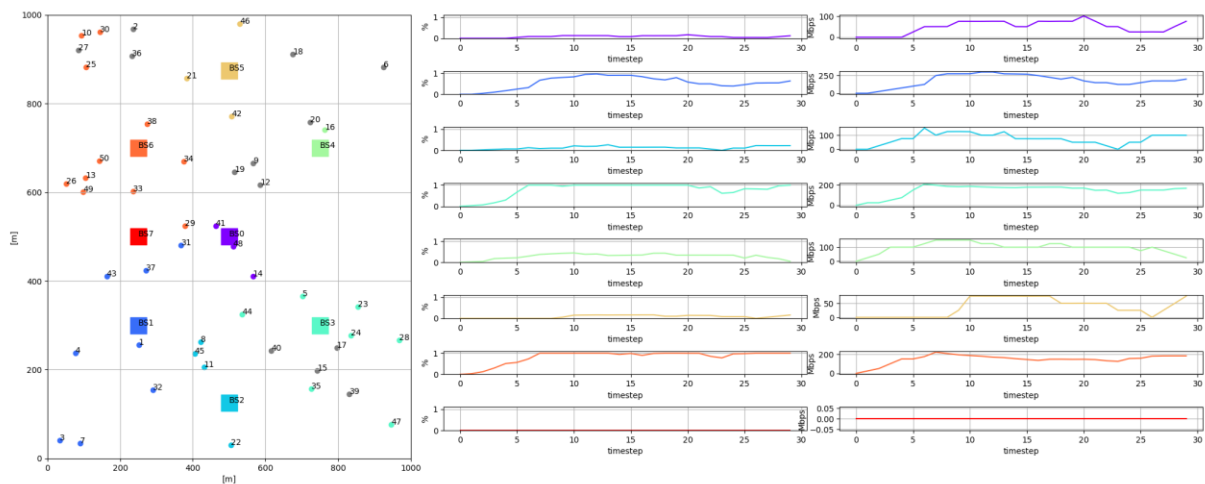
### 3.1 Environment

The environment class represents the wireless network environment. It contains a certain number of users and a certain number of generic base stations, that at this level are not further specified to which RAT they belong. This is because the simulator should be as much RAT-agnostic as possible, putting all the RAT details into the specific base station classes.

The environment simulates also a physical 2D/3D space, where the UEs and the base stations can move around.

The environment class has been designed to simulate the system step-by-step: each time the *step()* function is called, the system evolves as a discrete-time dynamical system and the current time step is increased. In this way the simulator can be configured to run in real-time (timing the call of the *step()* function) as well as in an offline manner.

It is possible to connect a *renderer* to an environment that can visualize the data in several ways. The render can be personalized to visualize the data in the most appropriate way for the scenario that has been simulated. A renderer has been developed to visualize some generic KPIs, like the load of the base stations and the allocated data rate, but it is possible to take other data from the simulator and process them for more tailored visualization systems. Figure 3-2 represents the output of the implemented render, where it is possible to see on the right the plots with the load of the APs and the allocated data rate and on the left a visual representation of the environment with all the APs and all the UE; the colour of the UE represents the AP where it is connected to (or it is grey if it is not connected to any AP).



**Figure 3-2 Multi-RAT simulator implemented renderer**

The *render()* function of the renderer can be called just after the *step()* function, to visualize the new situation of the environment or in different moments (e.g., this may be useful in case of complex visualization system, that may require some time to render the data on the screen, and so it may be better to update the visualization on a separate thread in an asynchronous way w.r.t. the environment *step()* function).

### 3.2 User Equipment

The User Equipment (UE) class models a terminal that can move in the environment (e.g., with a random pattern, following lines, etc.) and that requests some service (i.e., data rate) to the network in upload or download. In order to keep the simulator results as much interpretable as possible, it is currently not possible to simulate both uplink and downlink channels at the same time. This assumption does not affect the quality of the results obtained from the network simulator, since uplink and downlink channels are separated with TDD or FDD techniques, so they are somehow independent each other. Moreover, the low-level communication protocol is almost the same for uplink and downlink in 4G LTE, 5G NR and in the Satellite implementation taken as reference. Nevertheless, there are plans to include this feature in a new version of the simulator.

Being a simulator for multi-connectivity, the User Equipment class has been designed to allow a UE to be connected to multiple base stations at the same time, even belonging to different RATs. This opens the possibility to simulate scenarios where the data-rate request of the UE is split among multiple access points; in order to do so, the UE has been modelled to be unaware of the type of access point it is connecting to.

The basic action a UE can do is to move (in a random way or following a line, given by its direction) with a certain speed (through the function *move()*) and to connect/disconnect to access points (APs) according to some policy. In the provided implementation of the simulator the default policy is to connect to the single access point that has better signal power at the receiving end (RSRP), but this policy can be easily changed in order to test any user-centric, user-assisted or RAN-centric algorithm.

The UE requests behaviour has been implemented to follow a Poisson distribution over the connection/disconnection time and it can be tuned and modified (even in real-time) through the lambda coefficients of the Poisson distributions.

The UE connection process is as follows:

1. The UE waits  $t_c$  time steps for connection, where  $t_c \sim P_{\lambda_c}(n)$ ,  $n \in \mathbb{N}$
2. It asks the environment for the nearby<sup>1</sup> access points, together with their RSRP
3. It choses among the available APs the one(s) to connect to and the desired bitrate it would allocate on it (them) according to its connection policy
4. It waits  $t_d$  time steps for disconnection, where  $t_d \sim P_{\lambda_d}(n)$ ,  $n \in \mathbb{N}$ ; in the meanwhile, it updates its connection with the selected APs (since the UE may move, the resources the APs have allocated for its connection may change over time due to SNR/CQI variations)
5. It disconnects from the APs releasing all the resources allocated.

### 3.3 Base Stations

A Base Station class models the behaviour of an Access Point and it is the only technology-specific part of the simulator; indeed, it simulates the physical resource allocation for the specific RATs (4G LTE, 5G NR and Satellite). Each Base Station class shares a set of common APIs, in order to let the environment and the UEs be technology-unaware. These APIs are used to allocate and deallocate physical resources, update an existing connection (due to SNR/CQI change during time) and compute RSRP for a given UE. Moreover, they are designed to be as much generic as possible, in order to be compatible with many possible RATs.

As said before, in the networks simulator there are implemented three types of RATs: 4G LTE, 5G NR and a Satellite Inmarsat implementation taken from [3].

#### 3.3.1 4G LTE Base Station

4G LTE technology allocation units are called Physical Resource Blocks (PRBs) and they are composed by 12 frequency subcarriers (each one with 15kHz bandwidth) with a time slot of 1ms. A PRB is made of 12/14 OFDM symbols (respectively for Extended Cyclic Prefix and Normal Cyclic Prefix) and it is repeated every time frame (10ms). This implementation of 4G LTE Base Station takes as reference LTE FDD Frame Type 1, so uplink channel and downlink channel are separated in two different frequency ranges. The number of available PRBs depends on the bandwidth of the AP (that can range from 1.5MHz to 20MHz) based on the 3GPP specifications [4] as shown in table 3-1.

**Table 3-3-1: Available Resource Blocks (RB) for 4G LTE bandwidths and frequency bands (in bold the most used in EU)**

Bandwidth	Number of available RB	Frequency Bands
1.4 MHz	6	2, <b>3</b> , 4, 5, <b>8</b> , 12, 25, 26, 31, 66, 72, 73, 74, 87, 88
3 MHz	15	2, <b>3</b> , 4, 5, <b>8</b> , 12, 25, 26, <b>28</b> , 31, 66, 72, 73, 74, 87, 88
5 MHz	25	<b>1</b> , 2, <b>3</b> , 4, 5, <b>7</b> , <b>8</b> , 11, 12, 13, 14, 17, 18, 19, <b>20</b> , 21, 24, 25, 26, <b>28</b> , 30, 31, 65, 66, 68, 70, 71, 72, 73, 74, 85, 87, 88
10 MHz	50	<b>1</b> , 2, <b>3</b> , 4, 5, <b>7</b> , <b>8</b> , 11, 12, 13, 14, 17, 18, 19, <b>20</b> , 21, 24, 25, 26, <b>28</b> , 30, 31, 65, 66, 68, 70, 71, 74, 85

<sup>1</sup> Nearby means all the access points with a RSRP greater than a threshold value (e.g., -140dBm)

Bandwidth	Number of available RB	Frequency Bands
15 MHz	75	1, 2, 3, 4, 7, 18, 19, 20, 21, 25, 26, 28, 65, 66, 68, 70, 71, 74
20 MHz	100	1, 2, 3, 4, 7, 20, 25, 28, 71, 74

The 4G LTE Access Point  $j$  transmits its signals to a UE  $i$  with a certain subcarrier power  $P_{i,j}(t)$  that can be computed as follows:

$$P_{i,j}(t) = P_j \cdot G_j \cdot L_j \cdot L_{i,j}(t) \text{ [mW]}, \quad (3-1)$$

where  $P_j$  is the AP  $j$  subcarrier power (the power transmitted by a single PRB),  $G_j$  is the antenna gain,  $L_j$  are the AP feeder losses and  $L_{i,j}(t)$  is the pathloss term (that will be further specified in the following sections). This power is defined as the RSRP of AP  $j$ .

Upon a connection request from UE  $i$ , with a certain requested data rate  $\bar{R}_i$ , the 4G LTE base station has to compute how many resource blocks it should assign to the request. In order to do so, the SINR must be computed.

The SINR has the same physical meaning of the SNR, but it considers also the interference made by other communications on the same frequency band. This is very important on cellular communications since there are lots of base stations on the field and the number of frequency band is limited, so many of them work in the same frequency range, generating interference to one another. The interference term, in this simulator, considers also the interference produced by other RATs that use the same carrier frequency (e.g., many 5G NR frequency bands collapse over 4G LTE bands).

To compute the  $SINR_{i,j}$  between AP  $j$  and UE  $i \in I$ , the RSRP  $P_{i,k}$  of the neighbour APs  $k \neq j$  is first computed and then the interference term is computed.

The statistical interference model that has been considered, as shown in equation (3-3), directly depends on  $P_{i,k}$  (the higher interfering signal power is received by the UE, the higher is the interference), and on the Resource Block Utilization Ratio (RBUR)  $\Theta_k(t)$  at time  $t$ :

$$\Theta_k(t) = \frac{\sum_{\tau \in (t-T, t)} \sum_{h \in I \setminus \{i\}} C_{h,k}(\tau) N_{h,k}(\tau)}{T \cdot \#R_k}, \quad (3-2)$$

where  $C_{h,k}(t)$  is 1 if UE  $h$  is connected to AP  $k$  at time  $t$  and 0 otherwise,  $N_{h,k}(t)$  is the number of PRBs allocated to UE  $h$  at time  $t$ ,  $T$  is the length of the moving average window and  $\#R_k$  is the total number of resource blocks of AP  $k$ . Indeed, an interfering access point sharing the same frequency range, even if it has a strong signal power at UE side, does not interfere so much if it does not have many resources allocated.

$$I_{i,j}(t) = \sum_{k \neq j} \mathcal{F}_{j,k} \cdot P_{i,k}(t) \cdot \Theta_k(t), \quad (3-3)$$

where  $F_{j,k}$  is 1 if AP  $j$  and  $k$  share the same carrier frequency and 0 otherwise.

Once the interference term is computed for all the neighbour access points  $k \neq j$ , the noise term has to be computed:

$$\mathcal{N}_j = k_b T^{env} B_j, \quad (3-4)$$

where  $k_b$  is the Boltzmann constant,  $T^{env}$  is the environmental temperature (assumed 293.15 °K) and  $B_j$  is the bandwidth occupied by a PRB.

Once (3-1), (3-2), (3-3) and (3-4) are computed, it is possible to compute the  $SINR_{i,j}$  as:

$$SINR_{i,j} = \frac{P_{i,j}(t)}{I_{i,j}(t) + \mathcal{N}_j}, \quad (3-5)$$

$$r_{i,j} = \frac{B_j}{10} \log_2(1 + SINR_{i,j}). \quad (3-6)$$

With the SINR it is possible to estimate the data rate  $r_{i,j}(t)$  offered by a single PRB using (3-6), and so the number of PRBs needed to satisfy the bitrate request  $\bar{R}_i$  of UE  $i$ .

The assignment policy has been implemented as FIFO. This policy can of course be customized according to the simulation needs with much more complex allocation schemes. For the aim of this simulator the basic FIFO was enough, since it simulates a fair scenario where no UE can steal resources to a UE that is already allocated by the access point.

### 3.3.2 5G NR Base Station

5G NR technology is pretty similar to 4G LTE one: both of them uses OFDM/OFDMA symbols and many frequency bands are in common. On the contrary, 5G NR introduces a parameter called *numerology*  $\mu = 0,1,2,3,4$ . This parameter tunes the trade-off between the length of a timeslot (i.e., a group of 12/14 OFDM symbol) and the frequency bandwidth it occupies.

**Table 3-2: 5G NR subcarrier bandwidth and timeslot length depending on numerology**

Numerology ( $\mu$ )	Subcarrier Bandwidth (kHz)	Timeslot length (ms)
0	15	1
1	30	0.5
2	60	0.25
3	120	0.125

Numerology ( $\mu$ )	Subcarrier Bandwidth (kHz)	Timeslot length (ms)
4	240	0.0625

According to 3GPP standards [5], [6], the number of Physical Resource Blocks (PRBs) that are available to the 5G NR base station depends on the bandwidth allocated to the base station itself and on the numerology parameter, as shown in table 3-3 and 3-4, that are applied respectively for Frequency Range 1 (FR1, sub 6 GHz) and Frequency Range 2 (FR2, 24.25 GHz – 52.6 GHz)

**Table 3-3: Number of Resource Blocks for 5G NR FR1**

Subcarrier bandwidth (kHz)	5 MHz	10 MHz	15 MHz	20 MHz	25 MHz	30 MHz	40 MHz	50 MHz	60 MHz	80 MHz	90 MHz	100 MHz
	# RB											
15	25	52	79	106	133	160	216	270	N/A	N/A	N/A	N/A
30	11	24	38	51	65	78	106	133	162	201	245	273
60	N/A	11	18	24	31	35	51	65	79	107	121	135

**Table 3-4: Number of Resource Blocks for 5G NR FR2**

Subcarrier bandwidth (kHz)	50 MHz	100 MHz	200 MHz	400 MHz
	# RB			
60	66	132	264	N/A
120	32	66	132	264

Equations (3-1)-(3-5) stands also for 5G NR; Data rate is computed again from the Shannon formula, making explicit the dependence from the numerology parameter chosen for the resource block:

$$r_{i,j} = \frac{B_j(\mu)}{10 \cdot 2^\mu} \log_2(1 + SINR_{i,j}). \quad (3-7)$$

### 3.3.3 Satellite Base Station

The Satellite Base Station class models an implementation of an Inmarsat geostationary satellite with telecommunication payload, since no standards are publicly available. Anyway,

since the modularity of the simulator, it is possible to rewrite the Satellite Base Station class to implement different kinds of satellite communication systems (e.g., LEO satellites).

The satellite telecommunication system implemented in the simulator allocates different user requests using TDM/TDMA. TDM/TDMA uses all the available bandwidth for each transmitted symbol, but the user requests are multiplexed by allocating them into time slots (that are grouped in time frames). Each symbol, that carries user data/overhead, is transmitted in a portion of the time frame.

Given a time frame, as indicated in Figure 3-3,  $n_{tot}$  symbols are available to be allocated to UE transmissions and to their overhead (e.g., synchronization, headers, etc.). For what regards overhead terms,  $n_{sync}$  symbols each time frame are used for synchronization purposes; a header (of length  $n_{head}$ ) is attached to every UE communication and there is a guard space of  $n_{space}$  symbols between each communication, that is used to avoid intra-RAT interference.

In the implementation of the Satellite AP, only a slice of the total symbols is dedicated to the multi-RAT system ( $n_{slice}$  symbols).

Moreover, according to the reference implementation, it is not possible to allocate an arbitrary number of symbols for each UE request, but the minimum allocation block is composed by  $n_{block}$  symbols.

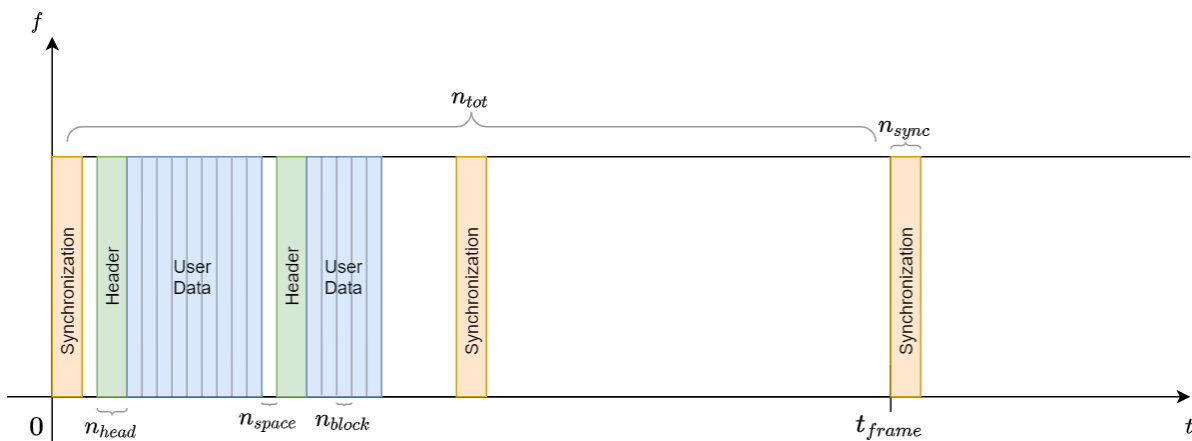


Figure 3-3 Frame structure of satellite telecommunication system .

As indicated by the Inmarsat implementation, the following parameters has been chosen:

- $n_{tot} = 120832$  symbols, transmitted in a time frame of 2ms
- $n_{sync} = 288$  symbols each, with 2 synchronization messages each time frame
- $n_{head} = 280$  symbols for each UE communication
- $n_{space} = 64$  symbols
- $n_{slice} = 39104$  symbols, that are about a third of the total symbols
- $n_{block} = 64$  symbols

The Satellite Base Station class, once received a connection request by a UE, has to determine the number of blocks of symbols to be allocated to the UE in order to satisfy its data rate request.

Again, the first term to compute is the SINR, that is expressed as equations (3-1)-(3-5). The noise term of equation (3-4) is different in this case since the bandwidth used by a symbol is the total bandwidth of the satellite AP (due to TDM/TDMA).

In the same way, equation (3-6) has to be slightly changed in order to address the TDM/TDMA frame structure:

$$r_{i,j} = \frac{B_j}{n_{tot}} \log_2(1 + SINR_{i,j}), \quad (3-7)$$

where now  $B_j$  is the total bandwidth of the satellite AP  $j$  and  $n_{tot}$  is the total number of symbols belonging to a time frame.

### 3.4 Pathloss Models

As shown in the previous section, the role of pathloss is crucial to compute correctly the RSRP and the SINR, and consequently to estimate the data rate obtainable by the UE. In the simulator two pathloss models have been implemented, to be applied both to satellite APs and to terrestrial APs, in order to capture the very different propagation models of the two AP types.

Moreover, given the simulator flexibility, it is possible to add new pathloss models to be applied by the Base Station classes in an easy way, with the only constraint to expose a *compute\_path\_loss()* function, that will be called by the AP classes to actually compute the RSRP for a given UE.

#### 3.4.1 Cost Hata Pathloss model

The pathloss model used for terrestrial APs is the Cost Hata Pathloss model [7]. This is a statistical model working for urban, suburban and rural environments, and it has been chosen because it keeps the complexity of the simulator low (there is no need of mapping buildings, obstacles, etc. on the map) while still having good modelling results. This model, shown in equation (3-8), is valid for carrier frequencies  $f \in [150MHz, 2000MHz]$ , that are most of the frequency bands used by 4G LTE and 5G NR (except for 5G NR FR2).

$$L_{i,j}^{costhata}_{[dB]} = C_0(f) + C_f \log_{10} \frac{f}{MHz} - b(h_j) - a(h_i, f) + \\ + (44.9 - 6.55 \log_{10} h_j) \log_{10} \frac{d_{i,j}}{Km} + C_m(f), \quad (3-8)$$



where:

$$C_0 = \begin{cases} 69.55 & \text{if } 150\text{MHz} \leq f \leq 1500\text{MHz} \\ 46.3 & \text{else} \end{cases}$$

$$C_f = 26.16$$

$$b(h_j) = 13.82 \log_{10} \frac{h_j}{m}$$

$$a(h_i, f) = \begin{cases} 8.29 (\log_{10}(1.54h_i))^2 & \text{if } 150\text{MHz} \leq f \leq 300\text{MHz} \\ 3.2 (\log_{10}(11.75h_i))^2 & \text{else} \end{cases}, \text{ if urban environment}$$

$$a(h_i, f) = (1.1 \log_{10} f - 0.7)h_i - 1.56 \log_{10} f + 0.8, \text{ if rural or suburban environment}$$

$$C_m(f) = \begin{cases} 0 & \text{if } 150\text{MHz} \leq f \leq 1500\text{MHz} \\ 3 & \text{else} \end{cases}, \text{ if urban environment}$$

$$C_m(f) = \begin{cases} -2 \left( \log_{10} \frac{f}{28} \right)^2 - 5.4 & \text{if } 150\text{MHz} \leq f \leq 1500\text{MHz} \\ 0 & \text{else} \end{cases}, \text{ if suburban environment}$$

$$C_m(f) = -4.78(\log_{10} f)^2 + 18.33 \log_{10} f - 40.94, \text{ if rural environment}$$

with  $f$  is the carrier frequency of the AP,  $h_j$  is the AP  $j$  height,  $h_i$  is the UE  $i$  height and  $d_{i,j}$  is the cartesian distance between AP  $j$  and UE  $i$ .

Moreover, this model requires  $h_j \in [30m, 200m]$  and  $h_i \in [1m, 10m]$ .

### 3.4.2 Free Space Pathloss model

The Free Space Pathloss (FSPL) model shown in equation (3-9) has been used as pathloss model for Satellite APs. This model is particularly adequate for Satellite APs, since the communication with UEs usually happens in the open air and without obstacles. Moreover, this pathloss model has been chosen because, even if it is very simple, it models well the satellite pathloss, avoiding considering terrestrial geo-reference systems (e.g., coordinates) and the terrestrial curvature.

Given AP  $j$  and UE  $i$  with their relative cartesian distance  $d_{i,j}$ , then the Free Space Pathloss model is as follows:

$$L_{i,j}^{FSPL} = \left( \frac{4\pi d_{i,j} f}{c} \right)^2, \quad (3-9)$$

where  $f$  is the carrier frequency of AP  $j$  and  $c$  is the speed of light.

As a sidenote, in case of particular type of satellites this pathloss model may not be accurate enough. For these cases often pathloss mean value, computed by the satellite producers, is made available for research purposes and so a specific (and very simple) pathloss class can be added to the simulator.

## 4 Description of KR simulator

The KR Simulator has been designed and customized for the testing and the development of a load balancing algorithm in a multi-RAT network with NTN and 5G networks [16], considering both intra-RAT and inter-RAT load balancing.

The main functionalities considered are intra-RAT and inter-RAT offloading of the UEs from the overloaded cells. In order to represent the load of each RAT the radio resource utilization ratio (RRUR) is proposed. With the RRUR, the main algorithm estimates the load status of the currently overloaded cells and the candidate target cells and chooses the UEs for offloading. An adaptive threshold is used to adopt the network traffic and measure the overload status of a cell. When offloading UEs, the system uses a 5G QoS model to maintain different queues for delay-sensitive and delay-tolerant data flows. Some KR Simulator outputs shown in this chapter prove that the main proposed algorithm ensures a balanced load among terrestrial cells as well as including satellite cells.

