H2020 5Growth Project
Grant No. 856709

User Guide of 5Gr-CTTC-RL

Abstract

This document provides the user guide for the 5Gr-RL reference implementation developed by CTTC, including instructions to install and to configure through its GUI.

Document properties

Document number	
Document title	Installation and First configuration of 5Gr-CTTC-RL
Document responsible	Luca Vettori (CTTC)
Document editor	Luca Vettori (CTTC)
Editorial team	Jorge Baranda (CTTC), Ricardo Martínez (CTTC), Josep Mangues-Bafalluy (CTTC), Engin Zeydan
Target dissemination level	Public
Status of the document	In progress
Version	0.1

Production properties

Reviewers	…

Document history

Revision	Date	Issued by	Description
0.1	May 21st 2020	Luca Vettori (CTTC)	First initial version

Disclaimer

This document has been produced in the context of the 5Growth Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement N° H2020-856709.

Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Description
5Gr-RL	5Gr-Resource Layer
5GT-SO	5Gr-Service Orchestrator
API	Application Programming Interface
DB	Database
ETSI	European Telecommunication Standardization Institute
GUI	Graphical User Interface
IFA	Interfaces and Architecture
NBI	Northbound Interface
NFV	Network Function Virtualization
NFVI	Network Functions Virtualisation Infrastructure
NS	Network Slice
NSD	Network Service Descriptor
pABNO	parent Application Based Network Operations
PoP	Point of Presence
REST	Representational State Transfer
VM	Virtual Machine
VNF	Virtual Network Function
VNFD	VNF Descriptor
VSB	Vertical Service Blueprint
VSD	Vertical Service Descriptor
VSI	Vertical Service Instance
CsA	Connectivity Service Abstraction
InA	Infrastructure Abstraction

1. Introduction

This document is the user guide for the 5Gr-RL implementation (done by CTTC). It includes all the step to install and first configure the tool. The final goal is to have a 5Gr-CTTC-RL platform up and running.

The 5Growth Resource Layer (5Gr-RL) CTTC implementation is an open source software prototype developed in python, which provides all the main 5Gr-RL functionalities required by the 5Gr cases and devised architecture [1]. It can be downloaded from the 5G-TRANSFORMER git internal repository [1] and its installation, configuration and usage are described in the following sections.

2. 5Gr-RL CTTC functionalities

The main features and functionalities of the 5Gr-RL CTTC implementation follow those contained and covered by the "official" 5Gr-RL version available in [1]. Key building blocks entailing DBs, APIs, as well as operational workflows are implemented as the "official" 5Gr-RL implementation does. This allows ensuring the interworking with other 5Gr architectural elements (e.g., 5Gr-SO). Nevertheless, one of the main differences with respect to the "official" version is that the CTTC implementation is mostly network-resource oriented rather than supporting both network and compute resource management. The rationale behind this is that the 5Gr-SO OSM Wrapper (considered in 5Gr CTTC deployment stack) takes over of all the operations related with computing (cloud / DC) resources.

For the sake of completeness, the main features and differences between the two implementations are reported in Table 2-1.

TABLE 2-1: 5GR-RL CTTC FEATURES AND DIFFERENCES W.R.T OFFICIAL VERSION

Functionality	Description
REST-based NBI	Fully integration with 5Gr-SO but only for handling networking resources • Retrieve abstracted network resources (i.e., set of LLs) • Creation/Termination of Inter-NFVI-PoPs connectivity Creation/Termination of Intra-NFVI-PoP virtual links
Inner structure of CTTC implementation	Referring to the "official" 5Gr-RL implementation, but different technology encompassing more blocks and tables are used and thus added • Python language
SBI	It interacts directly with both compute and networking resource controllers (i.e., VIM and WIM), avoiding relying on the ETSI IFA005 defined interface allocation/release of networking resources in both cloud and WAN environment
Orchestrator	Main core element of the 5Gr-RL CTTC implementation. It handles and coordinates the rest the components constituting the 5Gr-RL as well as implements specific algorithms/polices
Domain Resource Logic	It handles the algorithms and policies to retrieve the physical resources for both networking and computing elements. • Calculating the aggregation of computing hosts. • Handling networking WAN deployment: decomposing / reduction network element resource graph
Abstraction Logic	It handles the algorithms and policies to create an abstracted view for the resource DB. • Create the abstraction for the network resources • Create the abstraction for the federated network resources
Resource Orchestrator	It handles the algorithms and policies for actual resources used in WIM/interWIMs for the NS lifecycle phases (i.e., instantiation, modification, termination, etc.) • Instantiation/Termination of inter-NFVI-PoP connectivity

