User Guide

**Control Plane, User Plane Gateways;**

**LTE EPC;**

**Installation Guide;**

**Troubleshooting Guide;**

**(Release 1.6)**

Table of Contents

[1 Acronyms and Definitions 4](#_Toc37345633)

[2 Introduction 5](#_Toc37345634)

[3 Background 5](#_Toc37345635)

[3.1 Source Repository 6](#_Toc37345636)

[3.1.1 ngic-rtc 6](#_Toc37345637)

[3.1.2 oss\_util 6](#_Toc37345638)

[3.1.3 Freediameter 6](#_Toc37345639)

[3.2 License 7](#_Toc37345640)

[3.3 Specifications Referred 7](#_Toc37345641)

[3.4 Release Updates 7](#_Toc37345642)

[3.4.1 Release Details 7](#_Toc37345643)

[3.4.1.1 New Features 7](#_Toc37345644)

[3.4.1.2 Important Bug Fixes and Open Items 8](#_Toc37345645)

[4 Deployment Options 10](#_Toc37345646)

[4.1 Combined SGW-PGW Architecture 11](#_Toc37345647)

[4.1.1 Combined GW Deployment and Terminology 11](#_Toc37345648)

[4.1.2 Configuration Settings 11](#_Toc37345649)

[4.2 Split GW Architecture 11](#_Toc37345650)

[4.2.1 Split GW Deployment and Terminology 12](#_Toc37345651)

[4.2.2 Configuration Settings 12](#_Toc37345652)

[4.3 Decommission 13](#_Toc37345653)

[5 Downloading, Building and Launching 13](#_Toc37345654)

[5.1 Downloading 13](#_Toc37345655)

[5.2 Installation of Control Plane and Data Plane 13](#_Toc37345656)

[5.2.1 Control Plane Installation 13](#_Toc37345657)

[5.2.2 Data Plane Installation 15](#_Toc37345658)

[5.3 Building 16](#_Toc37345659)

[5.3.1 Runtime and Compile Time Flags 16](#_Toc37345660)

[5.4 Dependencies 18](#_Toc37345661)

[5.5 Configuration 18](#_Toc37345662)

[5.5.1 Editing Control plane Configuration 19](#_Toc37345663)

[5.5.2 Editing Date Plane Configuration 20](#_Toc37345664)

[5.5.3 Editing Gx App Configuration 21](#_Toc37345665)

[5.5.4 DNS server setup for UPF selection 22](#_Toc37345666)

[5.5.4.1 Basic DNS Server Installation 22](#_Toc37345667)

[5.5.4.2 Update /etc/hostname and /etc/hosts 24](#_Toc37345668)

[5.6 Launching 24](#_Toc37345669)

[5.6.1 Running Control Plane, Data Plane and Gx Applications 24](#_Toc37345670)

[5.6.2 Bring Up the Data Plane Path 24](#_Toc37345671)

[6 Monitoring 25](#_Toc37345672)

[6.1 Using CLI 25](#_Toc37345673)

[6.2 Logging 25](#_Toc37345674)

[6.2.1 c3pocli for Logging 26](#_Toc37345675)

[7 Troubleshooting 27](#_Toc37345676)

[7.1 Binding PCI Interface to dpdk Driver 27](#_Toc37345677)

[7.2 Setting Environment for Control and Data Planes 27](#_Toc37345678)

[7.3 Data Not Passing Through Data Plane 27](#_Toc37345679)

[7.4 Giving Permission to Shell Script File 28](#_Toc37345680)

[7.5 Memory Issue in Section 4.3 Control Plane build 28](#_Toc37345681)

[8 Backlog 29](#_Toc37345682)

[9 References 29](#_Toc37345683)

# Acronyms and Definitions

**Table 1: Acronyms and Definitions**

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| 3GPP | 3rd Generation Partnership Project |
| APN | Access Point Name |
| C3PO | Clean CUPS Core for Packet Optimization –a code repository under OMEC-project |
| CDF | Charging Data Function |
| CDR | Charging Data Record |
| CLI | Command Line Interface |
| CTF | Charging Trigger Function |
| CUPS | Control and User Plane Separation – Provides architecture enhancements for the separation of control and user functionality in EPC |
| DNS | Domain Name System |
| EPC | Evolved Packet Core |
| LTE | Long Term Evolution |
| MME | Mobile Management Entity |
| NGIC | Next Generation Infrastructure Core –a code repository under OMEC-project |
| OMEC | Open Mobile Evolved Core – the first full-featured, scalable, high performance open source EPC |
| ONF | Open Networking Foundation –non-profit operator led consortium driving transformation of network infrastructure and carrier business models |
| OSS | Operations Support System |
| PCRF | Policy and Charging Rules Function |
| PDN | Packet Data Network |
| PGW-C | Packet Data Network Gateway – Control Plane – an EPC gateway controller that processes control signal messages for PGW-Us. |
| PGW-U | Packet Data Network Gateway – User Plane – an EPC gateway that provides connectivity from the UE to external packet data networks |
| rtc | Run to completion |
| SAE | System Architecture Evolution |
| SAEGW-C | System Architecture Evolution Gateway – Control Plane – an EPC gateway that is a combination of S-GW and P-GW control plane nodes |
| SAEGW-U | System Architecture Evolution Gateway – User Plane – an EPC gateway that is a combination of S-GW and P-GW user plane nodes. |
| SGW-C | Serving Gateway – Control plane – an EPC gateway controller that processes, routes, and forwards control signal packets for SGW-Us |
| SGW-U | Serving Gateway – User plane – an EPC gateway that routes and forwards user data packets |
| SGX | Software Guard Extensions – Intel’s protected environment that contains the code and data of a security-sensitive computation |
| TST | Technical Steering Team |
| UE | User Equipment |
| UPF | User Plane Function |