### 4.1 Background

The mobility of UEs possibly results in a load imbalance across the cells in 5G RAT network [17]. The imbalance in the network affects the QoS of UEs and is an inefficient utilization of available resources. Furthermore, the requirement for high data rates for UEs and the uneven distribution of UEs in the network lead to overutilization of resources in some cells. To overcome these problems, it is necessary to share the load among the cells so that network resources are utilized efficiently to balance the network. For that purpose, intra-RAT load balancing is performed to balance load distributions in order to maintain an appropriate end-user experience and good network performance.

With intra-RAT load balancing, the load from an overloaded cell moves to under loaded neighbouring cells. The source and target cells are part of the same RAT. However, sometimes UEs cannot move to neighbouring cells due to a scarcity of resources and limited coverage. This affects efficient load balancing among cells and decreases QoS of the users. The combination of multiple RATs, referred to as a multi-RAT network, is considered for wireless networks to increase resource availability as well as coverage. The multi-RAT network enhances the QoS of UEs, because different RATs can support different services. Furthermore, the UEs access the radio resources of multiple RATs and dynamically route particular traffic to a RAT to satisfy QoS. In the multi-RAT network, it is necessary to determine which RAT should serve which UEs to increase network performance and satisfy QoS of the UEs. Both intra-RAT and inter-RAT load transfers from overloaded cells in the multi-RAT network lead to a well-balanced network and increase network throughput. Moreover, a common load metric is also necessary to measure the load of each RAT for the load balancing in a multi-RAT network. Based on the load metric, radio resources utilization of RATs can be determined and used to divide the network load among the cells of different RATs.

### 4.2 Network Architecture

KR simulator considers the coexistence of TNs and NTNs, as shown in Figure 4-1. The TN includes a set of 5G cells, T, with next-generation node B (en-gNB or gNB). An Xn interface is considered for direct communication between the neighboring gNBs. For the NTN, we consider a GEO satellite that is connected to the NTN gNB through a ground station. The GEO

satellite is always in the same relative position and therefore, inter-satellite handoff is unnecessary, and there is no Doppler shift. The terrestrial gNB connects with the NTN gNB via Xn to share control information. Management of traffic loads is provided over the Xn interface. For the core network (CN) connection, an NG interface is considered between gNBs and the 5G CN. The multi-connectivity feature for UEs is adopted in which a terrestrial gNB acts as an anchor and the satellite as a slave node. We consider the 3C configuration for the control plane and the 1A configuration for the user plane [19]. The 3C configuration splits the bearer in the anchor, which is the control plane only at the cellular gNBs, whereas the 1A configuration has a separate radio bearer for each of the UEs and splitting of the user plane occurs in the CN. Figure 4-2 shows the 1A and 3C configurations for the radio protocol architecture.

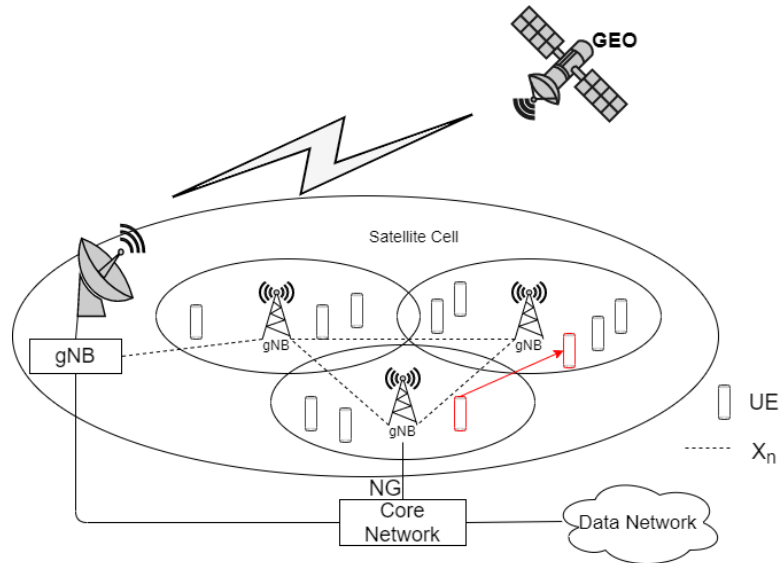


Figure 4-1 Network architecture for KR simulator.

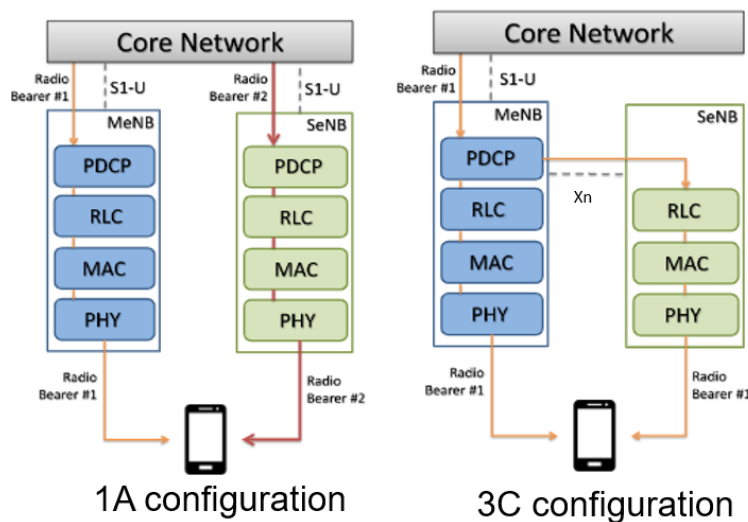


Figure 4-2 Radio protocol architecture for multi-connectivity [23].

There are two classes of the UEs' data flow; one class has a delay-tolerant flow, and the other class has a delay-sensitive flow. The packet delay budget (PDB) is defined by 3GPP for data flows in 5G system [20]. The PDB of flows greater than the satellite propagation delay are considered delay-tolerant flows, and flows with a PDB less than the satellite propagation delay are considered delay-sensitive flows. To support multiple data flows, different numerologies are introduced in 5G [21]. Based on the data flows, each UE uses different 5G numerologies, i.e., carrier spacing (CS). Multiple numerologies for 5G New Radio (NR) are shown in Table 4-1. A physical resource block (PRB) is the smallest unit of a resource block allocated to UEs by a gNB. Each 5G terrestrial cell has some available PRBs based on the system bandwidth and the CS. Furthermore, the PRB bandwidth depends on the CS, and one PRB occupies bandwidth equal to the number of consecutive sub-carriers into the CS. For the NTN, satellite bandwidth is assigned to UEs according to their required data rates using the Shannon capacity formula [22].

**Table 4-1: Simulation parameters**

Subcarrier spacing(KHz)	Slot Duration(ms)	Max Bandwidth (MHz)
15	1	50
30	0.5	100
60	0.25	200
120	0.125	400

### 4.3 Measurement Report Triggering

The purpose of the measurement report is to transfer measurement results from the UEs to the network. In 5G, reference signal received power (RSRP) measurements are important for mobility management. A network lets UEs report the signal quality of the current cell, i.e., serving cell, and the target cell. The 3GPP defined several sets of predefined measurement report mechanisms to be executed by UEs and these predefined measurement report types are called event. For 5G, there are six events (A1, A2, A3, A4, A5 and A6) for intra-RAT measurements and two events (B1 and B2) for inter-RAT measurements were specified and discussed in [24]. We consider both intra-RAT and inter-RAT offloading for load balancing among 5G cells. However, events for intra-RAT measurements are used in this work to determine edge UEs and target neighboring cells for intra-RAT offloading. UEs are in the coverage area of a satellite cell, therefore, there is no need to determine edge UEs and target neighboring cells for inter-RAT offloading. Data flows of the UEs and traffic loads of the serving cells are considered for inter-RAT offloading.

Two events (A3 and A4) are considered for intra-RAT load balancing in this paper. Event A3 is the most suitable for finding the best neighboring cells for handover of UEs [18], and A3 is widely used for inter-RAT handovers in wireless networks [25]. Event A3 is triggered when the signal of a neighboring cell is offset better than the serving cell, and UEs report measurements to the serving cell. The following equation shows the trigger condition of the A3 event

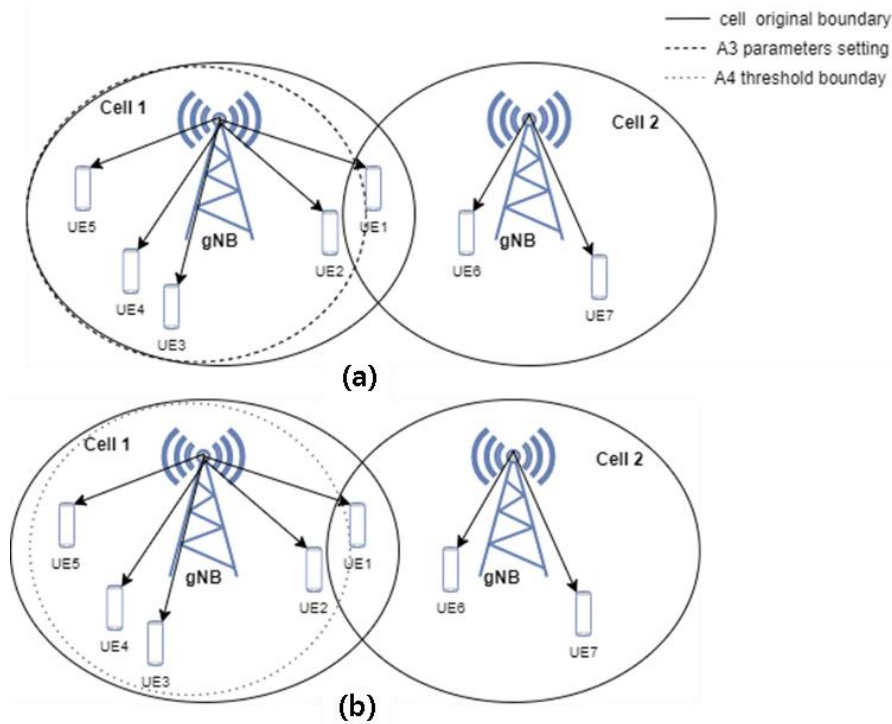
$$M_n + O_{fn} + O_{cn} - H_{ys} > M_p + O_{fp} + O_{cp} + O_{ff}$$

where  $M_n$  and  $M_p$  are the RSRP of the neighboring cell and the current cell, respectively.  $Of_n$  and  $Of_p$  are the frequency-specific offsets, and  $Ocn$  and  $Ocp$  are the cell individual offsets for the target and serving cells, respectively;  $Hys$  is the hysteresis parameter; and  $Off$  is the A3 event offset between the serving cell and the target neighboring cell. The frequency-specific offsets are used for inter-frequency handover, and therefore, we forgo  $Of_n$  and  $Of_p$  in this report. The intra-RAT handover decision changes based on the values of  $Ocn$ ,  $Ocp$ , and  $Off$ . Based on the load status of a cell load, the A3 variables ( $Ocn$ ,  $Ocp$  and  $Off$ ) are changed to intentionally delay or hasten the handovers of UEs.

Consider Figure 4-3a, where cell 1 is overloaded with five UEs, and neighboring cell 2 has less of a load. There are two edge UEs in cell 1, i.e., UE 1 and UE 2, which can be moved to a neighboring cell to reduce the cell 1 load. Either by decreasing  $Ocn$  and increasing  $Ocp$ , the range of cell 1 decreases and UE 1 can be offloaded to the cell 2 gNB to balance the network. For offloading UEs to a particular neighboring cell, only the  $Ocn$  parameter is adjusted, based on the RRUR of the serving cell. Hence, event A3 will be used to find a suitable target cell for offloading UEs of overloaded cells for intra-RAT load balancing. Moreover, information on the edge UEs of the overloaded cells is also needed prior to handover. For that purpose, event A4 is used to sort the outskirts UEs of the cell. Since event A4 is triggered when the RSRP of neighboring cell  $M_n$  becomes better than a provided threshold,  $Thresh$ . So, event A4 is defined as

$$M_n + Of_n + Ocn - Hys > Thresh \quad (4-1)$$

Measurement reports by UEs after triggering event A3 are used to determine the threshold for A4 events, as done in [18]. UEs that satisfy condition (4-1) will report the RSRP for the serving cell as well as neighboring cells. For example, in Figure 4-3b, UE 1 reports measurements to cell 1 because it is outside the A4 event boundary of serving cell 1. Hence, cell 1 reduces the  $Ocn$  of target cell 2 to offload UE 1 to the target cell. Based on the event A4 boundary, a cell



**Figure 4-3 Events A3 and A4 for the algorithm: (a) A3 event parameter settings for load balancing, and (b) getting candidate edge UEs and target cell information using A4 event parameters.**

will obtain edge UEs' information and will list candidate UEs,  $E = \{e_1, \dots, e_n\}$  where  $e_i$  is the edge UE  $i$  for  $0 \leq i \leq n$ , for intra-RAT load balancing.

#### 4.4 Flow Classification in 5G

To exploit multi-RAT connectivity in 5G networks, it is necessary to steer traffic across the available access networks optimally. A delay incurred by satellite access is orders of magnitude higher than its terrestrial counterpart. That is, in addition to achieving balanced radio resource utilization, we need to guarantee that delay-sensitive traffic is forwarded only through terrestrial access, whereas delay-tolerant traffic can be served through a satellite when the terrestrial network load surpasses a given threshold. To do so, it is necessary to classify data flows into different QoS classes.

In the 5G CN, a session management function (SMF) is introduced for the 5G QoS model [20]. The SMF manages the protocol data unit (PDU) session, which is a logical connection between UEs and the data network (DN), and the related QoS flows in the CN. The SMF assigns a QoS flow identifier (QFI) and a QoS profile to a flow based on information provided by the policy control function. A QFI value corresponds to a particular QoS flow, and each QoS flow is identified by the QFI. The service data flows (SDFs), which are groups of IP flows/packets, are classified based on IP flows received from a DN. Later, SDFs with the same QoS characteristics are grouped together in the 5G QoS flow and are marked with the same QFI. The SMF provides the user plane function (UPF) with the packet detection rules (PDRs) for mapping SDFs to the QoS flows. Each QoS flow is defined by a QoS profile, and the QoS profile identifies the 5G QoS characteristics with a 5G QoS Identifier (5QI). Based on the 5QI value, 5QI-to-QoS characteristic mapping is provided [20]. Furthermore, the PDB is defined for a QoS flow based on the 5QI value of the flow [20]. For example, the PDB is 150 ms for a 5QI value equal to 2, and the flow is considered delay-sensitive. The QoS flow model based on [26] is shown in Figure 4-4.

For the multi-RAT network, SDFs with same QoS flow can be directed to a particular RAT, and then, the SMF sends the QoS profile to the gNB via the access and mobility management function (AMF). Our work exploits the SMF service to maintain different queues for delay-sensitive and delay-tolerant flows by offloading flows to different RATs. Based on the QoS flows of the UEs, the data planes of the UEs switch to different RATs using the 5G QoS model.

#### 4.5 Load Measurement in 5G Multi-RATs

Proper load measurement of cells is crucial for optimizing the performance of a network through load balancing. For that purpose, a common load measurement metric is needed to measure the load of each RAT in a multi-RAT network. For LTE networks, PRB allocation information, called the resource block utilization ratio (RBUR), is mainly used to determine overloaded cells. For any given time,  $T$ , the average RBUR of a cell  $n$ ,  $RB_n$ , is expressed in [18] as

$$\overline{RB}_n = \frac{1}{T \cdot N_{PRB}} \sum_{\tau \in (t-T, t)} RB_n, \quad (4-2)$$



---

30

Based on the common load measure metric, i.e., RRUR, load distribution among cells of different RATs is determined. A higher RRUR of a cell indicates that the cell has a higher load to serve and fewer available resources. If RRUR is more than a predefined threshold, the cell is overloaded, and UEs moving to that cell will either be dropped or will experience low data rates. Hence, new UEs in an overloaded cell will reduce the per UE data rates. Therefore, it is necessary to reduce the load of the overloaded cell by switching the data plane of some UEs to a lightly loaded cell or another RAT. Furthermore, the RRUR overcomes the different physical layer channels properties of each RAT in a 5G multi-RAT network. Hence, the physical layer channel of each RAT does not affect the problem formulation of load balancing in 5G integrated satellite-terrestrial networks.

#### 4.6 Problem Formulation

For KR simulator, it was performed problem formulation as follows. In a network, if the RRUR of a RAT cell is close to 1, a user that moves into the cell will either be dropped or will experience a low data rate. Hence, a new user in an overloaded cell will reduce the per-user data rate, which affects the QoS of the UEs. To reduce the RRUR of a cell, load balancing among cells is necessary. In load balancing, the total network load is shared among the cells. For that purpose, loads from overloaded cells offload to underloaded neighbouring cells in the same RAT, referred to as intra-RAT load balancing. Another option is inter-RAT load balancing in which UEs of the overloaded cell move to another RAT to balance the cellular network.

For the algorithms, it was formulated the problem of load balancing as one of reducing the RRUR of the terrestrial cells to a target RRUR,  $\bar{\beta}$ , such that the square distance between the cell RRUR and  $\bar{\beta}$  is minimized. A multi-RAT network consists of a set of cells,  $N$ , in which there is a set of terrestrial cells,  $T$ , and a satellite cell,  $S$ , i.e.,  $N = T \cup S$ , and  $I$  users. The problem can be expressed as

$$\begin{aligned} \min \quad & \sum_{\forall n \in T} |\bar{\beta} - \beta_n|^2 \\ \text{subject to: } & \beta_S \leq Thr_{adp}, \\ & \beta_\kappa^i \geq \rho_i, \quad \kappa \in \mathcal{N} \end{aligned} \quad (4-5)$$

where  $\beta_n$  is the RRUR of terrestrial cell  $n$ ,  $\beta_S$  is the RRUR of a satellite cell  $S$ ,  $Thr_{adp}$  is the adaptive threshold,  $\beta_\kappa^i$  is the resource allocated to user  $i$  by cell  $S$ , and  $\rho_i$  is the resources required by user  $i$ , from which  $\rho_i$  is calculated based on the minimum data rate required by UE  $i$ . The cell allocates resources to UEs based on the UEs' required data rates and the channel quality.

To estimate  $\bar{\beta}$  mean square estimation of  $\beta_n$  can be phrased as was done in [27]. Consider random variable  $y$  and the mean square estimation of  $y$  by constant  $c$  as follows:

$$E[(c - y)^2] = \int_{-\infty}^{\infty} (c - y)^2 f(y) dy$$

The difference,  $|c - y|$ , is minimum if

$$\frac{de}{dc} = 0$$



Because the difference depends on  $c$ , constant  $c$  is equal to

$$c = \int_{-\infty}^{\infty} yf(y)dy$$

and  $E[y] = \int_{-\infty}^{\infty} yf(y)dy$ ,

$$c = E[y] \tag{4-6}$$

Considering (4-5) and (4-6),  $\bar{\beta}$  is equal to

$$\bar{\beta} = E[\beta_n] \tag{4-7}$$

Hence,  $\bar{\beta}$  is expected RRUR of terrestrial cells.

#### 4.7 Main Algorithm for Load Balancing

The proposed algorithm balances the load in 5G RAT based on data flows of UEs and by considering cell load status in a 5G multi-RAT network. The algorithm runs in each 5G gNB and initiates load balancing when terrestrial cells are overloaded. The proposed algorithm consists of three parts: information gathering, intra-RAT load balancing, and inter-RAT load balancing. For load balancing in 5G cells, the algorithm first gathers information on the load status of the cells using a function call *info\_gather*. After that, loads from overloaded 5G cells are released to underloaded cells by calling a function called *intRALb*. At the end, based on the load status of the cells, the algorithm calls a function called *intERlb* to transfer terrestrial loads to NTN RAT. Each part of the proposed algorithm is described in the subsections below. Algorithm 1 shows the proposed algorithm's process and Table 4-2 defines the notations used in the algorithm.

##### Algorithm 1 Proposed Load Balancing Algorithm

```

1: function Load_Balance (void)
2:   info_gather ()
3:   for all  $o \in \mathcal{O}$  do
4:     intRALb ( $\beta_o, Thr_{adp}$ )
5:     Determine  $\beta_S$  using (4)
6:     if  $\beta_S \leq Thr_{adp}$  and  $\beta_o \geq Thr_{adp}$  then
7:       intERlb ( $\beta_o, \beta_S, Thr_{adp}$ )
8:     end if
9:   end for

```

Table 4-2: Simulation parameters

Notations	Definitions
$\bar{\beta}$	Average RRUR of 5G cells
$\beta_n$	RRUR of terrestrial cell $n$
$\beta_S$	RRUR of the satellite cell
$\hat{\beta}_n^i$	Estimated resource utilization of cell $n$ by UE $i$
$\rho_i$	Required PRBs by UE $i$
$\Omega_i$	Allocated bandwidth assigned to UE $i$ by the satellite
$Thr_{adpt}$	Adaptive threshold to find an overloaded cell
$thr_{init}$	Initial threshold

#### 4.7.1 Information Gathering

For gathering the information, the function, `info_gather()`, measures the load of terrestrial cells, i.e., the RRUR, using (4-3), and then, the average load of 5G cells is calculated using (4-7). To estimate the overload status of a cell, adaptive threshold  $Thr_{adpt}$  is determined as follows

$$Thr_{adpt} = \max(\bar{\beta}, thr_{init}) \quad (4-8)$$

where  $thr_{init}$  is the fixed initial threshold used to determine whether there is a need for load balancing in the network. The adaptive threshold,  $Thr_{adpt}$ , is used to adopt the network load. The network load can vary over time because of user mobility and variances in required data rates of the UEs. After that, the algorithm estimates the overload status of a cell by using the following condition

$$\beta_n > Thr_{adpt}, \quad n \in \mathcal{T} \quad (4-9)$$

and establishes a set,  $\mathcal{O}$ , of terrestrial cells that satisfy the above condition, where  $\mathcal{O} \subset \mathcal{T}$ . The process of information gathering is summarized in Algorithm 2.

#### 4.7.2 Intra-RAT Load Balancing

In intra-RAT load balancing, UEs from an overloaded cell,  $o \in \mathcal{O}$ , move to underloaded neighboring cells. The function gathers information on the edge UEs that are moved from overloaded cell  $o$ . For that purpose, the function establishes a set,  $\nu_o$ , of edge UEs that report measurements to serving cell  $o$  based on the A3 event measurement reports. Then, another set of UEs is created,  $E_o \subseteq \nu_o$  which report the RSRPs of neighboring cells to the serving cell  $o$  during an event A4. The UEs in  $E_o = \{e_1, \dots, e_n\}$  are then sorted in ascending order of serving cell RSRPs and the UEs are arranged according to data flow type. For intra-RAT load balancing, first the UEs of  $E_o$  with delay-sensitive flows, and then UEs with delay-tolerant flows, move to underloaded neighboring cells one by one based on the load status of cell  $o$ .