	• Instantiation/Termination of intra-NFVI-PoP networks • Working in collaboration with RA element
Placement Algorithm	Algorithm to calculate and select the network resources to accommodate required inter-NFVI-PoP connectivity paths for demanded NS. • K-Paths as output of this Algorithm • External entity connected with a REST API
5Gr-RL CTTC Graphical User Interface (GUI)	GUI enabling to visualize the Abstract View, Resource View, database contents, NBI, etc. handled by the 5Gr-RL element

3. Installation Guide

In order install the 5Gr-CTTC-RL you should clone the software from the github repository of the 5Gr Project.

```
git clone https://github.com/5growth/5gt-mtp.git -b mtp-cttc
```

The instructions are also written in README.md of the repository

3.1. Installation

First, install python 3.6 and virtualenv in your host. This part is depending on your Linux distribution.

Then, in "main" folder (where is located README.md file):

- create virtual environment:

```
virtualenv -p python3.6 venv
source venv/bin/activate
pip3 install -r requirements.txt
```

- in the “conf” directory rename the “rl.cfg” and "ra.cfg" files (that define the parameters of RL and the RA):

```
mv rl.cfg.ini rl.cfg
mv ra.cfg.ini ra.cfg
```

- create "body_input.dat" and "body_output.dat" files:

```
touch body_input.dat body_output.dat
```

In "db" folder, rename mtp_db.db.ini to mtp_db.db

```
mv mtp_db.db.ini mtp_db.db
```

3.2. Configuration

To conclude the configuration of the CTTC-RL and use the software, you can edit the configuration files (conf/rl.cfg and conf/ra.cfg).

For the “rl.cfg”:

```
[rl]
# RL_IP=localhost
RL_PORT=8090
LOG_FILE=rl_log.log
LOG_LEVEL=INFO
OPTION_PA_ALGORITHM=OPTION_B
```

```
OPTION_A_BOOT_QUOTES=False
```

the default options allow you to correctly run the program. By default, the 5Gr-RL CTTC works with CsA option (OPTION_B in the implementation), but you can easily edit to the InA option (OPTION_A in the implementation).

For the "ra.cfg":

```
[DEFAULT]
ra_default=RA_1


[RA_1]
id=1000
ip=
port=
path=/compRoute/
kpaths_for_CsA=1
kpaths_for_InA=3
```

you must fill the RA_1 section with the IP/Port of the RA server. You can add more than one RA server instances add RA_2/N section with the same pattern. The DEFAULT field set the default RA.

Finally, to run the code in "main" folder:

- Activate the virtualenv if not:

```
source venv/bin/activate
```

Then:

```
export PYTHONPATH=$PYTHONPATH:/current-directory/ (pwd)
python nbi/nbi_server.py
```

Once the code is running, you can enter in the GUI putting in the browser:
http://your_ip_address:8090

Default User: admin (Password: admin)

4. 5Gr-RL CTTC user guide

This section provides a brief guideline about how to use the 5Gr-RL CTTC from its web GUI for administrative actions. It should be noted that the 5Gr-RL CTTC GUI interacts with the 5Gr-RL CTTC core functionalities using its REST APIs.

As said in the previous chapter, the GUI can be accessed from a web browser at the following link: http:/5Gr_RL_IP_ADDRESS:8090/ address.

At the beginning, the GUI will initially visualize the authentication page where the user can enter the credentials (i.e., username and password).

4.1. 5Gr-RL CTTC GUI

The 5Gr-RL CTTC GUI is used by the 5Gr service provider to manage the underlying transport stratum for integrated networks, including visualization of all the relevant data related with the abstracted view of infrastructure, visualization of the network resources over every service (NS) is accommodated, as well as visualization and modification of the internal DBs.

To enter in the 5Gr-RL CTTC administration GUI, the user should insert "test" and "test" as username and password in the authentication page (admin credentials can be modified through the 5Gr-RL CTTC user DB page). The main page shown in Figure 4-1 will be visualized.

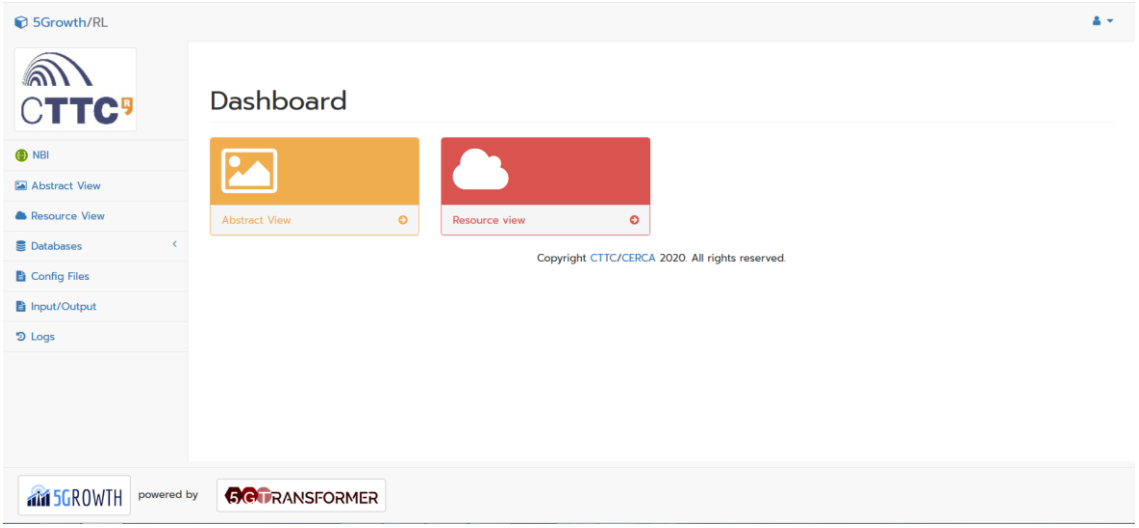


FIGURE 4-1: 5GR-RL CTTC ADMIN GUI – MAIN PAGE

From 5Gr-RL CTTC admin GUI the following actions can be performed:

- Exploration of the 5Gr-RL CTTC NBI swagger API
- Exploration of the all databases' contents (i.e., Domain, NFVI-PoP, Resource Attributes, Logical Links, ServiceId, Stitching, Virtual Links and the Users DB)
- Graphical visualization of the Abstracted View (showing 5Gr RL-configured NFVI-PoPs and Logical Links)

- Graphical visualization of the Resources View (showing the underlying transport stratum for networks)
- Inspection of the main 5Gr-RL CTTC configuration files
- Inspection of 5Gr-RL CTTC log