Based on event A3, the target neighboring cell is determined in order to offload UE  $e_1 \in E_o$  from overloaded cell  $o$ . The set  $\Gamma_{e_1} = \{\Gamma_1, \Gamma_2, \dots, \Gamma_m\}$  denotes the neighboring cells reported by UE  $e_1$  to serving cell  $o$  under event A4. The neighboring cells are listed in descending order

---

**Algorithm 2** Information Gathering

---

```

1: function info_gather ()
2:   Get RRUR of terrestrial cells  $\mathcal{T}$ 
3:   Compute average 5G cell load  $\bar{\beta}$ 
4:   Determine  $Thr_{adpt}$ 
5:   Establish a set of overloaded cells,  $\mathcal{O} \subseteq \mathcal{T}$ 
6:    $Thr_{adpt}, \mathcal{O}$ 

```

---

of RSRP values, i.e., the RSRP for  $\Gamma_1$  is greater than  $\Gamma_2$ . To offload UE  $e_1$ , the algorithm estimates  $\hat{\beta}_{\Gamma_k}^{e_1}$ , the resource utilization of target cell by UE  $e_1$ .  $\hat{\beta}_{\Gamma_k}^{e_1}$  is calculated based on (4-3)

as follows:

$$\hat{\beta}_{\Gamma_k}^{e_1} = \frac{\rho_{e_1} \zeta}{\omega_{\Gamma_k}} \quad (4-10)$$

where  $\rho_{e_1}$  is the PRB of cell  $\Gamma_k$  required by UE  $e_1$ ,  $\zeta$  is the bandwidth of the resource block, and  $\omega_{\Gamma_k}$  is the total bandwidth of target cell  $\Gamma_k$ . Before offloading UE  $e_1$  to cell  $\Gamma_k$ , the algorithm checks the following conditions in order to restrict the target cell load to below overload status and to avoid unnecessary offloading of UEs to neighboring cells, i.e., to avoid ping-pongs:

$$\beta_{\Gamma_k} + \hat{\beta}_{\Gamma_k}^{e_1} < Thr_{adpt} \quad (4-11)$$

$$\beta_o - \hat{\beta}_o^{e_1} > \beta_{\Gamma_k} + \hat{\beta}_{\Gamma_k}^{e_1}. \quad (4-12)$$

If the above conditions are satisfied, UE  $e_1$  moves to target cell  $0k$ . After offloading UE  $e_1$ , the RRURs of the previous and current serving cells are updated as follows:

$$\begin{aligned} \beta_o &= \beta_o - \hat{\beta}_o^{e_1}, \text{ and} \\ \beta_{\Gamma_k} &= \beta_{\Gamma_k} + \hat{\beta}_{\Gamma_k}^{e_1}. \end{aligned} \quad (4-13)$$

Then,  $\bar{\beta}$  and  $Thr_{adpt}$  are updated. The same process repeats for each UE in  $E_o$  based on the cell loads. Algorithm 3 summarizes the function  $intRAIb(\beta_o, Thr_{adpt})$ .

---

**Algorithm 3** Intra-RAT Load Balancing

---

```

1: function intRAIb ( $\beta_o, Thr_{adpt}$ )
2:   Get candidate edge UEs,  $E_o$ 
3:   Sort  $E_o$  in ascending order of RSRP and arrange
   according to data flow type.
4:   for  $i \leftarrow 1 : |E_o|$  do
5:     Determine set  $\Gamma_{e_i}$  of target cells for UE  $e_i$ 
6:     for  $k \leftarrow 1 : |\Gamma_{e_i}|$  do
7:       Estimate  $\hat{\beta}_{\Gamma_k}^{e_i}$  using (10)
8:       if (11) and (12) are satisfied then
9:         Offload flow of UE  $e_i$  to the target cell  $\Gamma_k$ 
10:        Update RRUR information
11:         $\beta_o \leftarrow \beta_o - \hat{\beta}_o^{e_i}$ 
12:         $\beta_{\Gamma_k} \leftarrow \beta_{\Gamma_k} + \hat{\beta}_{\Gamma_k}^{e_i}$ 
13:        Update  $\bar{\beta}$  and  $Thr_{adpt}$ 
14:        break;
15:       end if
16:     end for
17:     if  $\beta_o \leq Thr_{adpt}$  then
18:       break;
19:     end if
20:   end for
21: return  $\beta_o, Thr_{adpt}$ 

```

---

### 4.7.3 Inter-RAT Load Balancing

After intra-RAT load balancing, the algorithm again checks the load status of the cell  $o$ . If the cell is still overloaded, i.e.,  $\beta_o > Thr_{adpt}$ , the algorithm performs inter-RAT load balancing by transferring the load of cell  $o$  to satellite cell  $S$  by offloading the delay-tolerant flows of UEs if

$$\beta_s < Thr_{adpt} \quad (4-14)$$

To release the load of 5G cells to a satellite cell, the function generate a set of UEs  $E_o = \{\varepsilon_1, \dots, \varepsilon_n\}$ , where  $\varepsilon_o$  denotes the UEs of cell  $o$  with delay-tolerant data flows. After that, UEs in  $\varepsilon_o$  are sorted in ascending order of RSRPs from cell  $o$  and data flows of UEs in  $\varepsilon_o$  are offloaded to a satellite link one by one. Before offloading UE  $\varepsilon_1$ , the function first estimates  $\hat{\beta}_S^{\varepsilon_1}$ , i.e., the resource utilization of the satellite by UE  $\varepsilon_1$ . Then,  $\hat{\beta}_S^{\varepsilon_1}$  is calculated using the Shannon formula based on the data rate required by the UE:

$$\hat{\beta}_S^{\varepsilon_1} = \frac{\Omega_{\varepsilon_1}}{\omega_{sat}} \quad (4-15)$$

The above condition prevents the satellite from being overloaded. For the offloading of data flows, the UPF directs the flow of UE  $\varepsilon_1$  to NTN gNB as we considered the separate user plane for each RAT. And the SMF sends QoS policy information based on the 5QI to the NTN gNB through AMF as described II-C. The proposed algorithm offloads the UEs to the satellite cell irrespective of the position of UEs in the cell, since all UEs are within the coverage area of the

satellite. After offloading of UE  $\varepsilon_1$ , the algorithm updates the RRURs of terrestrial cell  $o$  and satellite cell  $S$  as follows:

$$\begin{aligned}\beta_S &= \beta_S + \hat{\beta}_S^{\varepsilon_1}, \text{ and} \\ \beta_o &= \beta_o - \hat{\beta}_o^{\varepsilon_1}.\end{aligned}\tag{4-16}$$

Then the algorithm updates  $\bar{\beta}$  and  $Thr_{adpt}$ . The algorithm again checks the RRURs of the satellite and cell  $o$  and repeats the process for each UE of  $E_o$ . Algorithm 4 summarizes the function `intERlb` ( $\beta_o, \beta_S, Thr_{adpt}$ ).

When UEs move to a satellite, they will experience a long delay. However, offloading UEs with delay-tolerant data flows will not affect the QoS of the UEs, whereas UEs with delay-sensitive data are served by the 5G RAT. Similar to NTN, the proposed algorithm can be extended to other RATs, i.e., unmanned aerial vehicle (UAV) communication systems [28]. Based on the RRUR, the load status of a RAT can be determined and UEs from an overloaded cell move to the RAT, taking into account the minimum QoS requirements of the users.

We analyzed the computational complexity of the proposed algorithm using big  $O$  notation. For load balancing of a terrestrial cell network, the considered number of cells in a multi-RAT network under the proposed algorithm is  $|N|$ , which represents the number of cells in set  $N$ . Set  $N$  consists of  $|T|$  terrestrial cells and a satellite cell  $S$ . Therefore, the maximum numbers of cells to be considered for intra-RAT and inter-RAT load balancing are  $|T|$  and  $|N|$ , respectively. Similarly, the maximum numbers of target cells in intra-RAT and inter-RAT offloading are limited by the  $|T|$  terrestrial cells and satellite cell  $S$ , respectively. In addition to

---

**Algorithm 4** Inter-RAT Load Balancing

---

```

1: function intERlb ( $\beta_o, \beta_S, Thr_{adpt}$ )
2:   Get list of UEs with delay-tolerant flows,  $\mathcal{E}_o$ 
3:   Sort UEs in ascending order of RSRP
4:   for  $i \leftarrow 1 : |\mathcal{E}_o|$  do
5:     if  $\beta_S < Thr_{adpt}$  then
6:       Estimate  $\hat{\beta}_S^{\varepsilon_i}$  using (14)
7:       if (14) is satisfied then
8:         UPF offloads flow of UE to satellite gNB
9:         Update RRUR
10:         $\beta_S \leftarrow \beta_S + \hat{\beta}_S^{\varepsilon_i}$ 
11:         $\beta_o \leftarrow \beta_o - \hat{\beta}_o^{\varepsilon_i}$ 
12:        Update  $\bar{\beta}$  and  $Thr_{adpt}$ 
13:      end if
14:    else
15:      break;
16:    end if
17:    if  $\beta_o \leq Thr_{adpt}$  then
18:      break;
19:    end if
20:  end for

```

---

the number of cells for load balancing, the algorithm also considers UEs in the network, and the number of considered UEs under the algorithm is  $I$ .

Since there are, at most,  $|T|$  serving and target cell pairs and  $I$  UEs involved in intra-RAT offloading, the loop in the intra-RAT offloading function of Algorithm 3 should take  $O(|T|) + O(I)$ . In the case of inter-RAT offloading, there are, at most,  $I$  UEs, and only one pairing of a terrestrial serving cell and a target satellite cell involved. Hence, the loop in the inter-RAT offloading function of Algorithm 4 should take  $O(I)$ . Furthermore, the number of overloaded cells is bounded by the number of terrestrial cells,  $|T|$ . So, the overall computational complexity of the proposed load balancing algorithm becomes  $O(|T|^2) + O(I|T|)$ . Generally,  $I \gg |T|$ , so we can say that the computational complexity for the proposed load balancing algorithm is  $O(I|T|)$ .

## 5 Tial multi-connectivity scenarios for PoC

This section briefly discusses the PoC scenarios relevant for multi-connectivity, highlighting for both EU and KR partners their rationale, main requirements and expected outcomes. From such analysis, the following chapter will discuss how the consortium selected which algorithms to use for the project demonstration and how the algorithms were tailored or tuned to meet the PoC needs and offer all expected functionalities.

### 5.1 EU side scenarios

With reference to D2.4, that details the service scenarios and applications for the PoC, here we briefly report the main use cases for the demonstration of the multi-connectivity capabilities developed by the EU partners in the scope of WP4. In fact, the design and the selection of the multi-connectivity algorithm to be deployed in the PoC was driven mostly by the functionalities that will have to be demonstrated on the final scenarios of the project.

In particular, regarding the use case 2 “Simultaneous satellite and cellular multi-connectivity”, the scenarios of interest are:

Scenario 1: simultaneous cellular and satellite access with enhanced total throughput

Scenario 2: simultaneous satellite non-stringent QoS services and cellular stringent QoS services delivery to UE

Both multi-connectivity scenarios involve the usage of traffic steering (i.e., the ability to dynamically route traffic over multiple RATs) and the corresponding traffic controller that manages the data flows. In the PoC architecture proposed for this use case, the RAN is constituted by two APs of different RATs: a terrestrial 5G connectivity and a satellite connection.

Scenario 1 is of particular interest, as the amount of traffic to be sent over one RAT or the other has to be determined depending on its performances in a feedback-based procedure, as one of the RATs may show less-than-ideal performance due to unaccounted factors (high interference, malfunctions, excessive load,...). Furthermore, in Scenario 1 the traffic flow to be split belongs to the same PDU session, meaning that a data flow for a single connection has to be divided and the recombined to enable multi-connectivity. On the contrary, in Scenario 2 two different QoS flows are routed separately, the less QoS-stringent one over the lower performing satellite connection and the more demanding flow over the terrestrial connectivity.

Regarding use case 3 “Public safety”, both the scenarios

Scenario 1: unpredictable traffic switching

Scenario 2: predictable traffic switching

highlight the need for a controller able to take decisions depending on the network state (or a prediction on the network operative level).

From the discussion above, it turns out that the algorithm to be selected for the project demonstration requires the following capabilities:

ability to manage multi-connectivity by dividing a single QoS flow into (at least) two separate flows to be routed individually;

ability to deploy a dynamic traffic steering decision, taking a feedback from the network state.



At the same time, given the overall storytelling of the use cases and their targets, a good candidate for the type of traffic to be transmitted during the demonstration was identified in video streaming. For this reason, the EU partners developed its multi-connectivity modules to be compliant and integrated with a custom video streaming platform that will generate the traffic flows needed for the PoC demonstration.

Another functionality that the algorithm of the PoC will need to be able to address is the capability of including QoE-related aspects in its traffic steering decisions. In order to evaluate the video features/characteristics that impacted the most the QoE of a typical user, the consortium organised a questionnaire that was detailed in D4.2. The questionnaire included several questions regarding smartphone and connection usage habits, but also proposed a few videos stream that tested the users for their sensitivity to certain video aspects (in particular, actual resolution, excessive buffering, audio sync issues and bitrate drops). From the analysis of the gathered responses, we observed that no clear clustering difference was to be found in these aspects between the EU and the KR users (slight differences were observed on usage habits, with KR users using more video streaming during trips). The different sensitivities observed were most likely due to the significant age difference between the typical responders (most EU responders were students from Italian and French universities, whereas KR responders were mostly professionals involved in the ICT business), with younger users focusing more on resolution and buffering and older ones focusing more on resolution and video bitrate. From these observations, we decided to enable QoE management by letting the traffic flow controller dynamically adapt some video features (in particular the codec) depending on a real-time feedback of the network performances.

## 5.2 KR side scenarios

Figure 3-1. illustrates a multi-connectivity scenario in KR side. In the automotive playground, a vehicle equipped with both terrestrial and satellite RUs, Traffic Controller, and client hand-sets will be driven. Also, a streaming server is broadcasting both high definition (HD) and 4K video contents to the clients. During the driving, the vehicle can face with an area where cellular signal is disturbed and the traffic controller detects signal blockage of terrestrial link and switch over to the satellite link for the service continuity such that users inside vehicle do not recognize any outage of the streaming service. The quality of the video can be degraded from 4K to HD since the satellite link may not support enough bandwidth capacity for 4K streaming. When the vehicle moves into an area where cellular signal is normal, TC failbacks to terrestrial link including content quality upgrade into 4K if needed.

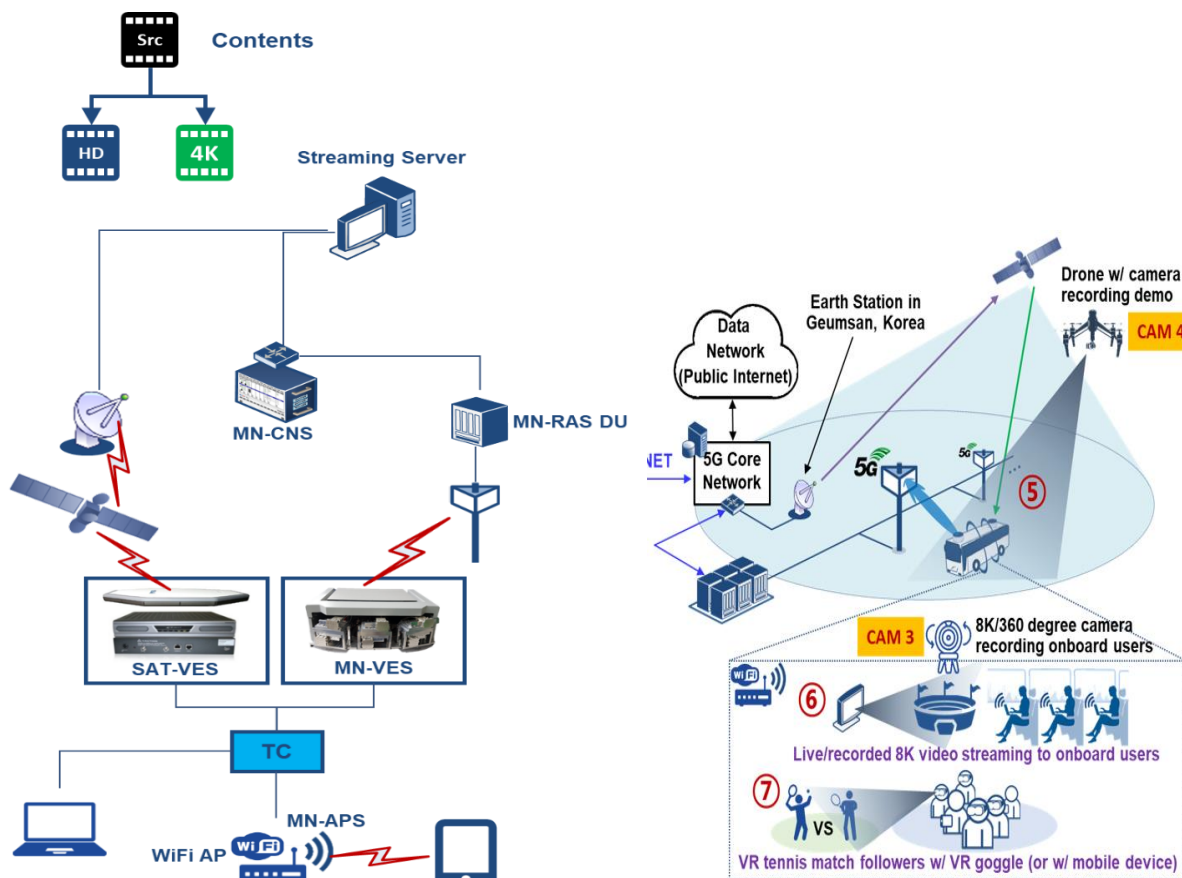


Figure 5-1 KR side multi-connectivity scenario

## 6 Selection of multi-connectivity algorithm for the demonstrator

This chapter discusses how the EU and KR partners selected the algorithms that will be employed in the final PoCs of the project.

The chapter starts with the presentation of the rationale behind the choice of the EU partners, detailing also how the selected algorithms has been adapted to meet the needs of the PoC scenarios (section 6.1.2) and briefly recalling and updating (section 6.1.3) the overview of the algorithm suite that was originally reported in the second issue of D4.2.

The chapter then (section 6.2) illustrates the performance evaluations performed by the KR partners on their algorithm of choice, detailing the simulation environment and how it responds to various different scenario characteristics (e.g., number of users).

### 6.1 EU side Algorithm Selection Rationale

In the light of the multi-connectivity use cases and scenarios, briefly reported in section 5, and of the comparison of the main characteristics of the suite of algorithms developed in the scope of WP4, which we reported after the second project review in the second issue of D4.2 (see its section 4.2 or section 6.1.2 of this document), this section aims at detailing the logic behind the selection of the algorithm that will be deployed in the project PoC.

We start by briefly discussing the functional architecture of the PoC, reported in Figure 6-1, that was developed in strict contact with WP5 activities.

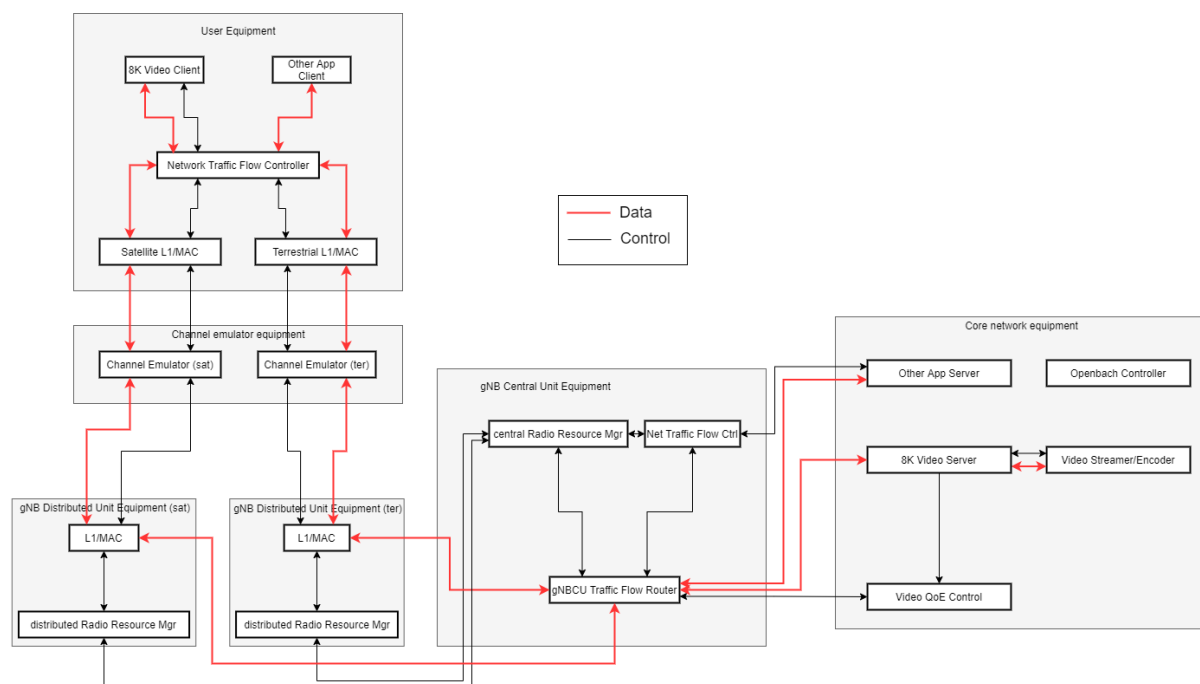


Figure 6-1 Latest EU PoC functional architecture from WP5

The functional blocks whose development is heavily linked to the activities of WP4 are:

- The traffic flow controller
- The traffic flow manager
- The QoE controller

The role of the traffic flow manager is to correctly reconstruct data flows that were split by the steering decisions taken by the traffic flow controller, that is a feedback-based system that oversees the steering decisions for all data flows in the PoC. The QoE controller is a system that has the purpose of tuning the data stream characteristics (e.g., codec) and/or the steering decisions depending on the user preferences.

We start by detailing the design choices behind the most important functional module, that is the traffic flow controller.

As mentioned, we developed an opensource multi-RAT network simulator to test the suite of algorithms developed in T4.2. The numerical validation of the proposed algorithms, that is reported in the second issue of D4.2 (and in their corresponding publications), allowed us to evaluate and tune the algorithms in scenarios similar to the PoC.

Among the algorithms developed by the EU side and reported in D4.2, the ones that offered traffic steering capabilities and were the most suitable to be deployed on the PoC are the following ones:

- Traffic steering and network selection in 5G networks based on reinforcement learning (full algorithm and simulative validation reported in D4.2, second issue, section 5.1)
- Wardrop Equilibrium based control (full algorithm and simulative validation reported in D4.2, second issue, section 5.2).

The former algorithm has the advantage of being able to be deployed without any complex modelling and identification phase, as it is based on Reinforcement Learning, a model-free control framework. The main limitations of the algorithm are related to the need of a training phase, in which the traffic flow controller *learns* how the underlying system (i.e., the RAN and its users) behave and evolve in response to its control decisions.

Such control scheme proves to be a powerful tool when dealing with medium size scenarios, where the controller ability to learn how (statistically) service requests arrive/are terminated and their mean spatial distribution without any off-line analysis allows it to surpass standard approaches for AP selection (e.g., maximum receiving power or least-loaded selections).

In large scale scenarios (hundreds of users and dozens of APs) the proposed algorithm could archive reasonable performances provided that it employs Deep Q-Networks (DQN) [8] instead of tabular Q-Learning (note that the underlying Markov Decision Process modelling would be unaffected by the change).

In low scale scenarios, such as the one of the PoC where a single traffic flow is steered over two different APs (a terrestrial and a satellite one), RL approaches are suited for large scenarios where their training process is aimed at learning some complex statistical properties to approximate an optimal control behaviour. The algorithm could still be tailored to the storytelling of the project demonstration and the considered scenarios, but overall it may not be an ideal choice for the purposes of the demonstration.

On the contrary, the second algorithm (based on the concept of Wardrop equilibria), requires an off-line analysis that grows in complexity with the number of controlled traffic flows and access points. Once the analysis is completed and the needed quantities are estimated, the algorithm is very simple to implement as it requires just a few basic computations (mostly multiplications). Another advantage of the algorithm is that its asymptotic properties have been

formally proven. On small scale scenarios, as the one in the PoC, the off-line analysis becomes trivial and the only complexity becomes the design choices linked to the selection of an adequate *latency function* (see section 5.2 in the second issue of D4.2) for the use case and scenario objectives. We mention that this particular algorithm was also successfully preliminarily validated on the adaptive 8k video streaming solution developed in WP4 (presented in chapter 7).

Following the discussion above, the algorithms that is overall the better candidate to be demonstrated on the PoC is the Wardrop-based one. The customization process of this algorithm will be detailed in section 6.1.1.

We mention that the rest of the EU algorithms presented in D4.2 were not selected for the PoC due to the fact that they have been judged as less suitable for the PoC trialscenarios and they would have required remarkable changes to fully offer the functionalities needed for the demonstration use cases. In section 6.1.2 we report, for the convenience of the reader, the overview of the algorithm suite that was included in the second issue of D4.2 (where it was section 4.2), highlighting for each algorithm its suitability for being selected in the PoC.

### 6.1.1 Customisation of the selected algorithm for the scenarios of the PoC

With reference to the nomenclature used in section 5.2 in the second issue of D4.2, this section will describe how the Wardrop equilibrium-based control algorithm was tailored to the scenario considered in the PoC.