4.1.1. Exploration of the 5Gr-RL CTTC NBI swagger API

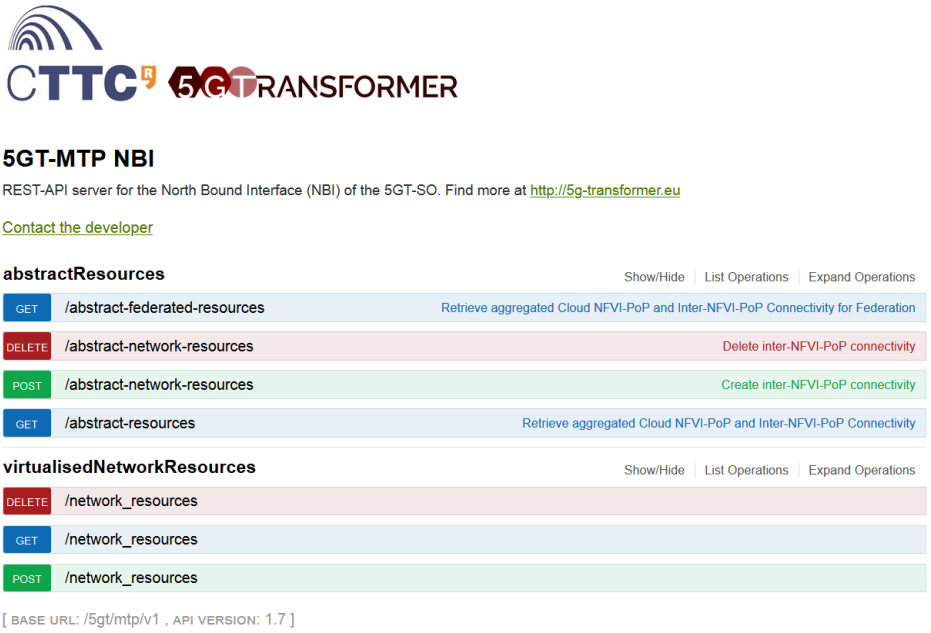


FIGURE 4-2: 5GR-RL CTTC ADMIN GUI – SWAGGER OF THE 5GR-RL NBI

4.1.2. Visualization of Abstracted view exposed to 5Gr-SO

This page represents a graphical representation of Abstracted Resourses exposed to the 5Gr-SO, especially the representation of NFVI-PoPs and Logical Links among them.

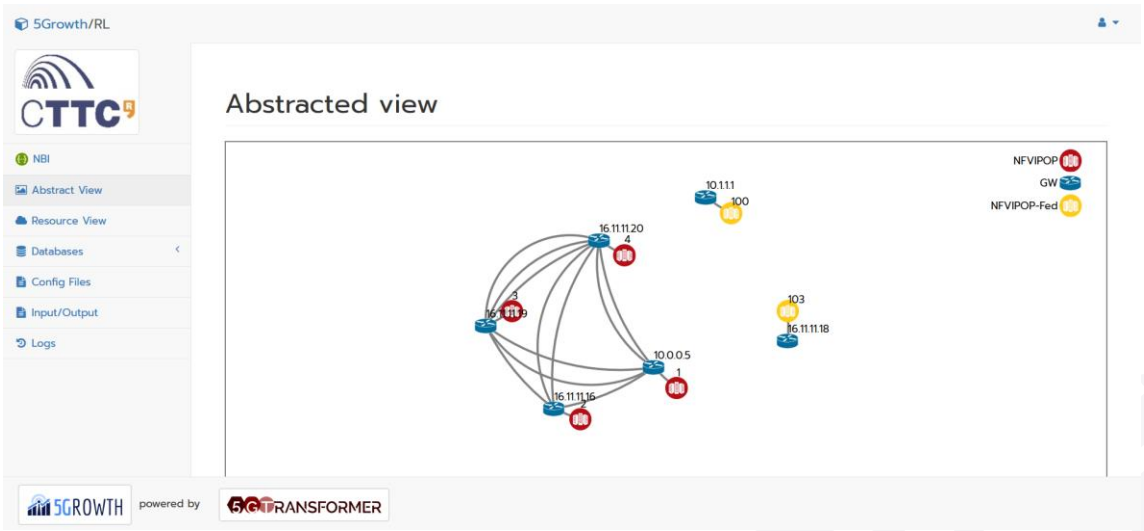


FIGURE 4-3: 5GR-RL CTTC ADMIN GUI – ABSTRACTED VIEW

4.1.3. Databases

Databases' contents (i.e., Domain, NFVI-PoP, Resource Attributes, Logical Links, ServiceId, Stitching, Virtual Links and the User DB) can be checked and seen. Notice that these pages allow adding/removing/modifying entries on most of those DBs. For the ServiceId table (each entry represents an inter-NFVI-PoP connectivity), the GUI enables to visualize the instantiated entry in the underlying infrastructure.

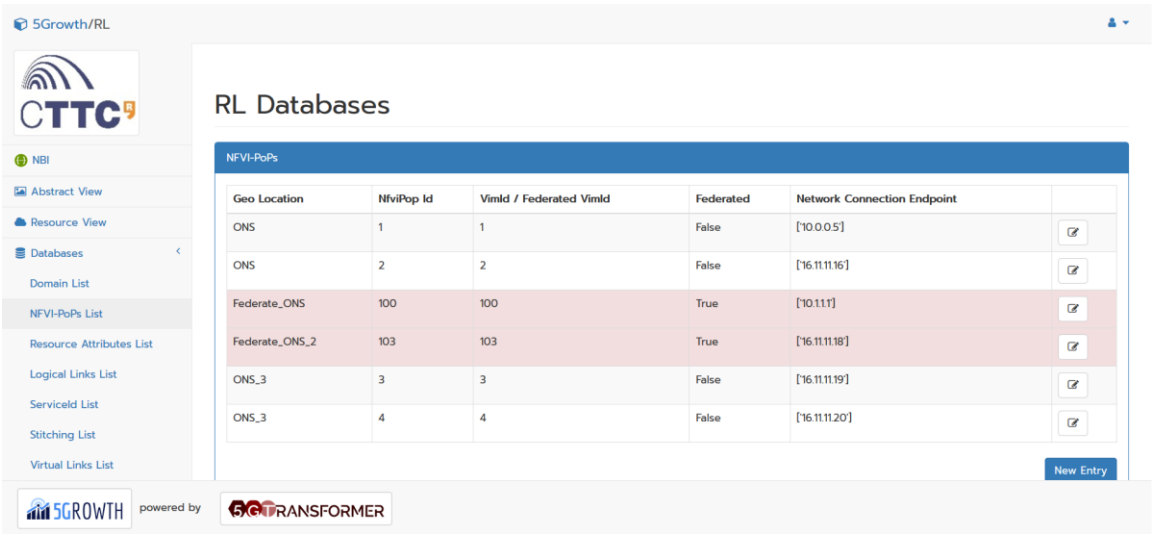


FIGURE 4-4: 5GR-RL CTTC ADMIN GUI – DATABASES

4.1.4. Graphical visualization of instantiated ServiceId

The graphical visualization of an instantiated ServiceId can be opened just clicking on the eye-shaped icon next to the corresponding entry in the DB. By clicking on the pencil-shaped icon next to each entry, some of the fields of the DB entry can be updated.