Let  $l_i(x_i)$  for  $i \in P = \{1,2\}$  be the latency functions of the two APs in the PoC and let  $x_i$  be the corresponding amount of traffic allocated on AP  $i$ . The total traffic request, denoted by  $\lambda$ , for the only *commodity*/controlled traffic flow/video stream is then  $\lambda = x_1 + x_2$ .

It was proven that the network, under a state feedback control law of the form

$$r_{i,j}(t) = \sigma \mu_{i,j}(t) x_i(t), \forall i, j \in P,$$

with a properly limited multiplicative gain  $\sigma$  (see 5.2.14 in the second issue of D4.2), and the Boolean decision variable  $\mu_{i,j}(t)$  indicating whether a traffic migration from AP  $i$  towards the AP  $j$  is allowed to take place, the network is steered towards a state where  $l_1(x_1) = l_2(x_2)$ . We mention that, according to the mathematical results reported in the second issue of D4.2, in the scenario of the PoC the gain  $\sigma$  can be set equal to large values and thus the convergence speed of the overall algorithm is expected to be fully compliant with the PoC specifications.

Selecting, for example,  $l_i(x_i(t)) = \frac{x_i(t)}{C_i(t)}$ , where  $C_i(t)$  is the IP-level capacity of the AP  $i$ , obtainable by the controller from the CRRM, at convergence the network would reach a state where: (i) the traffic demand is satisfied, as  $x_1(t) + x_2(t) = \lambda(t) \forall t$  and (ii) the percentage of used resources among the two AP would be equalised (i.e., the traffic would be allocated, dynamically, in a proportional way with respect to the available resources).

Note that several other choices can be made for latency functions. For example, a similar behaviour to the one above is observed when selecting the latencies to be equal to the number of resource blocks occupied over the APs (as in the tests, conducted on the opensource multi-connectivity simulator, reported in the second issue of D4.2); in the first issue of D4.2 we

reported a scenario in which we tested the transmission power consumed by the APs, obtaining at convergence a network state in which the transmission power was equalised over the various RATs.

The proposed latency function would meet the requirements for the scenario 1 “simultaneous cellular and satellite access with enhanced total throughput” of use case 2 from D2.4, and, thanks to the analysis of the continuous reports from the CRRM, would also automatically reconfigure the traffic flow after an unexpected failure on one of the two RATs (scenario 1 “unpredictable traffic switching” of use case 3). To address the second scenario “predictable traffic switching” from use case 3, the latency functions can also trivially be extended by considering an additional factor  $R_i(t)$  indicating the short-term risk of a failure in one of the channels:

$$l_i(x_i(t)) = \frac{x_i(t)R_i(t)}{C_i(t)},$$

so that, APs with a very high risk of failure level ( $R_i(t) \rightarrow +\infty$ , or an arbitrary max value) would be the less used ones (to return to the basic formulation, it is sufficient to assume  $R_i(t)=1$  if no risk of failure is predicted). Full details on the functioning of the traffic flow manager at TCP level are given in Section 7.1.5.

Regarding the QoE controller, given the video streaming nature of the traffic flow for multi-connectivity in the PoC we decided to employ an adaptive feedback-based algorithm called ELASTIC [10] (patent US20150146778A1 “Controlling Player Buffer and Video Encoder for Adaptive Video Streaming”) to maximize the User’s QoE by dynamically changing the video codec in order to minimize rebuffering events and providing the highest video bitrate. Full details on the algorithm functioning are given in Section 7.1.2.

Figure 6-2 reports the simulative results, visualized on a custom developed dashboard (detailed in section 7.1.8), of the proposed algorithms (including the Wardrop-based traffic flow controller, the adaptive video streaming provided by the QoE controller and the recombining action of the traffic flow manager).

From the image it is possible to see that the MPTCP quota of the two communication flows (green is terrestrial and yellow is satellite, both channels were emulated with a preliminary version of the channel emulators of the project) changes over time, in correspondence of the data coming from the cRRM. Due to the change of MPTCP quotas, also the bounds of the MPTCP congestion window change, together with the send data rate (the IP level capacity  $C_i(t)$ ).

It is possible to note that the video player (represented by the plots on the bottom) is still able to reach 8K video quality (level 6) despite the numerous changes in the sent data rate (the quality level 5 line is due to the video reloading phase, then video restarts at level 0 and it scales up to level 6, i.e., 8K).

More detailed testing are undergoing as the integration activities of WP5 continue, but combining the results obtained from the numerical opensource simulator for multi-connectivity [2] with the testing on a real data stream here reported we can conclude that the selected algorithm is capable of being deployed on the PoC.

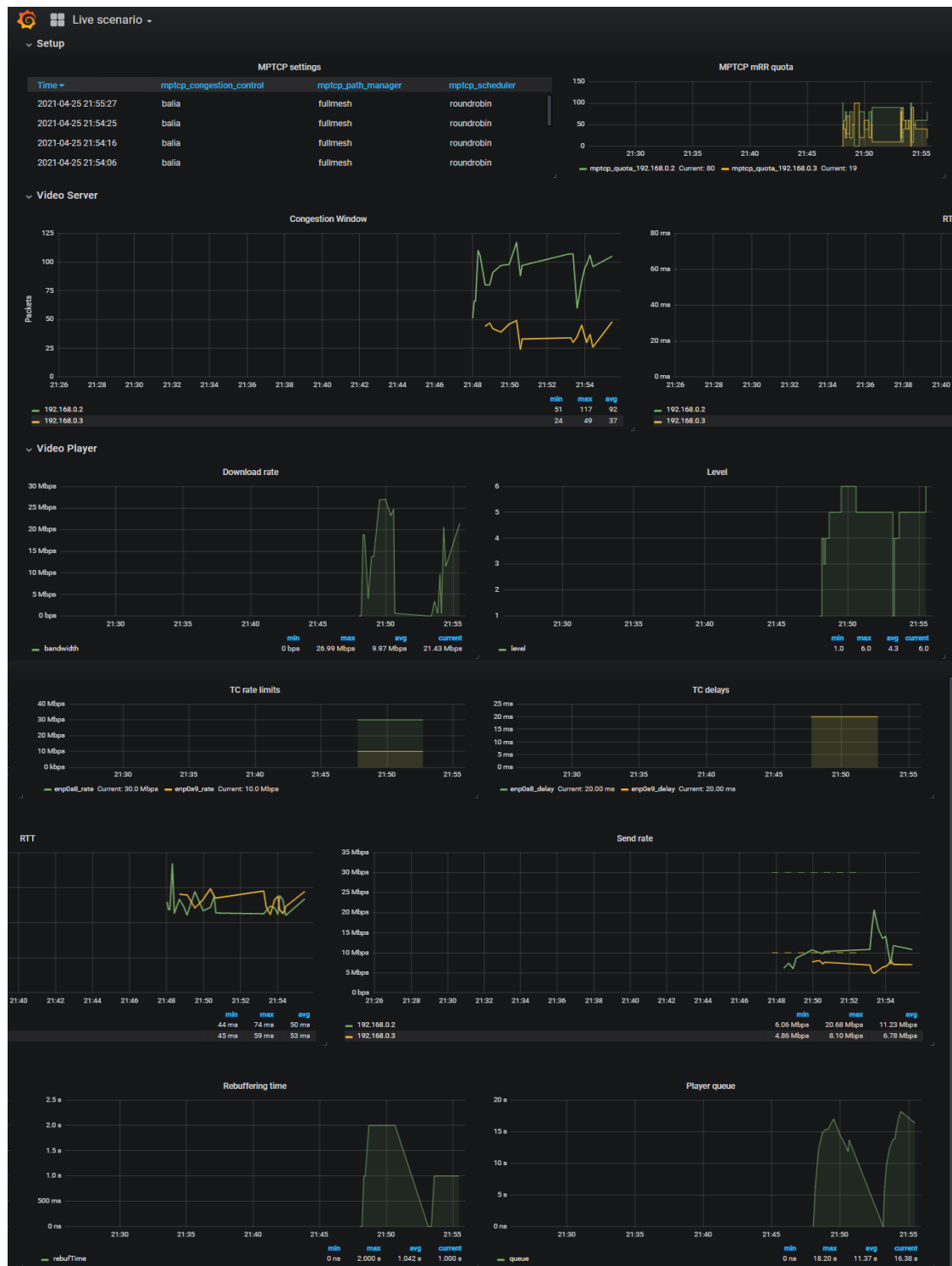


Figure 6-2 Algorithm testing on real video stream



### 6.1.2 Overview on the 5G-ALLSTAR suite of algorithms for multi-connectivity (from D4.2v2)

In this section we detail the main characteristics of each of the developed algorithm, as already reported in D4.2, along with its possible role in the project PoC, its level of testing (e.g., numerical validation, tested on the network simulator, tested on real data flows etc...) and its suitability for being selected in the PoC.

#### 6.1.2.1 Traffic steering and network selection in 5G networks based on reinforcement learning

**Main Characteristics:** Model free control scheme that has to be deployed in the C-RAN, as the controller requires knowledge on the entire state of the network (in terms of utilised resources).

**Control logic:** Network Selection -Every incoming request is allocated to an AP depending on the user/service profiles and the state of the network. Traffic Steering - The controller distributes the available network resources so that every connection receives a minimum amount determined by its minimum level QoS plus a share of resources that improves the QoE of the network users. Such allocation is dynamic (changes over time depending on the network state).

**Compliance with the 5G-ALLSTAR architecture:** Total, as the algorithm treats the AP as resource providers and governs how such resources shall be used. The algorithm assumes that the satellite resources can be used with no conflict with other C-RAN, thanks to an adequate slicing solution.

**Limits and cons:** being a model free solution it requires a training time, in which performances can not be assured. May be difficult to scale towards very broad C-RANs, as the training complexity increases with the number of AP (no problem for 10-20).

**Possibility to include QoE:** QoE is already included with different profiles depending on the service class (e.g., real-time, multi-codec video streaming, web browsing...) and user profile.

**Testing level:** validated on the open source network simulator and already produced a scientific publication .

**Likelihood of being integrated in the PoC:** Medium, as this algorithm is better suited for scenarios in which 5-10 APs are available to multiple UE, while in the PoC only two AP will be deployed. The algorithm has been tested, also in its Deep Reinforcement Learning variant, in the scope of T4.3 to evaluate its performances on both large-scale target scenarios (100+ UE) and scenarios more similar to the trial scenario of the PoC.

#### 6.1.2.2 Wardrop Equilibrium based control (now selected for final PoC)

**Main Characteristics:** Model based control scheme that can be distributed either in the UEs or in the APs (requires only partial knowledge on the system state). Strong theoretically proven convergence properties.

**Control logic:** Each UE steers (migrates) dynamically the traffic associated to the various QoS-flows over the APs it can connect to according to a mathematical control law. Following this law, every UE maximises the quality of its connection routing (according to user-defined preferences, such as QoE, cost, network resource usage,...) in an unilateral fashion (i.e., without cooperation among the UE or centralisation no better equilibrium state can be found).

**Compliance with the 5G-ALLSTAR architecture:** Total, as the algorithm treats the AP as resource providers and the control logic can be distributed in the UE or in the D-RANs that govern the various APs.



**Limits and cons:** The convergence speed of the algorithm slows as the number of UE grows (tested up to 50). Even in unfavourable network conditions (e.g., huge imbalance in initial routing, very scarce network resource availability), in our testing we stressed such unfavourable situations and still obtained reasonable convergence speeds.

**Possibility to include QoE:** QoE can be included as a performance metric on which the steering can be decided. This takes the name of “latency functions” in the literature related to Wardrop Equilibria.

**Testing level:** at the first issue of the D4.2 deliverable it was validated on a numerical simulator neglecting interference. The algorithm has now been tested (see D4.2v2) on the new network simulator that accounts for interference models and offers several other improvements compared to the previous study.

**Likelihood of being integrated in the PoC:** High, as this algorithm requires to function a very mild mathematical assumption (boundness of the derivative of the “latency function” selected to evaluate the traffic routing) and can be used in networks constituted even by hundreds of UE. The algorithm needs to be slightly simplified to cope with the reduced setting presented in PoC. A version of this algorithm, tailored to the PoC trial system, has been presented in section 6.1.1 of this document). The algorithm has also been tested on the real data flows generated by the adaptive 8k video streaming server developed in the scope of WP4 and described in chapter 7 of this document.

#### 6.1.2.3 Friend or Foe Q-Learning

**Main Characteristics:** Model free control scheme that has to be distributed at AP level.

**Control logic:** Each AP compete with the others to allocate the most lucrative connections to optimally exploit its network resources. This adversarial setting is typical of multi-tenant networks or in scenarios in which different user tariffs are present.

**Compliance with the 5G-ALLSTAR architecture:** Good, as the algorithm is distributed in the D-RAN controllers, requiring no cooperation among them. Only issue is related to the inclusion of the satellite, as it should be able to discern which C-RAN covers the area in which the UE is located and it may not be possible without cooperation with the ground segment or changes in the communication protocol.

**Limits and cons:** Being a model free solution it requires a training time, in which performances can not be assured. May be difficult to train in C-RANs with more than 10 APs.

**Possibility to include QoE:** QoE can be included with a re-design of the reward function, but the adversarial nature of the algorithm and the competition of the APs is probably not the best setting for QoE inclusion.

**Testing level:** Tested on a numerical simulator, produced a scientific publication.

**Likelihood of being integrated in the PoC:** Low, the case study considered for the PoC are not characterised by an adversarial setting and a significant redesign would be needed.

#### 6.1.2.4 Maximal Load Balancing (developed by the KR partners)

**Main Characteristics:** Model based solution for the optimal control of network resources by offloading traffic from overloaded APs to unloaded ones.

**Control logic:** The offloading of traffic from overloaded APs to less utilised ones happens on a threshold-based logic and takes into account the service type (e.g., is the connection delay tolerant for satellite routing?) and potentially the user preferences (e.g. battery usage).

**Compliance with the 5G-ALLSTAR architecture:** Total, as the algorithm can be distributed in the D-RAN controllers, with some level of cooperation (state information exchange) that is compliant with the existence of the C-RAN logical units or can be deployed in the centralised controller that oversees the whole RAN.

**Limits and cons:** The optimality of the resource usage has no clear mathematical backbone but empirically the improvements are significant. May require a good optimisation solution in scenarios with many APs.

**Possibility to include QoE:** This algorithm has been designed for network performances, but QoE may be used to determine how likely is for a connection to be offloaded on a satellite AP.

**Testing level:** Tested on a numerical simulator.

**Likelihood of being integrated in the PoC:** High, as the algorithm is very simple to implement in real settings and offers immediate benefits.

#### 6.1.2.5 Reinforcement learning resource optimization for mobile relays (developed by the KR partners)

**Main Characteristics:** Model free control solution to be deployed in UEs to optimise their frequency resource block and transmit power usage.

**Control logic:** Each UE learns the best strategy to use the available frequency block groups and transmit power on its own, by interacting with the environment. The goal is to have stable connection with good QoE profiles, while reducing interferences.

**Compliance with the 5G-ALLSTAR architecture:** Total, as the algorithm manages the up-link resources of the UE and requires some partial information on the network state from the various APs.

**Limits and cons:** The training time of this algorithm increases with the number of available APs, all the control decisions are taken by UEs (low computing devices may have difficulties with the training).

**Possibility to include QoE:** QoE is included in the so-called reward function, that guides the learning process of the UEs towards its maximisation.

**Testing level:** tested on a numerical simulator.

**Likelihood of being integrated in the PoC:** Low, as the PoC use cases don't focus particularly on the setting for which this algorithm was developed for.

#### 6.1.2.6 Analytic Hierarchy Process and Cooperative Differential Games for Multiconnectivity

**Main Characteristics:** Model based solution that decides an order (hierarchy) of priorities and objectives to maximise QoE.

**Control logic:** The APs of a C-RAN cooperate to maximise the QoE of their users by allocating the various connections on the available RATs depending on the (ordered) preferences of the users and their requirements in terms of QoS.

**Compliance with the 5G-ALLSTAR architecture:** Total, as the algorithm can be deployed in the C-RAN to oversee the functioning of the available APs and decide on the connection allocations.

**Limits and cons:** This algorithm requires a huge effort in identifying which are the user preferences, as the optimisation of the QoE and the satisfaction of the user requirements is the fundamental part of its optimisation. The questionnaire of the project may help in this direction but could still be too limited for a realistic application.

**Possibility to include QoE:** QoE optimisation is the core of the algorithm.

**Testing level:** preliminary tested on a numerical simulator, produced a scientific publication.

**Likelihood of being integrated in the PoC:** Low, as the QoE in the PoC will be demonstrated on a video stream by adapting its encoding and routing. Some aspects of this algorithm (e.g. connection preferences) may guide the tailoring of another algorithm to the PoC, as this algorithm is among the ones that better capture QoE related aspects.

#### 6.1.2.7 ZeroSum games and Minimax for Multi-Connectivity

**Main Characteristics:** Control solution for dual-connectivity (can be extended to multi-connectivity) between two AP. The AP compete to allocate the throughput requested to optimally use their radio resources.

**Control logic:** The two AP involved in the dual connectivity play a game to decide which AP routes the most traffic and consequently obtains the better return, while avoiding to overcommit their own resources.

**Compliance with the 5G-ALLSTAR architecture:** Total, as the control logic is distributed in the controllers of the two APs.

**Limits and cons:** The current formulation only takes into account dual connectivity but may be expanded to multi-connectivity. In the testing conducted so far the improvements are marginal.

**Possibility to include QoE:** QoE can be included in the optimisation process but in the current formulation the focus is on network resource usage.

**Testing level:** preliminarily tested on a numerical simulator.

**Likelihood of being integrated in the PoC:** Low, as this formulation was used as the basis to design more impactful algorithms based on Reinforcement Learning.

### 6.1.3 GAP analysis and comparison

The following table shows a GAP analysis of the proposed algorithms that will be used as a starting point for the activities of T4.3 regarding the selection of the algorithm(s) that will be implemented and tested in the PoC of the project and for the refinement of the algorithms themselves.

**Table 6-1: GAP analysis of the proposed algorithms for traffic flow control**

Section of D4.2v2	Methodology	Main Characteristics	Scalability	QoE integration
5.1	RL	Model Free, Optimal Network Selection based on feedback, C-RAN based	Limited but compliant with the typical number of AP available in an AN	Explicit, directly integrated with multiple services and their different QoE functions
5.2	Control Theory and Game Theory	Dynamic (in the sense that evolves over time), adversarial environment, feedback based	The performance may degrade with a high number of commodities	Explicit, as long as the latency functions are re-defined in terms of a QoE metric
5.3	Game Theory	Model Free, Optimal Network Selection based on feedback, D-RAN based in an adversarial framework	Limited but compliant with the typical number of AP available in an AN	A redesign of the reward function is needed, as at the moment the algorithm focuses on AP performance
5.4	Linear Programming	Model based, explicit optimisation, feedback-based	High, as the linear problem is characterised by low complexity	QoE is maximised by an optimal load balancing over the APs of the network resources
5.5	RL	Model free, feedback based, D2D	Compliant with the trial scenario, may benefit from ad-hoc solutions as state-space aggregation	Explicit, as it is directly included in the reward function
5.6	AHP, Game Theory	Model driven, hierarchical prioritization of objectives and performances	Limited in the preliminary formulation reported	explicit
5.7	Game Theory	Model driven, adversarial and distributed framework	Limited in the preliminary formulation reported	Explicit if included in the reward function

## 6.2 KR side Simulator Performance Evaluation

This section describes KR simulator's simulation results for performance evaluation of the main algorithm described in chapter 4.7.

### 6.2.1 Simulation Environment

We considered a 5G multi-RAT network including a satellite RAT and a 5G RAT. In the satellite RAT, a GEO satellite was connected to an NTN gNB through a ground station. The gNB was connected with a 5G CN that provided access to the public data network. There were seven 5G small cells deployed in a hexagonal pattern. A single satellite cell covered the whole terrestrial network. We considered 110 UEs in the network, and the required data rates for each UE were 5 Mbps to 15 Mbps. Regarding the UEs' distribution over the network area, UEs were randomly distributed among the cells. Half of them were static, and half were in random motion.

In the network, 70% of the UEs had delay-tolerant traffic, while the remaining UEs had delay-sensitive traffic. The UEs with delay-tolerant data flows had carrier spacing of 15KHz, and UEs with delay-sensitive data flows had either 15 KHz or 30 KHz carrier spacing. Transmission power was set to 46 dBm for 5G cells, and the bandwidth was 20 MHz. For the satellite, the C band was used for communications, and bands of frequencies from 3.7 to 4.2 GHz were used for downlink. The satellite had a channel bandwidth of 500 MHz and 12 transponders. Each transponder had a bandwidth of 36 MHz and a guard band of 4 MHz between adjacent transponders to avoid interference. The simulation parameters are summarized in Table 6-2.

**Table 6-2: Simulation parameter values**

Parameters	Values
Number of terrestrial cells	7
Tx power of terrestrial RAT	46 dBm
Terrestrial RAT bandwidth	20 MHz
Terrestrial path loss	$PL = 147.4 + 43.3\log_{10}(d)$
Satellite bandwidth	500 MHz (C band)
Satellite altitude	35780 Km
Number of transponders	12
Number of UEs	110
UEs data rates	5-15 Mbps
Initial threshold	75%

For the performance evaluation, we investigated the effect of the proposed algorithm on load distribution across the network and on network throughput. RRUR, which is defined in equation (4-3), was used to check load distribution among the cells. To validate the performance of the proposed algorithm, which is based on intra-RAT and inter-RAT load balancing, we compared it with an adaptive mobility load balancing algorithm [18]. Further scenarios with various numbers of UEs and cell bandwidths were simulated to show the effectiveness of the proposed algorithm. For the sake of simplicity, we denote the proposed mobility load balancing

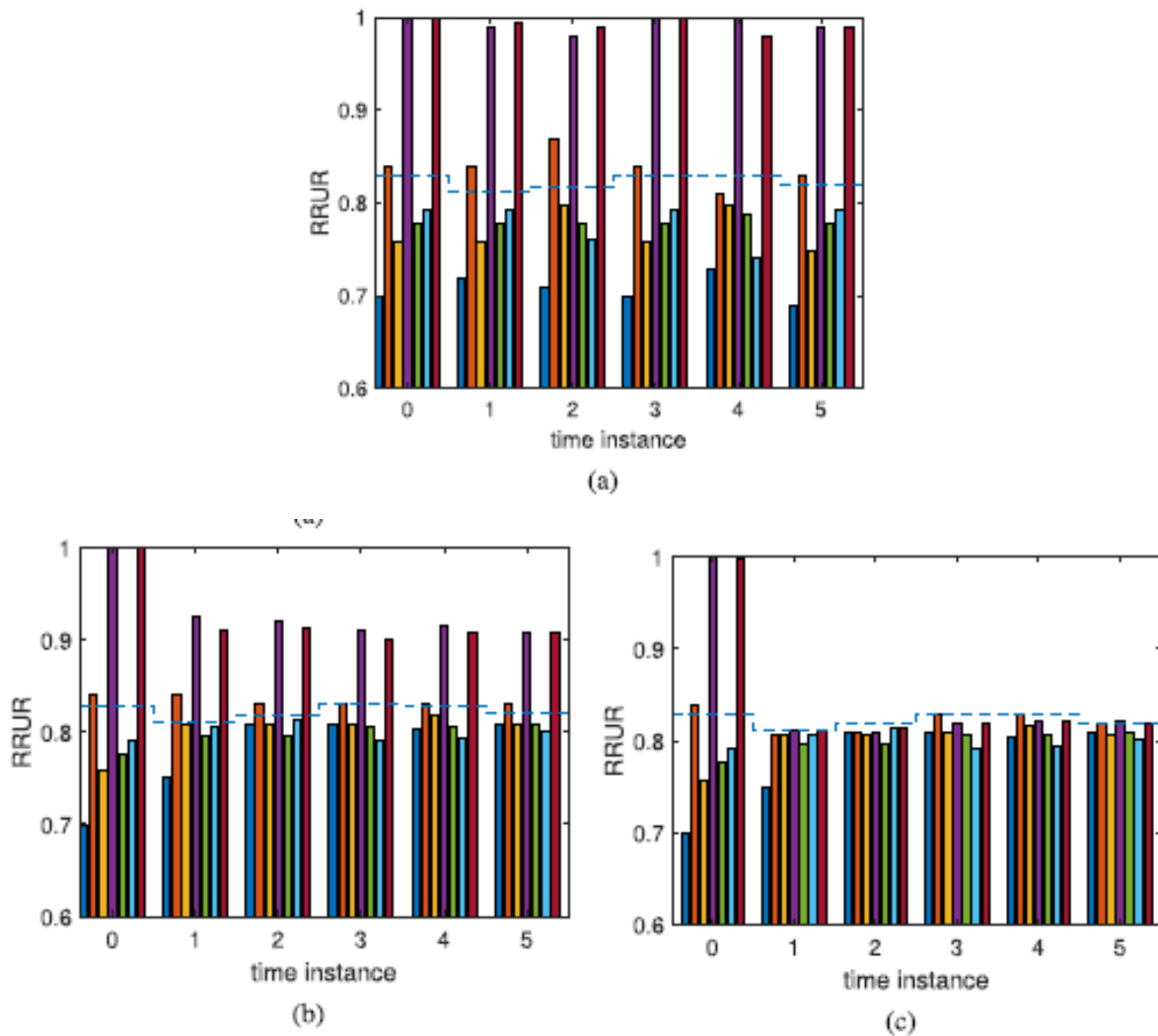
(MLB) algorithm as adaptive multi-RAT MLB, the adaptive mobility load balancing algorithm as adaptive intra-RAT MLB, and simulations without an MLB algorithm are denoted no MLB.