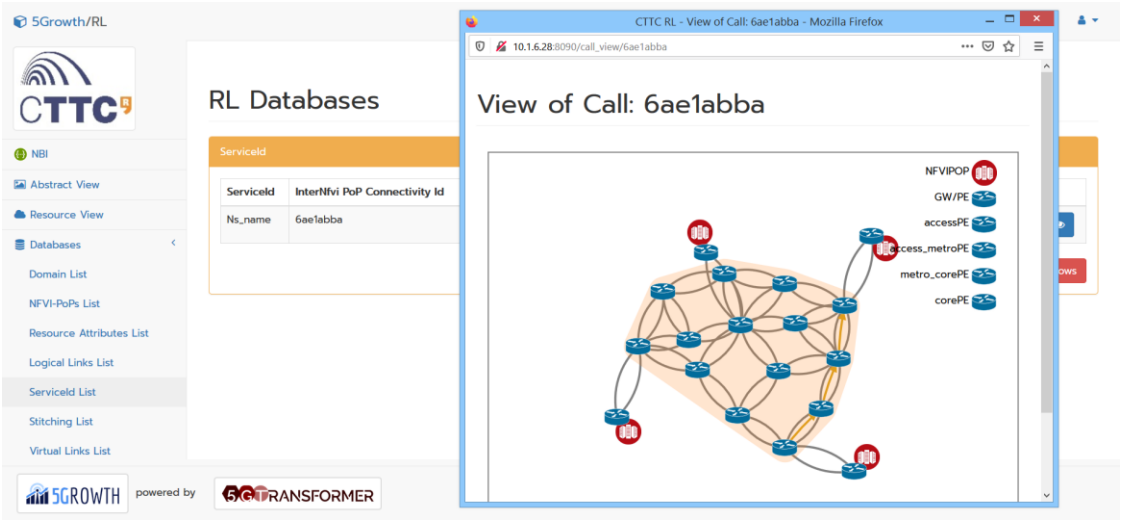


FIGURE 4-5: 5GR-RL CTTC ADMIN GUI – SERVICEID VISUALIZATION

4.1.5. Inspection of main 5Gr-RL CTTC configuration files

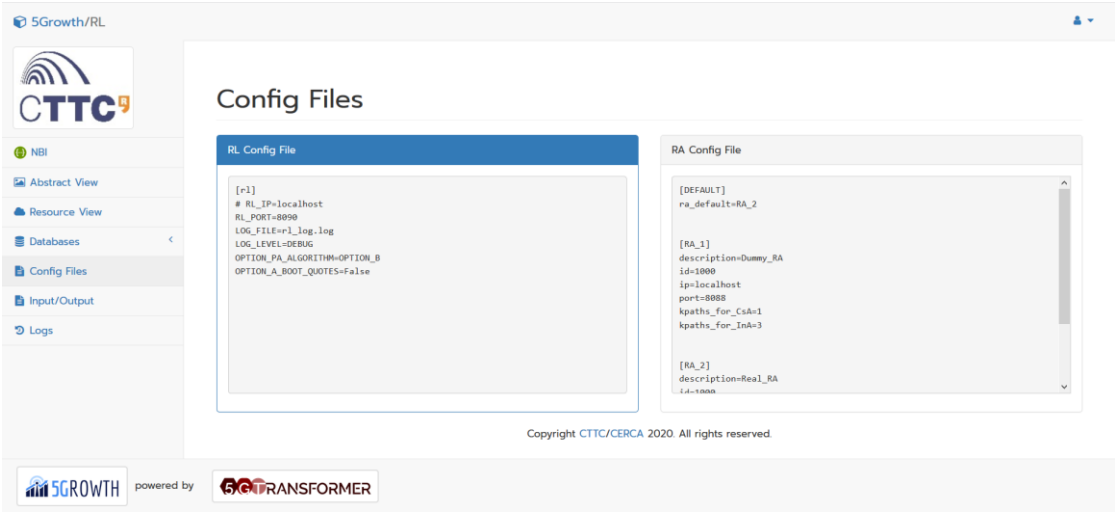


FIGURE 4-6: 5GR-RL CTTC ADMIN GUI – INSPECTION OF 5GR-RL CONFIG FILES

4.1.6. Inspection of 5Gr-RL CTTC log

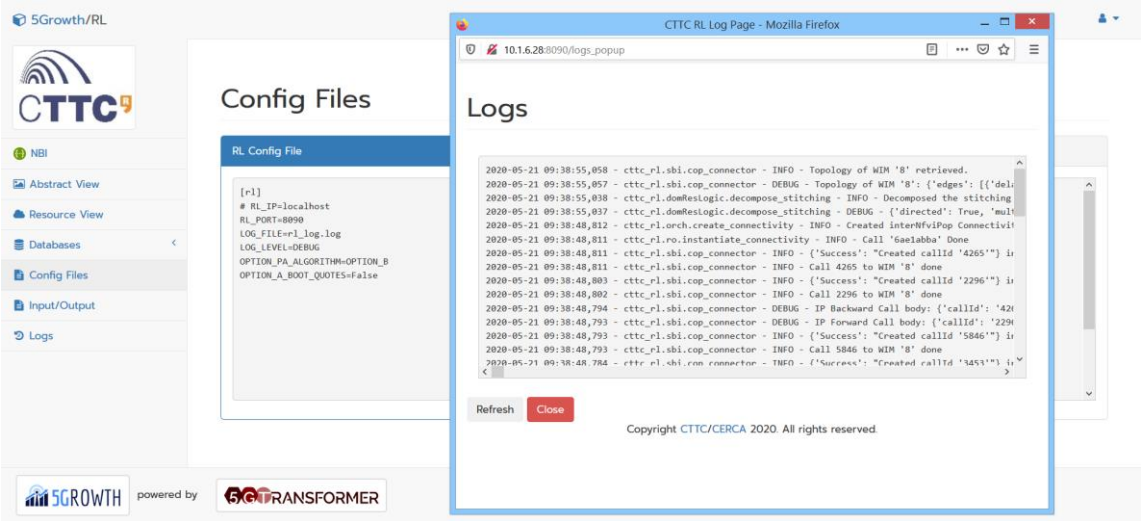


FIGURE 4-7: 5GR-RL CTTC ADMIN GUI – INSPECTION OF 5GR-RL LOGS

4.2. Databases Configuration

The next step is to configure the 5Gr-RL CTTC. To do that, Domain and Logical Link DB should be filled up with specific details of the targeted domain under-deployment. Also, the "Stitching DB" needs to be filled up, before starting the instantiation/termination of the network resources.

If the Domain List and Logical Links List are correctly configured as in the following paragraphs, the Abstract Resources can be visualized in the Abstract View page.

If the Domain List and the Stitching List are correctly configured as in the following paragraphs, the Resource View will visualize the registered underlying network.

4.2.1. Domain List

This page shows the list of Domain elements registered in the 5Gr-RL CTTC. To add a new entry to this list, click on the "New Entry" button. The field to be filled are:

- Name: name chosen for the Domain Element
- Domain_id: VIM_ID or WIM_ID (depending on the type), number
- Type: VIM/WIM
- Account Type: allowed values are: *pbano/cop/onos* for WIM and *openstack* for VIM (mandatory)
- IP Address: IP address of Domain Element
- Port: Port of Domain Element
- Url: url of the Domain Element (for example with an OpenStack http://10.1.6.29/identity/v3)
- Username
- User password

- Tenant Name: Tenant in case of Openstack VIM
- Monitoring EndPoint: field related with the Monitoring Platform (no mandatory)

In the 5Gr-RL, the ID can be chosen as number.

In the case of VIM, for the correct behaviour of the platform, you must create a new entry in the NFVI-PoP List and in the Resource Attributes List as well.

4.2.2. NFVI-PoPs List

This page shows the list of NFVI-PoPs registered in the 5Gr-RL CTTC. To add a new entry to this list, click on the "New Entry" button. The field to be filled are:

- Geographical Location Info: another identifier of the NFVI-PoP
- Nfvi PoP Id: numeric value to identify the NFVI-PoP. *Recommended to set the same value of Vim_Id*
- Federated: True/False if you want to register a regular or Federated Vim
- Vim Id: *Mandatory to choose a value that exists in the Domain List*
- Federated Vim Id: set a numeric value for the Federated Vim_Id
- Network Connection Endpoint: the IP address of the GW of NFVI-PoP. More than one GWs can be registered.

4.2.3. Resource Attribute List

This page shows the list of Resource Attributes of each NFVI-PoP registered in the 5Gr-RL CTTC. To add a new entry to this list, click on the "New Entry" button. The field to be filled are:

- Nfvi-PoP Id: *Mandatory to choose a value that exists in the NFVI-PoP List*
- Zone Id: numeric value to identify the ZoneId. *Recommended to set the same value of NFVI-PoP_Id*
- Zone Name/Zone State/Zone Property/Metadata: specific parameter of the ZoneId.

The other fields of the table are updated to the last time a GET abstract resources has been done.