## 6.2.2 Impact of the Proposed Algorithm on Load Distribution

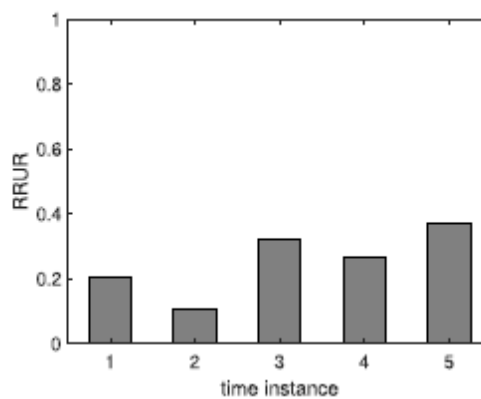
The algorithm's impact on load distribution across the network cells in terms of RRUR was compared with adaptive intra-RAT MLB and no MLB algorithms. The scenario with the initial setting was simulated without the MLB algorithm as well as with the MLB algorithms, and the RRUR of the terrestrial cells are shown in Figure 6-3. Each time instance shows the RRUR of seven 5G cells. Figure 6-3a shows the RRUR of the cells when no MLB was considered, and some terrestrial cell loads were more than the threshold, showing the cells were overloaded. The blue dotted line in each plot of Figure 6-3 shows the adaptive threshold, which changed with the network load. As we can see in Figure 6-3a, some cells had an RRUR greater than the threshold, i.e., 0.82, and some cells were underloaded, with an RRUR of less than 0.7. Consider time instance 2, cell 4 shows a maximum RRUR of 0.99, whereas cell 1 shows a minimum RRUR of 0.71, and the gap is 0.28. The RRURs of the cells with the adaptive intra-RAT MLB are shown in Figure 6-3b. As we can see in the figure, load from the overloaded cell moves to the underutilized cell to balance the network, and the gap between the maximum RRUR and the minimum RRUR was reduced to 0.10 in time instance 5. Although the adaptive intra-RAT MLB algorithm reduced the RRUR of the overloaded cells, cells had an RRUR greater than the threshold.

The RRURs of 5G cells were reduced to defined threshold under the adaptive multi-RAT MLB, as shown in Figure 6-3c. With the adaptive multi-RAT MLB, first the load from overloaded cells was released to underloaded neighboring cells, which increased the resource utilization of the underloaded that, the excess load from the overloaded cell, i.e., center UEs with a delay-tolerant data flow, moved to the satellite cell, which further reduced the load of the overloaded cells to the defined threshold. This eventually reduced the gap between the maximum RRUR and minimum RRUR until it reached 0.019. The RRUR of each terrestrial cell decreased to the threshold and the terrestrial cells network was evenly balanced under the adaptive multi-RAT MLB, as shown in Figure 6-3c. The satellite serves the UEs with delay-tolerant flows by keeping the RRUR at less than the threshold, which is shown in Figure 6-4. Considering the load status of the satellite, new users can easily be accommodated in the network and the satellite can assign more resources to satisfy the QoS of the users.

Figure 6-5 shows the standard deviation of 5G cell loads with and without load balancing algorithms. The standard deviation of the RRUR under the adaptive multi-RAT MLB algorithm is close to zero, and less than the adaptive intra-RAT MLB due the fact that the data flows of the center UEs in the overloaded cell can be offloaded to the satellite. Hence, the adaptive multi-RAT MLB performs load balancing considering 5G RAT and NTN RAT resources together and effectively released the load to balance the terrestrial network. Furthermore, the proposed algorithm considers the limitations of adaptive MLB as well as QoS of the UEs.



**Figure 6-3** RRUR of terrestrial cells in the network (a) without the MLB algorithm (b) with the adaptive intra-RAT MLB algorithm, and (c) with the adaptive multi-RAT MLB algorithm.



**Figure 6-4** RRUR of the satellite.



### 6.2.3 Impact of the MLB Algorithm on Network Throughput and QoS

The network performance in terms of average throughput and QoS of the UEs is shown in Figure 6-6. Without MLB, the resources of some cells were underutilized, whereas the UEs in overloaded cells could not have the required resources due to the scarcity of available resources. Therefore, the network had minimum throughput and fewer UEs getting the required resources, compared to load balancing algorithms. The adaptive intra-RAT MLB increased both the number of satisfied UEs and network throughput, but it was still less than the adaptive multi-RAT MLB, as shown in Figure 6-6. Considering the intra-RAT and inter-RAT offloading of UEs, the adaptive multi-RAT MLB allocated enough resources to all the UEs. More resources were available to UEs from multiple RATs that fulfilled the UEs' required data rates. The offloading of UEs from the overloaded cell to the neighboring cells, as well as to the satellite cell decreased the cell load and released more resources of the cells. This allowed the cells to allocate more resources to satisfy QoS of the UEs, and offloaded UEs got their required resources from the under loaded cells of different RATs, which satisfied the QoS of all UEs in the network, as shown in Figure 6-6a. These factors eventually led to an increase

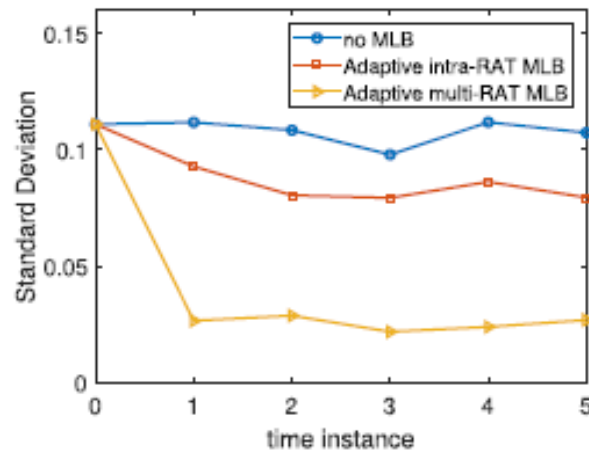


Figure 6-5 8. Standard deviation of RRUR among the cells of the 5G RAT

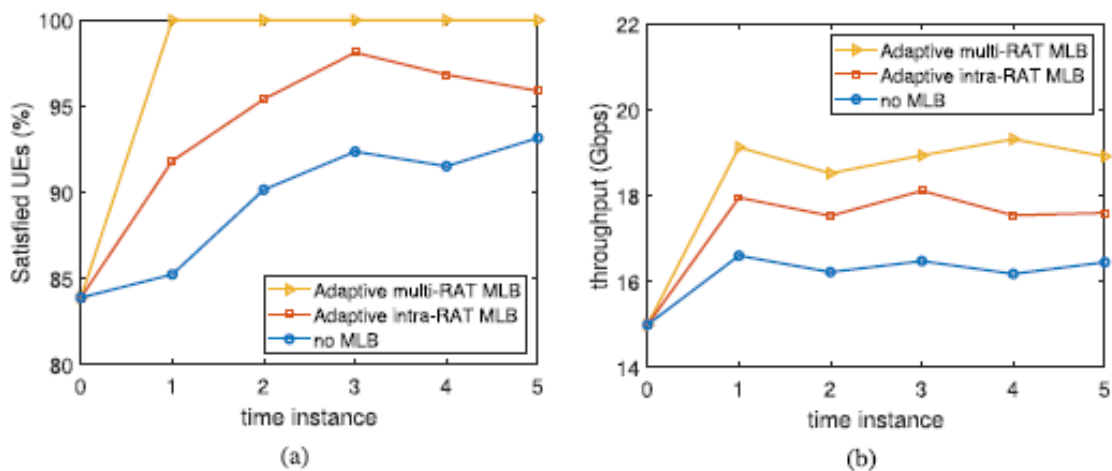


Figure 6-6 (a) The number of satisfied UEs in the network (b) Average throughput of the network.



in overall network throughput, as shown in Figure 6-6b. Thus, from Figure 6-3, Figure 6-5 and Figure 6-6, we can say that the adaptive multi-RAT MLB not only increased network capacity but also satisfied the QoS of the UEs Furthermore, the adaptive multi-RAT MLB balanced the terrestrial cells efficiently by keeping the RRURs of the cells of each RAT to less than the defined threshold.

#### 6.2.4 Impact of Various Numbers of Users

We studied the impact of various numbers of UEs in the network on the different approaches to load balancing. The network throughput and the standard deviation of the RRURs among terrestrial cells were observed. Network throughput increased under both of the MLB algorithms by increasing the number of UEs, as shown in Figure 6-7a. The adaptive multi-RAT MLB had more throughput as the resources of multiple RATs were efficiently utilized to satisfy the QoS flows of the UEs. However, the standard deviation in RRURs among terrestrial cells

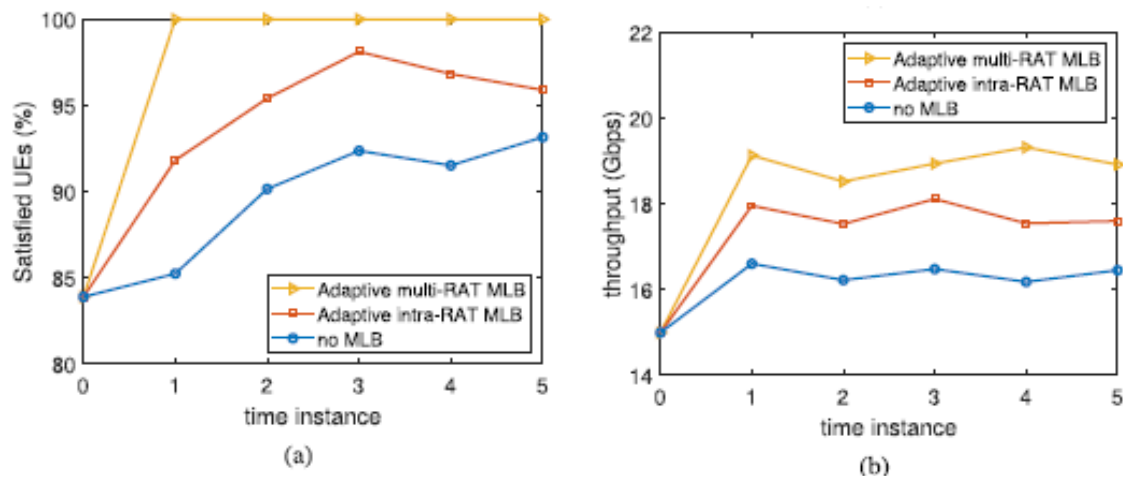


Figure 6-7 Performance of the load balancing algorithms with different numbers of UEs: (a) average throughput of the network, and (b) standard deviation of RRUR among the cells of 5G RAT.

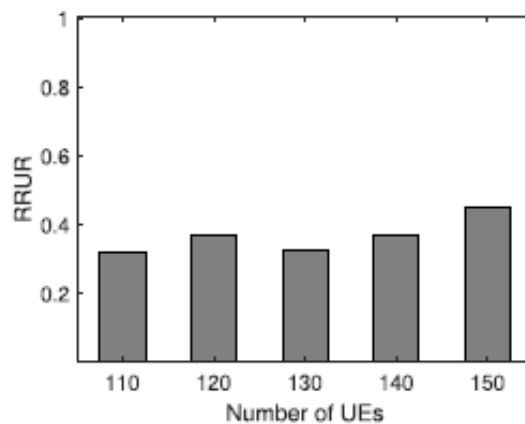
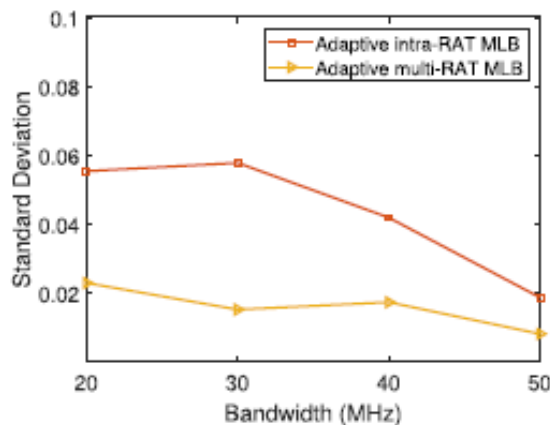


Figure 6-8 11. RRUR of the satellite cell.

increased with the increasing numbers of UEs, as shown in Figure 6-7b. The standard deviation of the RRUR increased by a very small amount under the adaptive multi-RAT MLB, and by less than the adaptive intra-RAT MLB. The gap between maximum RRUR and minimum RRUR increased more under the adaptive intra-RAT MLB, compared to the adaptive multi-RAT MLB with the increasing numbers of UEs. The adaptive multi-RAT MLB with intra-RAT and inter-RAT offloading transferred loads that cannot move to terrestrial neighboring cells to the satellite cell. The RRUR of the satellite is shown in Figure 6-8, and the utilized resources of the satellite were less than half of the available resources with large number of UEs in the network. Thus, the proposed algorithm keeps the network balanced with a large number of UEs, keeping the RRUR of the satellite minimal.

### 6.2.5 Impact of Different Channel Bandwidth

We changed the terrestrial cell bandwidth to observe the impact on the load balancing algorithms. The standard deviation of RRURs among terrestrial cells with different 5G cell bandwidth is shown in Figure 6-9. The standard deviation keeps decreasing when increasing the channel bandwidth in the adaptive intra-RAT MLB above the 30MHz bandwidth and came close to matching the adaptive multi-RAT MLB. The available resources were increasing in the 5G RAT when increasing the channel bandwidth, which reduced the gap between maximum RRUR and minimum RRUR. The network throughput increased with increasing bandwidths under both MLB algorithms. Network throughput under the adaptive intra-RAT MLB increases more rapidly, compared to the adaptive multi-RAT MLB, as shown in Figure 6-11. However, the adaptive multi-RAT MLB had more throughput because racecourses of multiple RATs were available to more UEs at the same time. Hence, the proposed algorithm was able to achieve more even load balancing and increased the capacity of the network at the same time.



**Figure 6-9 Standard deviation of RRURs among the terrestrial cells with varied 5G cell bandwidths.**

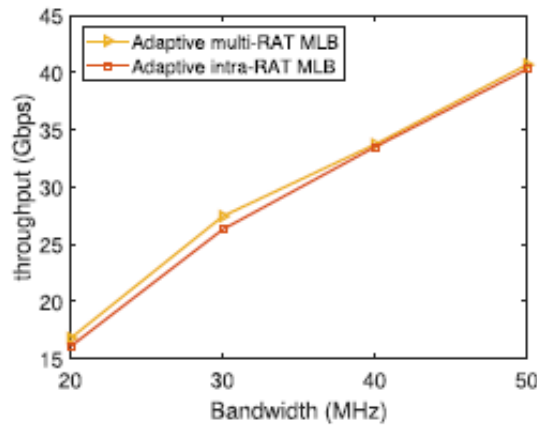


Figure 6-10 Average throughput of the network with varied terrestrial RAT bandwidths.

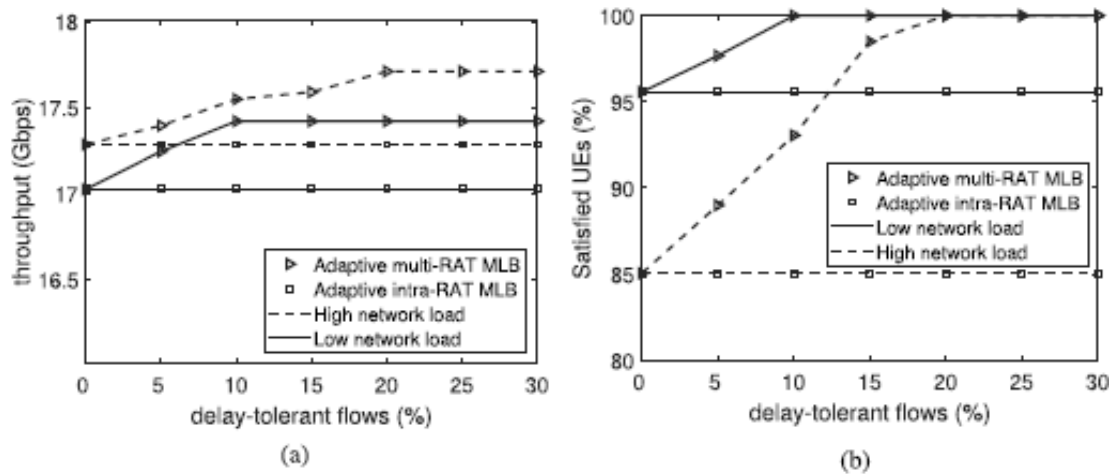


Figure 6-11 (a) Average throughput of the network, and (b) the number of satisfied UEs in the network with different delay-tolerant and delay-sensitive traffic ratios.

### 6.2.6 Impact of Delay-Tolerant Flows with Different Network Load

We varied the delay-tolerant flow ratio in the network to observe the impact on the proposed algorithm. For a given total number of UEs, the percentage of UEs with delay-tolerant flows was changed from 0 to 30 for different network loads. For the different network load, we changed the required data rate of each UE. The required data rates for each UE were 5-10 Mbps and 10-15 Mbps for low and high network load, respectively. Figures 14a and 14b show the network throughput and the number of satisfied UEs, respectively, for different delay-tolerant flow ratios under different network loads. The adaptive multi-RAT MLB has better performance than the adaptive intra-RAT MLB when there are UEs with delay-tolerant flows in the network. When there is no delay-tolerant traffic, i.e., all UEs have delay-sensitive flows, the adaptive multi-RAT MLB only performs intra-RAT offloading. So, the performance of the adaptive multi-RAT MLB returns to the adaptive intra-RAT MLB when there is no UE with a delay-tolerant flow for inter-RAT offloading. The performance of the adaptive intra-RAT MLB

remains constant for different delay-tolerant and delay-sensitive ratios, as shown in Figure 14. The reason is that the adaptive intra-RAT MLB performs terrestrial to terrestrial offloading of the UEs irrespective of the data flow type to balance cell loads, whereas the performance of the adaptive multi-RAT MLB increases with increases in delay-tolerant traffic. By increasing delay-tolerant traffic, the adaptive multi-RAT MLB finds more UEs with delay-tolerant flows and offloads the UEs from overloaded cells to a satellite to balance the network. As a result, more UEs get the required resources from multiple RATs, and the network throughput and percentage of satisfied UEs increases. After a required minimum number of delay-tolerant flows, the network throughput and number of satisfied UEs become constant under the adaptive multi-RAT MLB under different network load conditions. When the network load is high, the adaptive multi-RAT MLB requires a higher ratio of delay-tolerant flows to balance the terrestrial cells. Hence, we can say that the adaptive multi-RAT MLB depends on the availability of delay-tolerant flows for inter-RAT offloading to achieve better performance.

## 7 Description of developed SW components

### 7.1 SW Components of EU side

The following sections of this chapter detail, from a functional and architectural point of view, the software components developed by the EU partners related to multi-connectivity algorithms to further validate the multi-connectivity solutions developed which will be used in the framework of the project PoC. The software here described will then be integrated with the rest of the components developed (and in particular the channel emulators) in the project in the scope of the activities of WP5, for the overall project demonstration.

In fact, the tool here developed and reported will be responsible in the PoC for: 1) the data flow generation (for the scenarios involving video streams); 2) the multi-connectivity traffic steering/control capabilities of the EU PoC; 3) The logical reconstruction of the split traffic flows to enable a seamless multi-connectivity solution.

The chapter starts with the description of the custom video server architecture and then details the adaptive, QoE-aware, video streaming solution that aims at maximising the quality of the video and the user QoE with a feedback control law. The chapter then briefly describes the testing performed to validate the video platform functioning and details the custom multi-path TCP implementation that enables the division and reconstruction of the video QoS-flows. Full details on the graphical interface of the tool and its HTTP proxy are also given.

#### 7.1.1 The Video Server architecture

The Video Server component is a web service for streaming video contents at 8K resolution (7680x4320 pixels) using a Video-on-Demand (VoD) distribution. The service is fully compliant to the HTTP Adaptive Streaming (HAS) architecture; in particular, it implements both the *Dynamic Adaptive Streaming over HTTP* (DASH) ISO standard and the *Apple HTTP Live Streaming* (HLS) standard.

The Video Server architecture is composed of these components (Figure 7-1):

- **API service:** it implements the core logic of the system (the control layer), exposing the REST API used for communicating with the other components; it is implemented as a NodeJS microservice;
- **Database:** the persistent storage of structured data (users, streams, logs, etc.); it uses the MongoDB database;
- **Key-value store:** a volatile storage for caching intermediate values; it uses the Redis server;
- **Web interface:** it implements all the web frontend graphic and logic components, including the Adaptive Video Player; the implementation is based on Angular framework;
- **Adaptive Video Player:** the video player which implements an adaptive streaming algorithm that will support resolutions up to 8K, selecting automatically the most suitable video representation bitrate to play, based on the actual client network bandwidth and the video screen available resolution;
- **VOD encoder:** it performs the video encoding of the uploaded contents, using the configuration presets sent by the API service; the content is encoded using multiple synchronized file versions at increasing resolution and bandwidth; the video encoder outputs are encoded using H.264 video codec and AAC video codec, packaged into MP4 format;
- **VOD module:** it implements the DASH and HLS stream packagers used for delivering the video contents playable by the Adaptive Video player;
- **HTTP proxy:** it is the web proxy for all the services exposed by the Video Server architecture to the external; it uses the Nginx http server.



In a HTTP streaming system, the control algorithm used for selecting the video level to be downloaded is placed at the client side, running directly in the browser web page as a JavaScript executable that uses the Media Source Extensions<sup>2</sup> for parsing, decoding and displaying the video contents. At the server side, each video is encoded in multiple versions (video levels  $L = l_0, l_1, \dots, l_N$ ) with different resolutions and bitrates, ranging typically from 240p to 1080p; each video is composed by a sequence of chunks that are time-aligned between the versions, with a chunk duration ranging from 2 to 10 seconds; each chunk contains one or more Group

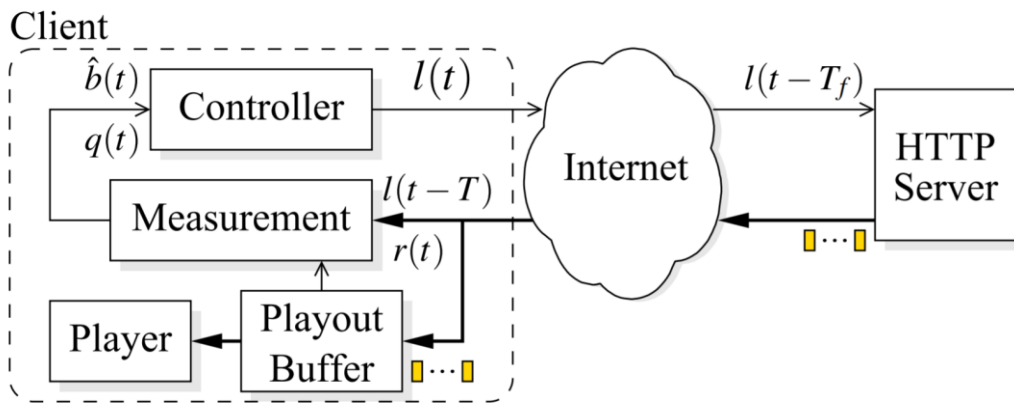
---

61

of Pictures (GoP), that are video frames that can be independently decoded, thus allowing the client decoder to switch from different versions in real-time.

The control algorithm goal is selecting a video level  $l(t)$  to achieve the maximum Quality of Experience (QoE). The main factor that improves the QoE is the received video level, that should be the highest possible [13] [14]. The key factors that decrease the user engagement are described in [12] and [11]: the number of rebuffering and the number of quality switches are the more important aspects that should be minimized in order to obtain an acceptable user QoE. Summarizing, the control algorithm design goals are:

1. obtain the maximum video quality level;
2. minimize the rebuffering events;
3. minimize the level switch events;
4. equally share the network bottleneck with concurrent video streams and TCP flows.



**Figure 7-2 The ELASTIC controller block diagram.**

The ELASTIC controller block diagram is shown in Figure 7-2: the controller calculates the video level  $l(t)$  to be downloaded and it requests the next video fragment to the server; the server replies after a total time of  $T$  with the video content that is put on a playout buffer (modeled as a queue). As the HTTP application protocol is based on the TCP transport protocol, for each downloaded chunk we calculate an end-to-end bandwidth  $b(t)$  and a round-trip-time (RTT)  $T$ . The client receives the video chunks at a rate  $r(t) < b(t)$ , and stores them in a playout buffer that feeds the video decoder. The controller (running in the web page), when a new chunk is downloaded using a HTTP GET method, dynamically decides the next video level  $l(t)$  to be downloaded. The controller, given the chunks sizes and the download times, measures the estimated bandwidth  $b(t)$ ; the actual playout buffer level  $q(t)$  expressed in seconds is provided by the HTML <video> element.

The playout buffer can be modelled with the following nonlinear differential equation:

$$\dot{q}(t) = \frac{r(t)}{l(t)} - d(t) \quad (7.1)$$

The goal is to design a control law  $l(t)$  such that  $q(t)$  is steered to the set  $[q_L, q_H]$  avoiding inserting OFF phases (i.e., timeouts between the download of consecutive chunks) to control  $q(t)$  and achieving full link utilization.

The ELASTIC control algorithm has three different states, according to the current playout buffer length:

1. Steady-state zone when  $q_L < q(t) < q_H$ : the video level  $l(t)$  is kept unchanged;



2. Under buffering zone when  $q(t) \leq q_L$ : the video level  $l(t)$  is changed according to the control law that will be described in the following;
3. Over buffering zone when  $q(t) \geq q_H$ : it means that we have enough buffer and the video level should not be decreased in this zone. Towards this end we employ the same control law used in the “Under buffering zone” but we inhibit the video level to decrease.

The variables involved in the control loop are the following: the video level  $l(t)$  is the control variable,  $q(t)$  is the output of the controlled system, whereas  $r(t)$  (i.e., the received rate) can be modelled as a disturbance. It is important to notice that  $l(t)$  can only assume values in the discrete set  $L$ , meaning that the output of the controller cannot vary with continuity.

In the following we employ the feedback linearization technique [9] to compute a control law that linearizes (7.1) and that aims at steering  $q(t)$  to the set  $[q_L, q_H]$ . Towards this end, we impose the following linear closed-loop dynamics for the queue:

$$\begin{aligned}\dot{q}(t) &= -k_p q(t) - k_i q_I(t) \\ \dot{q}_I(t) &= q(t) - q_T\end{aligned}\tag{7.2}$$

where  $q_T = (q_L + q_H)/2$ ,  $q_I$  is an additional state that holds the integral error,  $k_p \in R_+$  and  $k_i \in R_+$  are the two parameters of the controller.

Now, by equating the right-hand sides of (7.1) and (7.2), it turns out:

$$l(t) = \frac{r(t)}{1 - k_p q(t) - k_i q_I(t)}\tag{7.3}$$

Analysing the equilibrium of (7.2), if we impose the derivatives of (7.2) to be zero we obtain  $q = q_T$  and  $q_I = -(k_p/k_i)q_T$ . If we plug the equilibrium of (7.2) in (7.3) we obtain a steady state video level that exactly matches the end-to-end measured bandwidth  $r$ , i.e., at steady-state  $l = r$ . However, since the video level cannot be chosen with continuity and has to be quantized to a value belonging to the discrete set  $L$ , instead of having a constant steady-state, we will have a triangular wave trend once the system gets inside the set  $[q_L, q_H]$ .

The control law (7.3) will be used in the zones “Over Buffer zone” and “Under buffer zone”. In particular, the control law (7.3) needs to be discretized and then quantized in order to be implemented



---

```
1: On segment download:
2:  $\Delta T \leftarrow \text{getDownloadTime}()$ 
3:  $S \leftarrow \text{getSegmentSize}()$ 
4:  $q \leftarrow \text{getQueueLength}()$ 
5:  $r \leftarrow h(S/\Delta T)$ 
6:  $q_I \leftarrow q_I + \Delta T \cdot (q - q_T)$ 
7: return  $\text{Quantize}(r/(1 - k_p q - k_i q_I))$ 
```

---

**Algorithm 7.1:** The ELASTIC controller pseudo-code.

Algorithm 7.1 shows how to implement (7.3): upon a segment is downloaded, the following quantities are measured:

1. the time spent to download the segment  $\Delta T$  (line 2);
2. the last downloaded segment size  $S$  in bytes (line 3);
3. the playout buffer length (line 4);

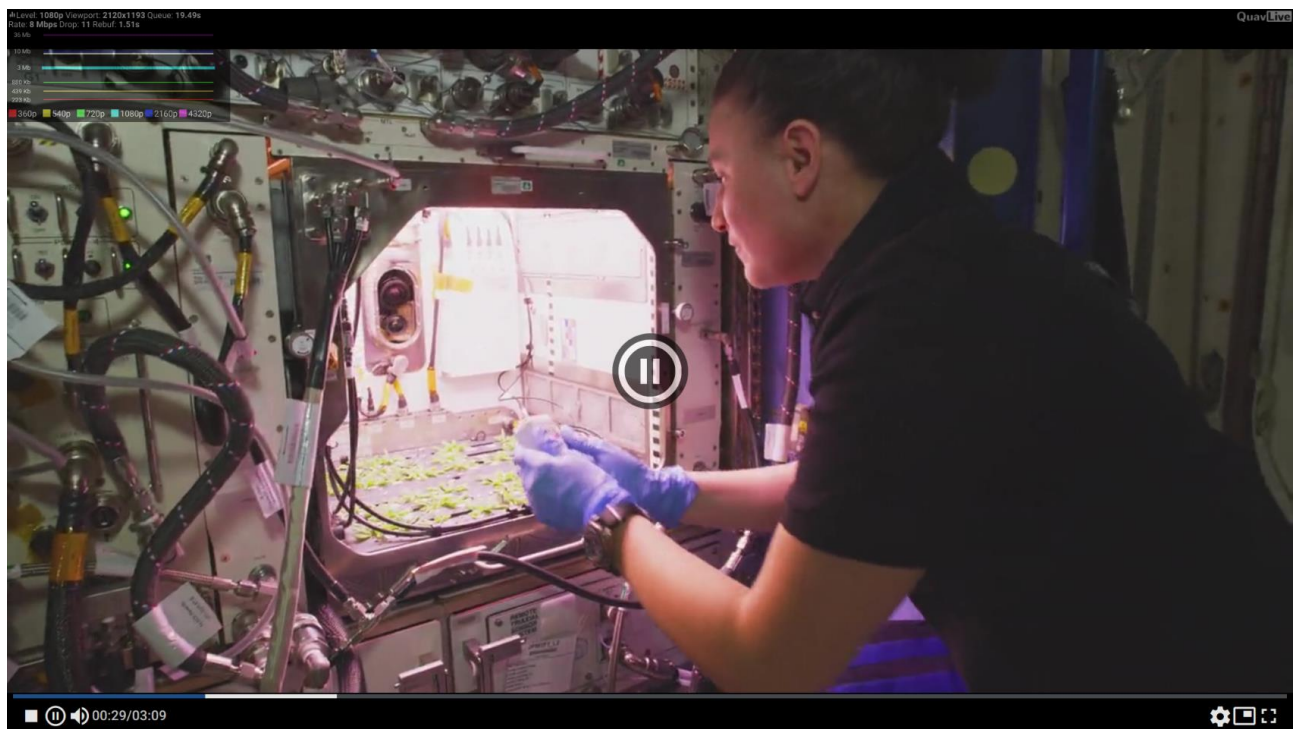
The received rate  $r$  is estimated by passing the last segment download rate  $S/\Delta T$  through a harmonic filter  $h(\cdot)$  over the last 5 samples of  $r$  (line 5). Then, the integral error  $q_I$  is updated (line 6) and the control law is computed using (7.3) (line 7).

### 7.1.3 The encoding settings

The VOD Encoder module has been configured to encode the 8K video streams using the configuration described in Table 2.3. The test video sequence used in the experimentations is “*First-8K-Video-from-Space*” from NASA. The DASH/HLS chunk duration is fixed to 3.2 seconds and the player buffering target is set to 9.6 seconds. The video codec used is H.264 (x264 software encoder with preset=medium and tune=film), while the audio codec is AAC (2 channels audio using a bitrate of 128 kbps).

**Table 7-1 The encoding settings.**

Level	Resolution	Framerate	Bitrate	Profile
L1 (QHD)	640x360	15	500 kbps	baseline
L2 (QFHD)	960x540	25	1 Mbps	main
L3 (HD)	1280x720	25	2 Mbps	high
L4 (FHD)	1920x1080	25	6 Mbps	high
L5 (4K)	3840x2160	25	20 Mbps	high
L6 (8K)	7680x4320	25	30 Mbps	high



**Figure 7-3 : Video Player screenshot.**

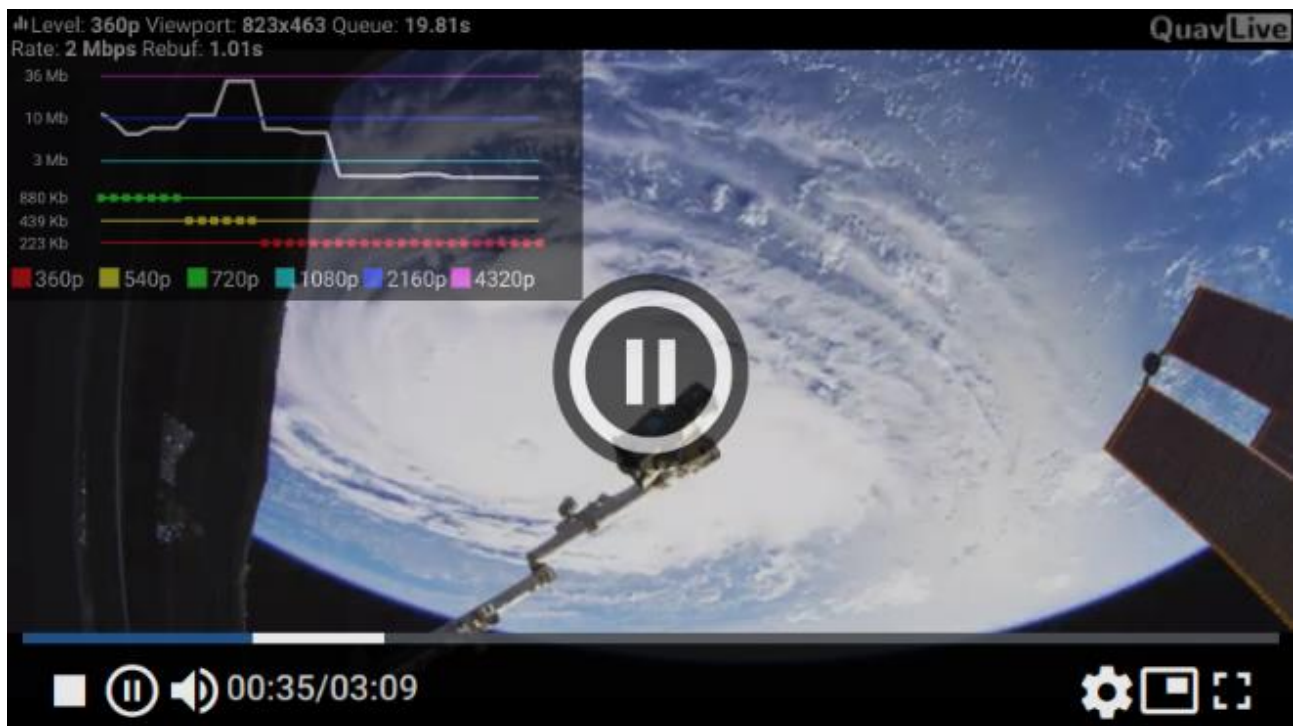


Figure 7-4 Video Player screenshot with quality levels detail.

#### 7.1.4 Testing setup and tools

In order to perform automated experiments inside the testbed platform, we developed the Video Player Test Client (Figure 7-5). This software is based on the Selenium library<sup>3</sup> and it allows to run an instance of a web browser (Google Chrome, Chromium, Firefox) in headless mode (i.e. without graphical interface rendering) and controllable by an automated script. It is worth noting that, while running in headless mode, the browser functionalities are kept the same of running in normal mode. Inside this controlled environment, the browser is redirected to the test web server HTTP address, using a configuration that allows to automatically start the player session. Since the adaptation algorithm takes into account also the browser window resolution to cap the maximum video rate selected, we set the browser window resolution to 7680x4320 pixels, hiding all the interface controls.

<sup>3</sup> <https://www.selenium.dev/>

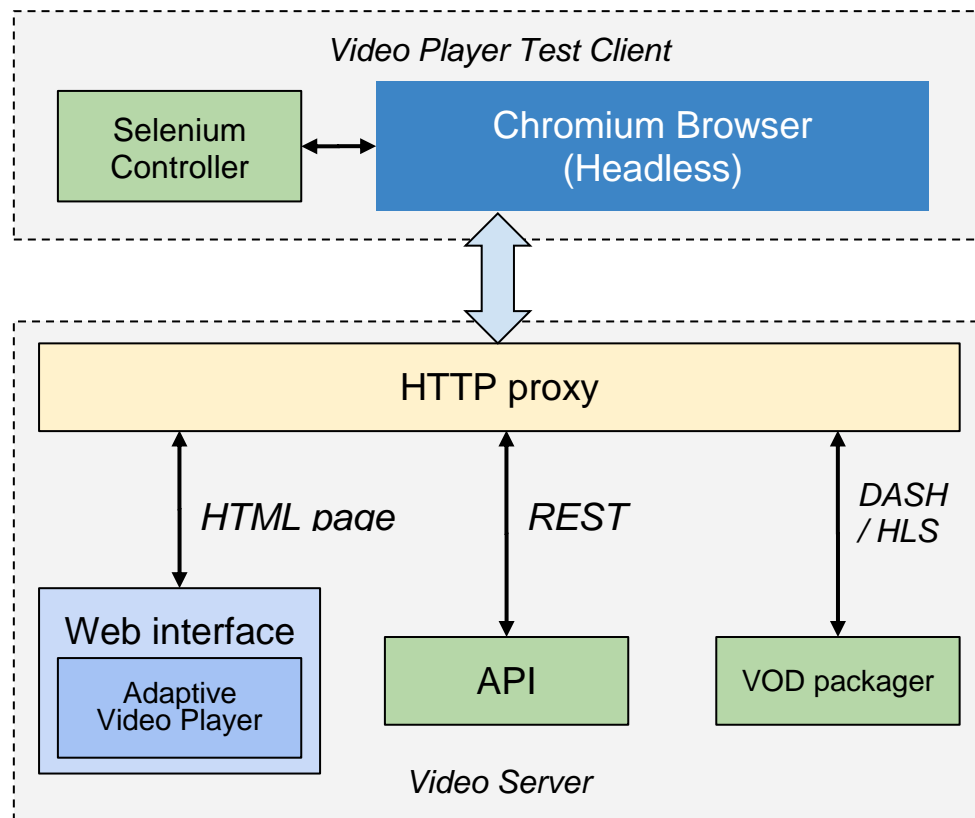


Figure 7-5 The testing architecture.

The browser used for the experimentation is Chromium, started with the following command line switches:

- **--headless**: run without rendering the graphical user interface
- **--disable-gpu**: disable rendering using GPU
- **--kiosk**: removes status bars and other interface elements;
- **--ignore-certificate-errors**: allow to visit self-signed certificates HTTPS pages
- **--no-user-gesture-required, --autoplay-policy=no-user-gesture-required**: allow to start the playing without any user interaction
- **--window-size=width,height**: the window size applied to the browser
- **--disable-application-cache**: disable the application content caching; in order to start without any data from the previous test runs, the `~/.cache/chromium` directory is removed before any test run.

The server accepts the client requests sent via HTTP with this scheme:

`https://<server_ip>/embed/<stream_id>?autoplay=1`

Where:

- **<server\_ip>** is the server IP address visible to the client;
- **<stream\_id>** is the VOD stream unique identifier;
- **autoplay=1** allows the playing to be started automatically when the page is loaded.

Inside the controlled environment, the tool is able to capture from the running browser instance some QoE metrics about the current playing session, as described in Table 7-2.

**Table 7-2 The Video Player QoE metrics.**

Metric name	Description
<b>t</b>	The current playing time in seconds
<b>buffer</b>	The player queued data expressed in seconds
<b>totalVideoFrames</b>	The total number of video frames received
<b>droppedVideoFrames</b>	The total number of video frames dropped by the player if the client CPU is not capable of decoding the video in real time
<b>index</b>	The video chunk fragment index requested
<b>level</b>	The video quality level currently selected in the range L0-L5
<b>rebuf</b>	The number of rebuffering events, including the startup buffering event
<b>rebufTime</b>	The total time elapsed in rebuffering states in seconds, including the startup buffering event
<b>screenWidth, screenHeight</b>	The client browser window size
<b>bandwidth</b>	The average download rate in Mbps
<b>paused, ended</b>	If the player state is “paused” or “ended”
<b>bytesSent</b>	The total bytes downloaded by the client

At the server side, a set of metrics related to the opened TCP connections are described on Table 7-3. Each metric has an associated **suffix** value, that is the destination IP of each TCP connection. The metrics are obtained using the **iproute2** tools [15] and in particular, using the **ss** command<sup>4</sup>.

**Table 7-3 The server side metrics.**

Metric name (for each dest IP)	Description
<b>rtt, rtt_dev, rtt_min</b>	The Round Trip Time current value, the standard deviation and the minimum value
<b>cwnd</b>	The TCP congestion window size
<b>bytes_sent</b>	The total bytes sent
<b>bytes_retrans</b>	The total bytes retransmitted
<b>bytes_received</b>	The total bytes received
<b>send_bitrate</b>	The send bitrate
<b>delivery_bitrate</b>	The delivery bitrate

<sup>4</sup> <https://man7.org/linux/man-pages/man8/ss.8.html>

### 7.1.5 Custom Multipath TCP

In the PoC architecture developed in WP5, the functionality of reconstructing the original video stream from the multipath flows was delegated to a functional module named “Traffic Flow Manager”. In order to develop such functionality, several issues had to be addressed, and the content of this section reports the design choices that allow the video streaming service to utilise multi-connectivity.

The TCP/IP classical communication uses a single path connection between two peers. When multiple paths exist, using them would improve the throughput allowing to transfer contents in less time and to switch in a seamless way between one path to another without interrupting the communication (Figure 7-6). The mechanism that allows to simultaneously use multiple paths between peers is called Multipath TCP (MPTCP) and it is an open standard documented in [29]. The MPTCP implementation for the Linux kernel<sup>5</sup> has currently reached the version 0.95, that is the version used in the server deployment.

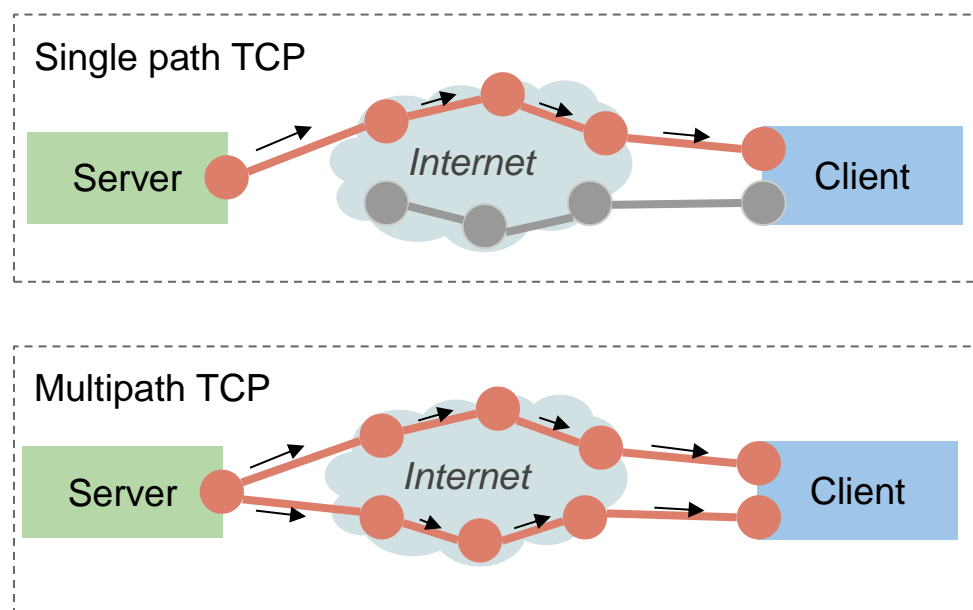


Figure 7-6 Single path vs Multipath TCP.

The MPTCP protocol is implemented in Kernel mode and extends the standard TCP transport layer (Figure 7-7) adding a control layer between multiple TCP connections and the socket interface; these TCP connections (called **subflows**) are established over multiple network links, thus allowing to demuxing the data flow sent from user space to multiple paths. At the application layer, no modifications are required in order to use MPTCP, as the protocol offers the same type of service as TCP (reliable, byte stream oriented connections) [30]. Moreover, MPTCP-handled TCP flows can co-exist with other regular TCP connections established on the same path.