4.2.4. Logical Links List

This page shows the list of Logical Links registered in the 5Gr-RL CTTC. This part can be skipped if the InA option for RL has been chosen. To add a new entry to this list, click on the "New Entry" button. The field to be filled are:

- Logical Link Id: As convention, it is a 6-digit numeric identifier plus "_f" or "_b" at the end. *It is very important to create two Logical Links (representing the forward and the backward link) with the same Id (for example: "151515_f" and "151515_b")*
- Total BW & Available BW: representing the total and available BW of the Logical Link
- Link Cost Value & Link Cost: representing the Cost of the Link and the type.
- Link Delay Value & Link Delay: representing the Delay of the Link and the type

- SRC Gateway IP Address: *Mandatory to set the value of Network Connection Endpoint of the source NFVI-PoP we want to connect*
- Local Link Id: a logic identifier for the source of the Logical Link (numeric)
- DST Gateway IP Address: *Mandatory to set the value of Network Connection Endpoint of the destination NFVI-PoP we want to connect*
- Remote Link Id: a logic identifier for the destination of the Logical Link (numeric)
- Nfvi-PoP Id: *Mandatory to choose a value that exists in the NFVI-PoP List*
- Zone Id: numeric value to identify the ZoneId. *Recommended to set the same value of NFVI-PoP_Id*
- Network Layer: set it to "L2"
- InterNfvi PoP Network Type & InterNfvi PoP Network Topology: other specific fields of the Logical Links.

As new feature, the Logical Links List can be filled by uploading a file.

4.2.5. Service Id List

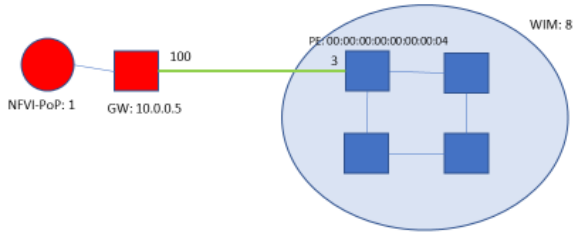
This page shows the list of Service Id created in the 5Gr-RL CTTC. This table will be filled by the 5Gr-RL itself when an inter-NFVI-PoP connectivity has been requested and correctly executed.

4.2.6. Stitching List

This page shows the list of Stitching links registered in the 5Gr-RL CTTC. To add a new entry to this list, click on the "New Entry" button. *It is very important to notice that for each forward stitching link you created, the backward link must be created as well*.

The field to be filled are:

- Ingress Element Id: *Mandatory to insert a value of VIM/WIM from the Domain List, as ingress element.*
- Ingress Element Address: *Mandatory to insert the Network Connection Endpoint (if VIM) or the PE (if WIM) of the corresponding Ingress Element Id*
- aEnd Id: is the Link Id for the Ingress Element
- Egress Element Id: *Mandatory to insert a value of VIM/WIM from the Domain List, as egress element.*
- Egress Element Address: *Mandatory to insert the Network Connection Endpoint (if VIM) or the PE (if WIM) of the corresponding Egress Element Id*
- zEnd Id: is the Link Id for the Egress Element
- Link Cost: numeric
- Link Delay: numeric
- Link Available BW: numeric



2 Entry for the Green Stitching Link:

→ *Forward Stitching Link*:

- Ingress Element Id: 1
- Ingress Element Address: 10.0.0.5
- aLink Id: 100 (arbitrary value)
- Egress Element Id: 8
- Egress Element Address: 00:00:00:00:00:00:00:04
- zLink Id: 3

→ *Backward Stitching Link*:

- Ingress Element Id: 8
- Ingress Element Address: 00:00:00:00:00:00:00:04
- aLink Id: 3
- Egress Element Id: 1
- Egress Element Address: 10.0.0.5
- zLink Id: 100 (arbitrary value)

FIGURE 4-8: EXAMPLE OF STITCHING LINKS

As new feature, the Stitching List can be filled by uploading a file.

4.2.7. Virtual Link List

This page shows the list of Virtual Links created in the 5Gr-RL CTTC. This table will be filled by the 5Gr-RL itself when an intra-NFVI-PoP network has been requested and correctly executed.

4.2.8. User List

This page shows the list of Users registered in the 5Gr-RL CTTC. To add a new entry to this list, click on the “New Entry” button.

References

[1] 5Growth, D2.1: Initial Design of 5G End-to-End Service Platform.
[2] 5Growth RL git internal repository available at: https://5growth.eu/redmine/projects/5growth/repository/5gr-rl (branch: cttc-rl)