<sup>5</sup> <https://www.multipath-tcp.org/>

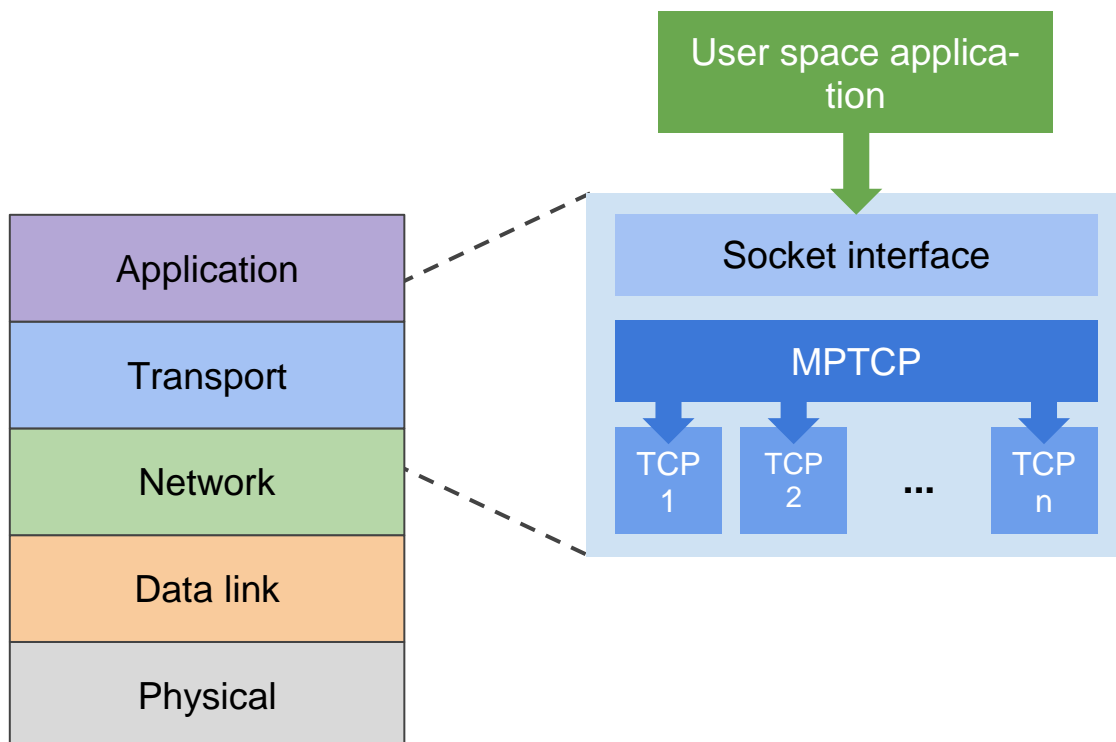


Figure 7-7 Multipath TCP protocol stack.

MPTCP adopts various algorithms for scheduling the transmission of packets over different subflows. The path manager allows to decide how to use the available paths, in particular in the Linux kernel implementation we have these choices:

- **default**: the path manager works in passive mode, i.e. the host won't initiate the creation of new subflows, but it will accept the creation of new subflows from other hosts;
- **fullmesh**: a full-mesh of subflows among all available peers is created;
- **ndiffports**: this will create N subflows across the same pair of IP addresses, modifying the connection source port;
- **binder**: it uses the algorithm described in [32].

The scheduler regulates how the traffic should be distributed among different connections [31]. The Linux implementation allows to choose between these algorithms:

- **Default**: the data is sent on subflows with the lowest RTT until their congestion window is full; then, it will start transmitting on the subflows with the next higher RTT;
- **Round-Robin (RR)**: it transmits traffic in round-robin over all the available subflows; it can be tuned using these sysctl parameters:
  - *"num\_segments"*: the number of consecutive segments to be sent at each round;
  - *"cwnd\_limited"*: if true, if the scheduler tries to fill the congestion window on all subflows; if false, it prefers to leave open space in the congestion window to achieve real round-robin (even if the subflows have very different capacities);
- **Redundant**: the traffic is transmitted on all available subflows in a redundant way; with this approach the transmission latency is minimized, at the expense of no throughput increase than using a single path TCP connection.

Various congestion control algorithms have been proposed to improve the performance of TCP flows over MPTCP [33]. The algorithms currently implemented in the Linux kernel are:



- **LIA** (Coupled Congestion Control for Multipath Transport Protocols) [34]
- **OLIA** (Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP) [35]
- **WVEGAS** (Delay-based Congestion Control for MPTCP) [36]
- **BALIA** (Balanced Linked Adaptation Congestion Control Algorithm for MPTCP) [37][38]

LIA, OLIA and BALIA are an extension of TCP Reno congestion control. The RFC 6356 [34] defines the following requirements for the MPTCP congestion control:

1. MPTCP performance should be at least equivalent to a single available TCP subflow;
2. The combined traffic of all MPTCP subflows should not take more capacity than a single TCP connection;
3. Each individual subflow should use a congestion control that is dependent on the congestion of the single path.

The Lia congestion control algorithm allows to couple the additive increase function of each subflow, without modifying the decreasing phase. As stated in [35], using the LIA congestion control, MPTCP flows may be excessively aggressive toward concurrent TCP flows; in order to resolve these problems, the OLIA congestion control was proposed. BALIA is another congestion control [37] that focuses on balancing the friendliness and the responsiveness. The weighted Vegas (wVegas) [36] is another congestion control algorithm which uses packet queuing delay as congestion signals, thus achieving fine-grained load balancing between subflows.

#### 7.1.5.1 The modified Round-Robin (mRR) scheduler

To the purpose of proving a Multipath implementation allowing to decide on user space the quota of data to be sent over multiple connections (Satellite and 5G), a modified version of the Round-Robin scheduler has been implemented.

In the Round-Robin scheduler implementation, when a data segment is queued from the user space, the scheduler<sup>6</sup> decides which subflow should be used for data transmission assigning to each subflow a quota value starting from 0, incremented at each send iteration until it reaches the “num\_segments” value; when the “num\_segments” value is reached, the scheduler switches to the next available subflow.

In the mRR implementation, the “num\_segments” value can be customized assigning to a particular IP destination a custom value; mRR uses a “quota\_table” array where, for each active TCP subflow, the following values are stored:

- **daddr**: the subflow destination IP address;
- **num\_segments**: the maximum quota value to use for scheduling over the subflow;
- **count**: the total number of segments sent over the subflow.

The “quota\_table” array is updated at each iteration and the settings are exposed to the user-space virtual file “/sys/kernel/mptcp\_rr/config”. Reading from “/sys/kernel/mptcp\_rr/config” allows to obtain a list of the latest active subflows with the number of segments sent; writing to “/sys/kernel/mptcp\_rr/config” changes the “num\_segments” value for the selected destination IP address.

Usage example:

```
# Usage: echo "daddr=quota" > /sys/kernel/mptcp_rr/config
> echo "33597632=80" > /sys/kernel/mptcp_rr/config
> echo "50374848=20" > /sys/kernel/mptcp_rr/config
```

<sup>6</sup>[https://github.com/multipath-tcp/mptcp/blob/mptcp\\_v0.95/net/mptcp/mptcp\\_rr.c#L179](https://github.com/multipath-tcp/mptcp/blob/mptcp_v0.95/net/mptcp/mptcp_rr.c#L179)



The subflows destination addresses must be provided in integer binary form and in network order using the conversion:

1. **inet\_pton**<sup>7</sup>: converts the IP address from string format to binary format;
2. **b2a\_hex**<sup>8</sup>: converts the binary data into hexadecimal integer representation;
3. **htonl**<sup>9</sup>: converts the integer to network byte order.

e.g.:

```
daddr = htonl(int(b2a_hex(inet_pton("192.168.0.2")), base=16))
```

Using the previous command line, the mRR module will schedule the 80% of input segments to the "192.168.0.2" destination IP and the 20% to "192.168.0.3". These settings can be applied in real time from user space, changing the behavior of the RR scheduler. The mRR scheduler keeps the latest 20 used subflows addresses.

Reading from "/sys/kernel/mptcp\_rr/config" returns the last applied settings and the subflow usage statistics:

```
> cat /sys/kernel/mptcp_rr/config
192.168.0.2 daddr=33597632 num_segments=80 count=10
192.168.0.3 daddr=50374848 num_segments=20 count=2
```

<sup>7</sup> [https://docs.python.org/3/library/socket.html#socket.inet\\_pton](https://docs.python.org/3/library/socket.html#socket.inet_pton)

<sup>8</sup> [https://docs.python.org/3/library/binascii.html#binascii.b2a\\_hex](https://docs.python.org/3/library/binascii.html#binascii.b2a_hex)

<sup>9</sup> <https://docs.python.org/3/library/socket.html#socket.htonl>

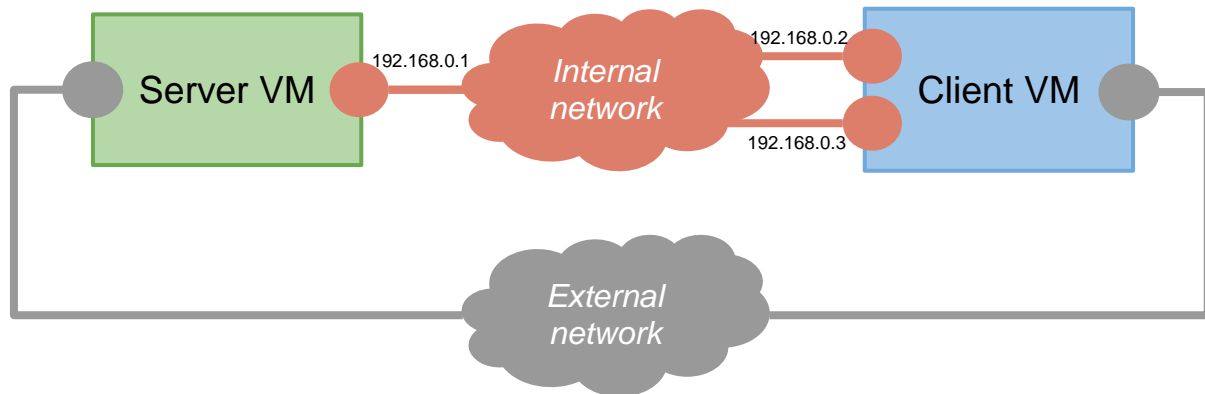
### 7.1.6 Video Server and Video Client Virtual Machines

The Virtual Machine environment used for the implementation is VirtualBox. We used two VM instances configured as follow:

**Table 7-4 VM instances configuration**

	<b>Server openbach-0</b>	<b>Client openbach-1</b>
<b>vCPU</b>	2	4
<b>RAM</b>	4GB	4GB
<b>Network in- terfaces</b>	enp3s0 (Bridged adapter)	enp3s0 (Bridged adapter)
	enp3s8 (Internal Network) 192.168.0.1	enp3s8 (Internal Network) 192.168.0.2
		enp3s08 (Internal Network) 192.168.0.3
<b>Distribution</b>	Ubuntu 16.04.6 LTS	
<b>Kernel</b>	Linux 4.19.105-mptcp-custom (includes the mRR patch)	
<b>Software installed</b>	Docker 20.10.3 OpenBach 2.2.0	
	Video Server	Video Player Test Client Chrome Browser
	OpenBach Conductor OpenBach Agent	OpenBach Agent

The network configuration (Figure 7-8) includes a bridged interface attached to each VM, used for communicating with the external network and an internal interface used for communication between the services running inside the VMs. The client VM has two internal network interfaces (192.168.0.{2,3}) while the server has one internal network interface (192.168.0.1); in order to run the tests in a controlled environment emulating a real infrastructure, we turned MPTCP off for external interfaces and on for internal network.



**Figure 7-8 Virtual machines network configuration.**

On the client VM we used the following network configuration hook script to configure a network routing suitable for MPTCP:

```

/etc/network/if-up.d/mptcp_up

#!/bin/bash

if [ `grep ${IFACE} /etc/iproute2/rt_tables | wc -l` -eq 0 ]; then
    NUM=`cat /etc/iproute2/rt_tables | wc -l`
    echo "$NUM ${IFACE}" >> /etc/iproute2/rt_tables
fi

ip=$(ip addr show dev ${IFACE} | grep 'inet ' | awk '{print $2}' | sed s/"\./"\"/)
netmask=$(ip addr show ${IFACE} | grep 'inet ' | awk '{print $4}')
net=$(ipcalc $ip $netmask | grep Network | awk '{print $2}')

# add a rule for the interface IP address
ip rule add from ${ip} table ${IFACE}

# add a custom route
ip route add to ${net} dev ${interface} scope link table ${IFACE}

```

On the server VM we installed the Video Server service using a docker-compose deployment that starts all the required docker containers; the proxy service listens on HTTPS port 443. We uploaded the 8K video sequence “*First-8K-Video-from-Space*” from NASA that can be viewed accessing the service using the URL

“[https://<server\\_ip>/embed/5e6a09fe83bd4389fc372cbf](https://<server_ip>/embed/5e6a09fe83bd4389fc372cbf)”.

The Video Server setup includes the files reported in Table Table 7-5:

**Table 7-5 Video Server Setup**

<b>openbach-0:/home/openbach/quavstreams</b>	
data/	VOD and database storage directory
le/	letsencrypt certificates directory
server-configs/ quavstreams.local.http.conf.template quavstreams.local.stream.conf.template	nginx proxy configuration files
docker-compose.yml	docker-compose definition file
run.sh	startup script
nginx-frontend.tar.xz nginx-vod-module.tar.xz quavencoder-service-vod.tar.xz quavstreams-api.tar.xz quavstreams-web-ng.tar.xz	docker images

The docker-compose definition starts all the containers using the “restart:unless-stopped” option, so the configuration is persistent across reboots.

The local HTTPS port uses a self-signed certificate that should be manually accepted by the user when visiting the page. Renaming the server-configs/quavstreams.local.\* files to a well formed DNS name (e.g. openbach-server.org) and attaching a public accessible IP to the Video Server instance, allows the letsencrypt service to obtain a valid SSL certificate. Note that after any change to these configurations files, the “./run.sh up” command must be executed.

### 7.1.7 OpenBACH architecture and integration

We set up an OpenBach architecture composed by two nodes:

- **openbach-0**: controller and agent, used for controlling the Video Server host;
- **openbach-1**: agent, used for controlling the Video Test clients.

The OpenBach installation procedure<sup>10</sup> requires settings the IP addresses to which the host-names are resolved:

```
/etc/hosts
...
<external_ip1> openbach-0
<external_ip2> openbach-1
```

The same `/etc/hosts` configuration should be applied inside both the VM instances in order to make the OpenBach components communicate only via the external interfaces to not interfere with the traffic generated by video tests.

The ansible inventory is configured using the following YAML file:

```
openbach-inventory.yaml
all:
  vars:
    ansible_user: "openbach"
    ansible_ssh_pass: "openbach"
    ansible_become_pass: "openbach"
  children:
    controller:
      hosts:
        openbach-0:
          openbach_name: openbach-0
    agent:
      hosts:
        openbach-1:
          openbach_name: openbach-1
```

Openbach installation commands:

```
> git clone --depth 1 https://forge.net4sat.org/openbach/openbach.git
```

<sup>10</sup> <http://www.openbach.org/content/get.php>

```
> cd openbach/ansible

> ansible-playbook -i ../../openbach-inventory.yaml install.yaml \
  -e '{"openbach_jobs_folders": ["../../openbach-jobs/"]}'
```

#### 7.1.7.1 The OpenBach custom jobs

The integration of the Video Server and the Video Client components into OpenBach infrastructures required the development of the following OpenBach jobs:

- **quavstreams**: it wraps the Video Server component and the the MPTCP scheduler controller; the job parameters allow to control the MPTCP path manager, scheduler, the roundrobin quota and the congestion control used at server side;
- **quavstreams-client**: it wraps the Selenium/Chromium browser Video test client; is allows to choose the HTTP address to be open in the headless browser and the browser window size.

Each job exposes the following parameters to the OpenBach admin interface:

quavstreams job parameters	
mptcp-path-manager	The mptcp path manager (default, fullmesh, ndiffports, binder)
mptcp-scheduler	The mptcp scheduler (default, roundrobin, redundant)
mptcp-rr-quota	The mptcp roundrobin quota ("IP:quota" comma separated list)
mptcp-congestion-control	The mptcp congestion control (olia, balia, wvegas)

Is it worth noticing that the controlled MPTCP parameters have a system wide validity, so it does not make sense to run more than one quavstreams job instance at once, since the latest started job will overwrite the current MPTCP system configuration.

The quavstreams job provides the following statistic values for each job run:

quavstreams job statistics	
index	The play chunk index. The play time can be calculated as: "index × 3.2s"
level	The play level in the range 0..5 (Table 2.3)
rebuf	The play rebuffering count
rebufTime	The play total rebuffering time (seconds)
screenWidth	The play screenWidth (pixels)
screenHeight	The play screenHeight (pixels)
bandwidth	The play bandwidth (bps)
queue	The play queue (seconds)
currentTime	The play currentTime (seconds)
bytesSent	The play total bytes sent
<dest_ip>_rtt	The Round Trip Time (ms)
<dest_ip>_cwnd	The TCP Congestion Window (number of packets)
<dest_ip>_bytes_sent	The total bytes sent
<dest_ip>_delivery_bitrate	The delivery bitrate (bps)
<dest_ip>_mptcp_path_manager	The MPTCP configured path manager
<dest_ip>_mptcp_scheduler	The MPTCP configured scheduler
<dest_ip>_mptcp_quota_<IP>	The MPTCP configured mRR quota for each IP
<dest_ip>_mptcp_congestion_control	The MPTCP configured congestion control

The <dest\_ip> prefix is set to the destination IP address of each TCP subflow.

The quavstreams-client job can be configured using the following parameters:

quavstreams-client job parameters	
url	The connection URL
time	The job duration (seconds)
width	The window width (pixels)
height	The window height (pixels)
mptcp-interfaces-disabled	The mptcp interfaces to disable (comma separated list)
tc-setup	The TC setup to apply in JSON format (ex. [{"interface": "enp0s8", "index": 0, "delay": 10, "rate": 10000}]). If present, a TC configuration is applied to the selected interfaces adding a delay (expressed in milliseconds) and a rate limit (expressed in kbps).

The tc-setup parameter is useful to test a network scenario with network links with limited bandwidth capacity and/or delay. The TC configuration is applied globally, so only one instance of the client should be started on the same host with this option activated.

The TC setup function is implemented in the job bash script mptcp.sh:

#### /opt/openbach/agent/jobs/quavstreams-client/mptcp.sh

```
function tc_start() {
    local interface=${1}; shift
    local index=${1}; shift
    local delay=${1:-10}; shift
    local rate=${1:-100000}; shift
    # calculates a buffer size (in packets) equal to two times the
    # bandwidth-delay product
    local limit=$((2*$rate*$delay/(8*1500)))
    if [ $limit -lt 20 ]; then
        limit=20
    fi
    # adds a input device
    modprobe ifb numifbs=2
    ip link set dev ifb${index} up

    tc qdisc add dev $interface handle ffff: ingress
    tc filter add dev $interface parent ffff: protocol ip \
        u32 match u32 0 0 action mirred egress redirect dev ifb${index}
```



```
# configures the delay
tc qdisc add dev ifb${index} root handle 1:0 netem delay \
  ${delay}ms rate ${rate}kbit limit ${limit}
# configures the rate limiter
tc qdisc add dev $interface root handle 1:0 netem delay \
  ${delay}ms rate ${rate}kbit limit ${limit}

tc qdisc show dev ifb${index}
}
```

The quavstreams-client job outputs the following statistics:

quavstreams-client job statistics	
t	The current play time in seconds
buffer	The play buffer in seconds. It has the same value of “buffer” on the server side.
totalVideoFrames	The total video frames decoded by the player
droppedVideoFrames	The total video frames dropped by the player. An high value of dropped frames is a symptom of excessive CPU load at the client.
<interface>_rate	The <interface> rate limitation applied (in Kbps)
<interface>_delay	The <interface> delay applied (in milliseconds)

### 7.1.8 The Grafana “Live scenario” dashboard

A custom Grafana dashboard has been configured to make the services debugging more simple. At this purpose, the following values are displayed into the “Live scenario” Grafana dashboard (accessible at <http://openbach-0:3000/d/live/live-scenario>):

Setup			
MPTCP settings(congestion control, path manager and scheduler)	mRR quota	TC rate limits at client	TC delays at client
Video Server			
Congestion Window (for each internal interface destination)	RTT (for each internal interface destination)	Send rate (delivery rate for each internal interface destination)	
Video Player			
Download rate (measured at client)	Video quality level	Total rebuffering time	Player queue (in seconds)

#### 7.1.8.1 mRR test scenarios

In the following test scenarios we configured a client-server setup with a percentage of network traffic assigned to both 192.168.0.2 ( $P_1$ ) and 192.168.0.3 ( $P_2$ ) destination IPs. In all the tests, the client points to the server url address, running an headless browser with a window size that allows to play the 8K video at the higher quality level. At the client side we disable MPTCP on the interfaces connected to the external networking (enp0s3 and tun0). The following settings are applied to all the tests, replacing the  $Q_1$ ,  $Q_2$ ,  $D_1$ ,  $D_2$ ,  $R_1$ ,  $R_2$  variables:

- **server job** (running on openbach-0):
  - mptcp-path-manager=fullmesh
  - mptcp-scheduler=roundrobin<sup>11</sup>
  - mptcp-rr-quota=192.168.0.2= $Q_1$ , 192.168.0.3= $Q_2$
  - mptcp-congestion-control=balia
- **client job** (running on openbach-1):
  - url=https://192.168.0.1/embed/5e6a09fe83bd4389fc372cbf?autoplay=1
  - width=7680
  - height=4320
    - mptcp-interfaces-disabled=enp0s3,tun0
  - tc-setup=[

<sup>11</sup> With mRR patch

```
{ "interface": "enp0s8", "index": 0, "delay":  $D_1$ , "rate":  $R_1$  },
{ "interface": "enp0s9", "index": 1, "delay":  $D_2$ , "rate":  $R_2$  }
]
```

### 7.1.8.1.1 Test scenario 1

In this scenario we apply a traffic shaping configuration where we assign to the path  $P_2$  less network capacity and more delay than the path  $P_1$ . The MPTCP quota is kept equal.

Path	$Q_i$ (mRR quota)	$D_i$ (delay)	$R_i$ (rate)
$P_1$	100	20	10000
$P_2$	100	200	2000

From the test results of Figure 7-9 we can see that the average quality level is equal to 3.



Figure 7-9 Test scenario 1 results

### 7.1.8.1.2 Test scenario 2

In this scenario we apply the same traffic shaping configuration of Test scenario 1, but we assign a different mRR quota to the two paths: taking advantage of the knowledge of the rate limitation on path  $P_2$ , we assign a quota  $Q_2=10$ , while we set  $Q_1=90$  to the first path.

Path	$Q_i$ (mRR quota)	$D_i$ (delay)	$R_i$ (rate)
$P_1$	90	20	10000
$P_2$	10	200	2000

From the test results of Figure 7-10 we can see that the average quality level is equal to 4.7, that is rough one level above the previous test. This can be explained looking at the Send rate value for  $P_1$  (6.56 Mbps) that is greater than in the previous scenario (4.42 Mbps); changing the mRR quota forced the scheduler to send more data on path  $P_1$  instead of  $P_2$ .



Figure 7-10 Test scenario 2 results

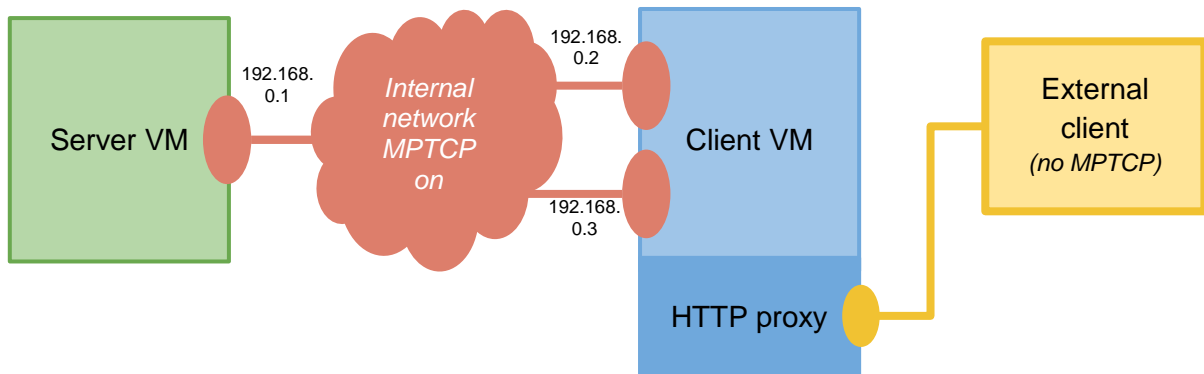
### 7.1.9 HTTP Proxy configuration

Testing the communication with the server taking advantage of the multiple paths availability requires to run a browser on a machine with a multipath TCP supported kernel. At the purpose of using a generic machine to access the video server (Figure 7-11), an HTTP proxy (Nginx) is installed on the openbach-1 instance. With this setup the Client Virtual Machine acts as a bridge between a generic client and the Video Server Virtual Machine. The configuration used for proxying the requests to the server instance is the following<sup>12</sup>:

<sup>12</sup> [http://nginx.org/en/docs/http/nginx\\_http\\_proxy\\_module.html](http://nginx.org/en/docs/http/nginx_http_proxy_module.html)

**/etc/nginx/sites-enabled/default**

```
server {
    location / {
        proxy_pass      https://192.168.0.1:443;
        proxy_set_header Host      $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_buffering off;
    }
}
```



**Figure 7-11 Virtual machines network configuration with HTTP proxy.**

## 7.2 KR side Traffic Controller description

The Traffic Controller, as described in the section 5.2, is located behind terrestrial and satellite DUs and before the user terminals to provide multi-connectivity control. It consists of a traffic status monitoring module, a control decision module, and a control action module. Its functionality includes traffic monitoring of status of multi-channels, traffic hand-off/recovery decision making, and traffic control by streaming session management.

We implemented TC functionality over a super-micro 1U server (Figure 7-12) with Intel Atom 8core 2.2Ghz CPU and 8GB RAM, and 1TB HDD. The OS used is Ubuntu 18.04.04 LTS.



**Figure 7-12 TC hardware**

To verify the functionality of the TC. We setup a testbed as depicted in a Figure 7-13. It consists of a streaming server, WAN aggregation switch, two WAN gateways for both a cellular network and satellite network, a TC, an Access Point (AP) for WiFi connectivity, a management console, and user terminals.

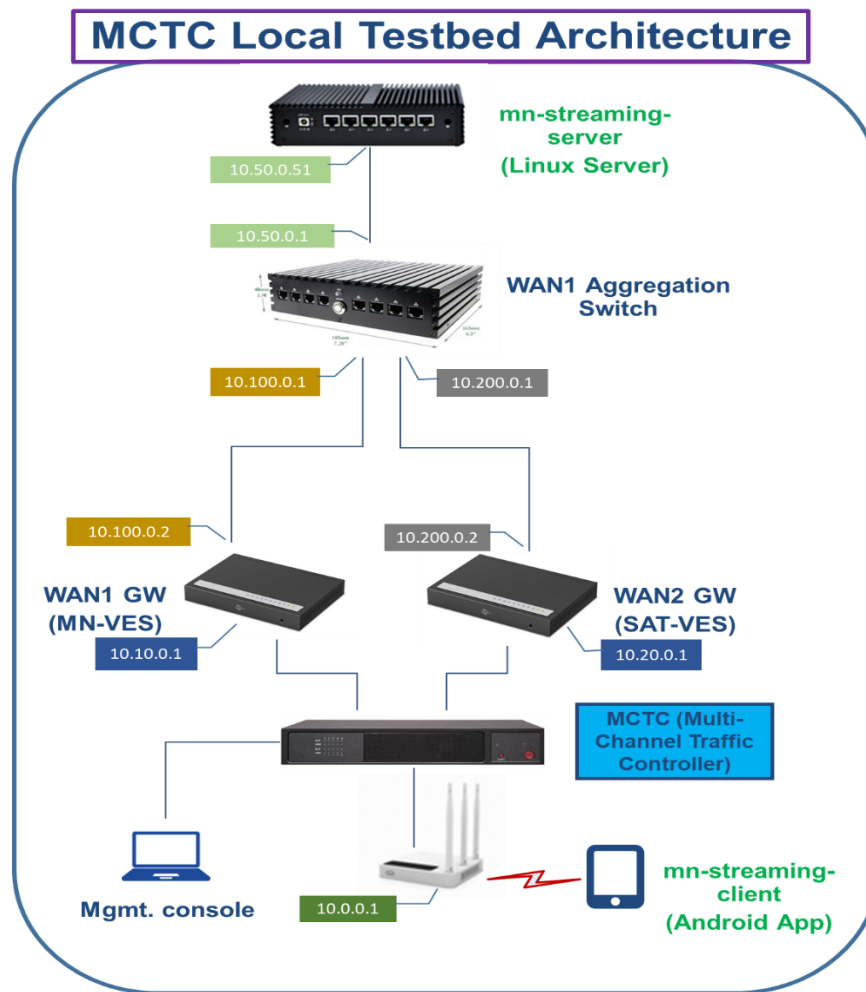


Figure 7-13 Test-bed for multi-connectivity functionality testing.

For video streaming, we installed a HTTP live streaming (HLS) server in the streaming server and HLS clients in the android handsets. WAN aggregation switch which sits before the streaming server plays a role of switching and routing client traffic to the streaming server and vice versa. It simply checks incoming packets of either links from the clients, and routes them to the streaming server and consequently routes response packets from the streaming sever to the active link toward the clients. Two WAN gateways plays roles of the termination interfaces of the data network operator where the streaming service is provided. There exists both terrestrial and satellite links between two gateways and TC. We assume that TC performs traffic monitoring and control at IP layer.

For the traffic control, we use an open source firewall software called pfSense. Once the TC is installed and configured, it monitors the status of both links. The status is shown in the Figure 7-14. It monitors PPS and BPS of the links.

The screenshot shows the pfSense web interface at the 'Diagnostics / States' page. The 'State Filter' section is at the top, with 'Interface' set to 'all' and a 'Filter expression' field. Below this is a table of active states.

Interface	Protocol	Source (Original Source) -> Destination (Original Destination)	State	Packets	Bytes
OPT2_MGMT_WAN	udp	192.168.0.1:45494 -> 255.255.255.255:7437	NO_TRAFFIC:SINGLE	8,098 K / 0	1.55 MiB / 0 B
WAN1_MIN	icmp	10.10.0.33:3749 -> 10.100.0.1:3749	0:0	572 / 572	16 KiB / 16 KiB
WAN1_MIN	udp	10.10.0.33:34524 -> 192.5.5.241:53	SINGLE:NO_TRAFFIC	1 / 1	56 B / 84 B
LAN	tcp	10.12.0.10:51854 -> 10.12.0.1:443	FIN_WAIT_2:FIN_WAIT_2	6 / 6	776 B / 398 B
WAN1_MIN	udp	10.10.0.33:36976 -> 199.9.14.201:53	SINGLE:NO_TRAFFIC	1 / 1	56 B / 84 B
WAN1_MIN	udp	10.10.0.33:37993 -> 199.7.91.13:53	SINGLE:NO_TRAFFIC	1 / 1	56 B / 84 B

**Figure 7-14 Snapshot of link status monitored by TC.**

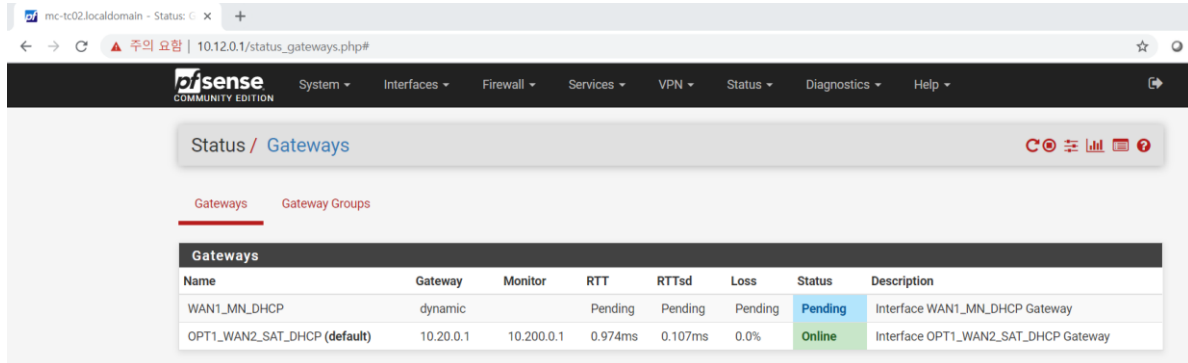
When a particular link is physically down or congested due to long delay, TC decides to failover to another active link and also failback when the link is recovered. Link failure can occur for various reasons such as entering into gap and heavy traffic load, etc. To test link failure, we physically remove a cable from the emulated terrestrial link as shown in Figure 7-15.



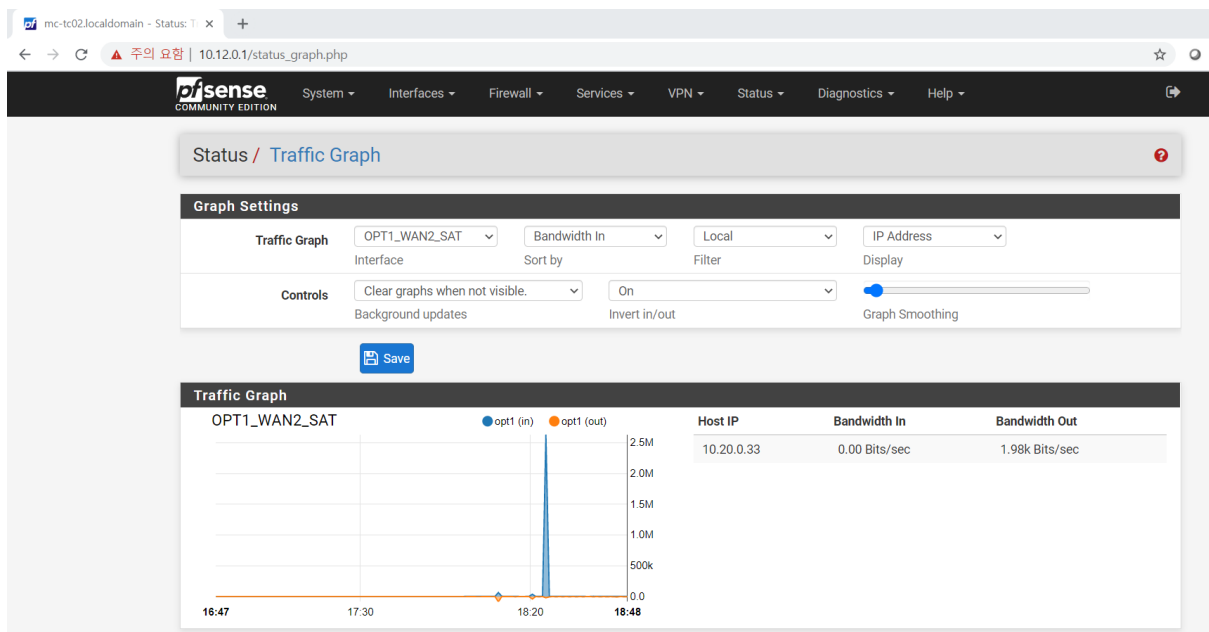
**Figure 7-15 Link failure test setup**



Figure x shows when a terrestrial link is down which becomes a “pending” status and satellite link is now a default link with an “online” status. Traffic status of the satellite link at the time of failover is depicted in the Figure 7-16.



**Figure 7-16 Status of the links when a failover from terrestrial to satellite links occurred**



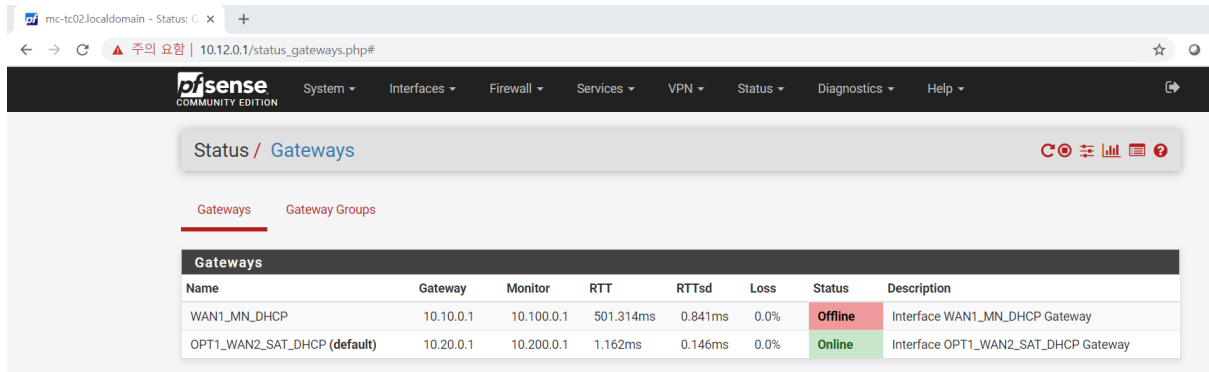
**Figure 7-17 Satellite traffic status when a failover occurred**

We also tested a failover case when traffic congestion happened. While streaming service is in progress over the terrestrial link, we intently applied 500ms delay to this link as shown in the Figure 7-18 below.

```
etri@qkdn-node01: ~
etri@qkdn-node01:~$ sudo tc qdisc show dev enp16s0
qdisc mq 0: root
qdisc fq_codel 0: parent :2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval 100.0ms memory_limit 32Mb ecn
qdisc fq_codel 0: parent :1 limit 10240p flows 1024 quantum 1514 target 5.0ms interval 100.0ms memory_limit 32Mb ecn
etri@qkdn-node01:~$ sudo tc qdisc add dev enp16s0 root netem delay 500ms
etri@qkdn-node01:~$ sudo tc qdisc show dev enp16s0
qdisc netem 8002: root refcnt 9 limit 1000 delay 500.0ms
```

**Figure 7-18 Test result of the failover case i traffic congestion**

TC then decides a failover to the satellite link as illustrated in the Figure 7-19. Again, the satellite link becomes default link with an “online” status and terrestrial link becomes an “offline” status.

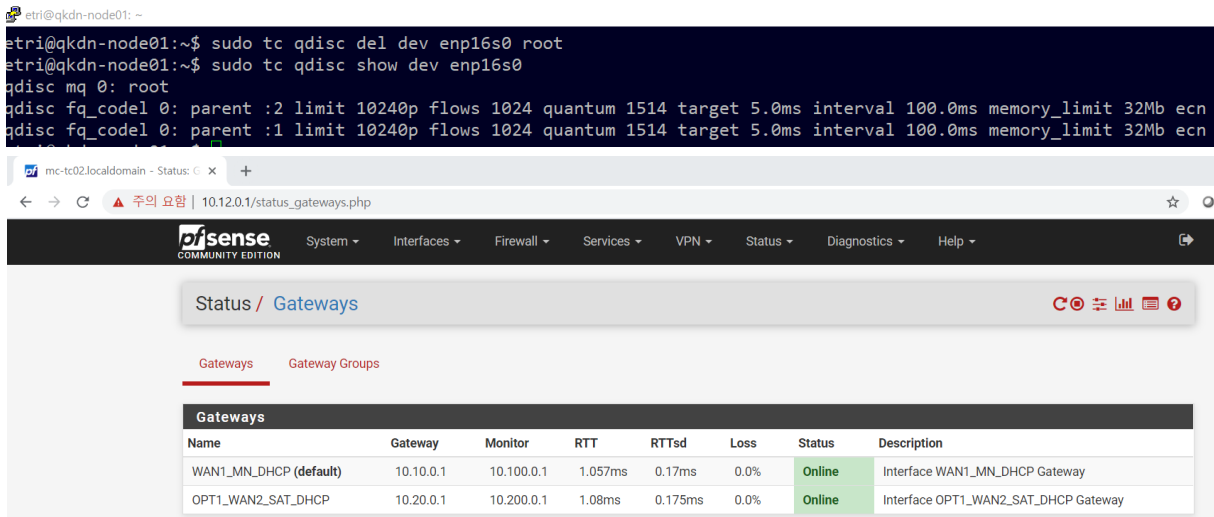


Name	Gateway	Monitor	RTT	RTTsd	Loss	Status	Description
WAN1_MN_DHCP	10.10.0.1	10.100.0.1	501.314ms	0.841ms	0.0%	Offline	Interface WAN1_MN_DHCP Gateway
OPT1_WAN2_SAT_DHCP (default)	10.20.0.1	10.200.0.1	1.162ms	0.146ms	0.0%	Online	Interface OPT1_WAN2_SAT_DHCP Gateway

**Figure 7-19 Failover decision testing**

To test failback functionality, we removed the applied artificial delay. TC then recovers the default link from satellite into terrestrial as depicted in Figure 7-20.

```
etri@qkdn-node01: ~
etri@qkdn-node01:~$ sudo tc qdisc del dev enp16s0 root
etri@qkdn-node01:~$ sudo tc qdisc show dev enp16s0
qdisc mq 0: root
qdisc fq_codel 0: parent :2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval 100.0ms memory_limit 32Mb ecn
qdisc fq_codel 0: parent :1 limit 10240p flows 1024 quantum 1514 target 5.0ms interval 100.0ms memory_limit 32Mb ecn
```



Name	Gateway	Monitor	RTT	RTTsd	Loss	Status	Description
WAN1_MN_DHCP (default)	10.10.0.1	10.100.0.1	1.057ms	0.17ms	0.0%	Online	Interface WAN1_MN_DHCP Gateway
OPT1_WAN2_SAT_DHCP	10.20.0.1	10.200.0.1	1.08ms	0.175ms	0.0%	Online	Interface OPT1_WAN2_SAT_DHCP Gateway

**Figure 7-20 Failback functionality testing**

We have described the TC functionality and verification testing in various scenarios. We are going to perform actual testing integrated into real PoC environment according to the milestone of WP5. The detailed testing scenarios and verification results will be described in detail in the relevant deliverables of WP5.

## 8 Conclusions

The document reported the rationale behind the selection of the multi-connectivity algorithms that will be demonstrated in the PoC of the project, as well simulation and implementation details of these algorithms.

Details were provided for both the simulation testing environments and for the software developed in view of the final demonstration.

The EU partners detailed their opensource multi-connectivity simulator specifications and main functionalities and then reported the developed components needed to assure QoE-aware multi-path adaptive video streaming.

The KR partners presented their simulative environment and algorithm of choice for multi-RAT load balancing, together with the detail of their implementation of a traffic flow controller.

The reported result are adequate to assure that the activities of the project may enter their final stages. The future activities related to multi-connectivity will continue in the scope of WP5 with the final implementation activities and system-level testing.

## 9 References

- [1] E. De Santis, A. Giuseppe, A. Pietrabissa, and F. D. Priscoli, "Multi-RAT Wireless Network Simulator," Apr. 2020.
- [2] Emanuele De Santis et al., "5G-ALLSTAR Wireless Network Simulator," 2020. [Online]. Available: <https://github.com/trunk96/wireless-network-simulator-v2>.
- [3] G. Maral and M. Bousquet, *Satellite Communications Systems - Systems, Techniques and Technologies*. Chichester, UK: John Wiley & Sons, Ltd, 2009.
- [4] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 2020.
- [5] TSGR, "TS 138 101-1 - V15.3.0 - 5G; NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone (3GPP TS 38.101-1 version 15.3.0 Release 15)," 2018.
- [6] TSGR, "TS 138 101-2 - V15.3.0 - 5G; NR; User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone (3GPP TS 38.101-2 version 15.3.0 Release 15)," 2018.
- [7] "Final report for COST Action 231."
- [8] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," Dec. 2013.
- [9] Khalil, Hassan K., and J. W. Grizzle. *Nonlinear systems*. Vol. 3. Prentice hall, 1996.
- [10] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH)", *Packet Video Workshop 2013*, San Jose, CA, USA, December 2013
- [11] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen. Flicker effects in adaptive video streaming to handheld devices. In *Proc. of 19th ACM international conference on Multimedia*, pages 463–472. ACM, 2011
- [12] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *Proc. of ACM SIGCOMM 2011*, pages 362–373, 2011.
- [13] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated internet video control plane. In *Proc. of ACM SIGCOMM 2012*, *SIGCOMM '12*, pages 359–370, 2012.
- [14] A. B. V. Sekar, A. Akella, S. S. I. Stoica, and H. Zhang. Developing a predictive model of quality of experience for internet video. In *Proc. of ACM SIGCOMM '13*, 2013.
- [15] ROSEN, Rami. *Network Administration*. In: *Linux Kernel Networking*. Apress, Berkeley, CA, 2014. p. 571-588.
- [16] S. M. Shahid, Y. T. Seyoum, S. H. Won, and S. Kwon, "Load balancing for 5G integrated satellite-terrestrial networks," *IEEE Access*, vol. 8, pp. 132144-132156, 2020.
- [17] K. M. Addali, S. Y. B. Melhem, Y. Khamayseh, Z. Zhang, and M. Kadoch, "Dynamic mobility load balancing for 5G small-cell networks based on utility functions," *IEEE Access*, vol. 7, pp. 126998–127011, 2019.
- [18] M. M. Hasan, S. Kwon, and J.-H. Na, "Adaptive mobility load balancing algorithm for LTE small-cell networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2205–2217, Apr. 2018.

- [19] C. Rosa, K. Pedersen, H. Wang, P.-H. Michaelsen, S. Barbera, E. Malkamaki, T. Henttonen, and B. Sebire, "Dual connectivity for LTE small cell evolution: Functionality and performance aspects," *IEEE Commun. Mag.*, vol. 54, no. 6, pp. 137–143, Jun. 2016.
- [20] System Architecture for the 5G System; Stage 2, document TS 23.501, Version 16.3.0, 3GPP, Dec. 2019. [Online]. Available: <https://portal.3gpp.org>
- [21] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*. New York, NY, USA: Academic, 2018.
- [22] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
- [23] R. Antonioli, G. Parente, C. Silva, D. Sousa, E. Rodrigues, T. Maciel, and F. Cavalcanti, "Dual connectivity for LTE-NR cellular networks: challenges and open issues," *J. Commun. Inf. Syst.*, vol. 33, no. 1, pp. 282–294, 2018.
- [24] Radio Resource Control (RRC) Protocol Specification, document TS 38.331, Version 15.8.0, 3GPP, Dec. 2019. [Online]. Available: <https://portal.3gpp.org/>
- [25] R. Kwan, R. Arnott, R. Paterson, R. Trivisonno, and M. Kubota, "On mobility load balancing for LTE systems," in *Proc. IEEE 72nd Veh. Technol. Conf. (Fall)*, Sep. 2010, pp. 1–5.
- [26] NR; NR and NG-RAN Overall Description, document TS 38.300, Version 16.0.0, 3GPP, Dec. 2019. [Online]. Available: <https://portal.3gpp.org>
- [27] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*. New York, NY, USA: McGraw-Hill, 2002.
- [28] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, Jun. 2019.
- [29] TCP Extensions for Multipath Operation with Multiple Addresses, March 2020, <https://tools.ietf.org/html/rfc8684>
- [30] PAASCH, Christoph; BONAVENTURE, Olivier. Multipath tcp. *Communications of the ACM*, 2014, 57.4: 51-57.
- [31] PAASCH, Christoph, et al. Experimental evaluation of multipath TCP schedulers. In: *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*. 2014. p. 27-32.
- [32] Luca Boccassi, Marwan M. Fayed, and Mahesh K. Marina. 2013. Binder: a system to aggregate multiple internet gateways in community networks. In *Proceedings of the 2013 ACM MobiCom workshop on Lowest cost denominator networking for universal access (LCDNet '13)*. Association for Computing Machinery, New York, NY, USA, 3–8. DOI:<https://doi.org/10.1145/2502880.2502894>
- [33] KATO, T., et al. Experimental analysis of MPTCP congestion control algorithms; LIA, OLIA and BALIA. In: *8th International Conference on Theory and Practice in Modern Computing (TPMC 2019)*. p. 135-142.
- [34] RAICIU, Costin; HANDLEY, Mark; WISCHIK, Damon. Coupled congestion control for multipath transport protocols. *IETF RFC 6356*, Oct, 2011.
- [35] KHALILI, Ramin, et al. Opportunistic linked-increases congestion control algorithm for MPTCP. 2013.

- [36] CAO, Yu; XU, Mingwei; FU, Xiaoming. Delay-based congestion control for multipath TCP. In: 2012 20th IEEE international conference on network protocols (ICNP). IEEE, 2012. p. 1-10.
- [37] WALID, Anwar, et al. Balanced linked adaptation congestion control algorithm for MPTCP. Working Draft, IETF Secretariat, Internet-Draft draft-walid-mptcp-congestion-control-04, 2016.
- [38] PENG, Qiuyu, et al. Multipath TCP: Analysis, design, and implementation. IEEE/ACM Transactions on networking, 2014, 24.1: 596-609.