# Introduction

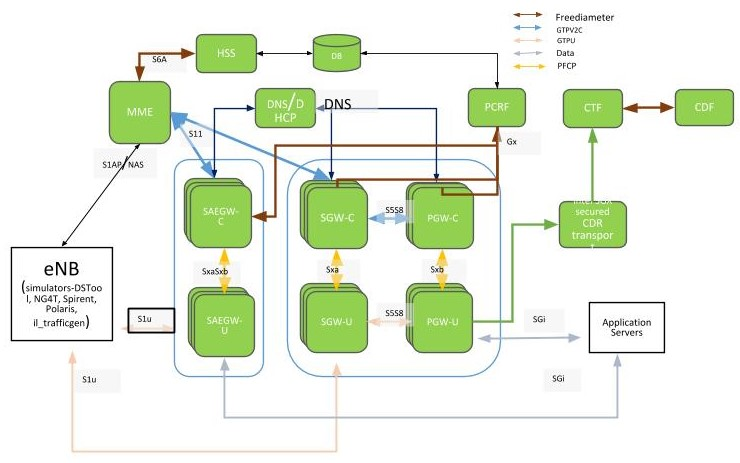
This user guide is for LTE EPC Gateways developers and users as part of OMEC (Open Mobile Enhanced Core). This guide is intended to help users build, configure and deploy OMEC with the supported Gateway combinations. This user guide also provides some technical insights into the architecture of the Gateway code and documents open issues and backlog.

# Background

The OMEC project is an open source LTE EPC development under Open Networking Foundation (ONF) consortium. Under OMEC, there are a suite of projects working together to create EPC solutions. The ‘ngic-rtc’ project provides Gateway components of the EPC.

Figure 1 shows the components under EPC. The ‘ngic-rtc’ project provides source code for the SGW-C, PGW-C, SGW-U and PGW-U components shown in Figure 1, and SAEGW-C and SAEGW-U components that are not shown in the figure.

This document is a user guide for ngic-rtc source code and Gateways combinations mentioned above. Throughout this document all gateway combinations (SGW-C, PGW-C, SGW-U, PGW-U, SAEGW-C and SAEGW-U) are commonly referred to as Gateways.

****

**Figure 1. EPC Architecture**

## Source Repository

The following are the source code repositories relevant for Gateway development, deployment and testing.

### ngic-rtc

The public repository for the Gateways is hosted at the following location and is maintained by OMEC Technical Steering Team (TST) members.

<https://github.com/omec-project/ngic-rtc>

Branch: e-utran-features

### oss\_util

The public repository for oss\_util is at <https://github.com/omec-project/oss-util>.

### Freediameter

The Public repository for freediameter is at <https://github.com/omec-project/freediameter>.

## License

The complete source code is under [**Apache 2.0**](https://www.apache.org/licenses/LICENSE-2.0) license. Any new open to be linked with must follow the license restriction and license contamination against Apache 2.0 has to be avoided.

The source code has gone through the following source control and license check process:

<License check process, contamination checks, blackduck etc.>

## Specifications Referred

All development was based on release 15 of the 3gpp specifications listed in Section 9.

## Release Updates

This document describes release version 1.6. The source code is available at the following source control:

<https://github.com/omec-project/ngic-rtc>

Branch: e-utran-features

### Release Details

#### New Features

Although release 1.6 is the first to be publicly available, prior releases were developed without being released. The new features included in release 1.6 are listed in Table 2.

**Table 2. New Features**

|  |  |
| --- | --- |
| **Feature** | **Reference** |
| UPF Selection by DNS | TS 23.003, TS 29.244, TS 23.401, TS 29.303 |
| PDN Initial Attach  EUTRAN UE Initial Detach  MME Initiated UE Detach Procedure  HSS Initiated UE Detach Procedure | TS 23.401 - 5.3.2.1-1, 5.10.2, 5.3.8.3-1, 5.3.8.4-1  TS 23.214 - 6.3.1.1, 6.3.1.6-1 |
| Dedicated Bearer Activation | TS 23.214 - 6.3.1.7-1 |
| Bearer Modification Procedure without Bearer QoS Update | TS 23.401 – 5.4.3-1  TS 23.214 – 6.3.1.7-1 |
| PDN GW Initiated Dedicated Bearer Deactivation  MME Initiated Dedicated Bearer Deactivation | TS 23.401 – 5.4.4.1-1, 5.4.4.2-1  TS 23.214 – 6.3.1.7 |
| UE Requested PDN Connectivity | TS 23.401 – 5.10.2-1  TS 23.214 – 6.3.1.1 |
| UE or MME Requested PDN Disconnection | TS 23.401 – 5.10.3-1  TS 23.214 – 6.3.1.6-1 |
| MME triggered SGW relocation | TS 23.214 - 6.3.1.2.2 |
| X2 handover SGW relocation | TS 23.214 - 6.3.1.2.2 |
| eNB F-TEIDu - X2 handover without SGW relocation | TS 23.401 - 5.5.1.1.2  TS 23.214 - 6.3.1.3 |
| 3D State Machine with Procedure | TS 23.401, TS 23.214, TS 29.274, TS 29.244, TS 29.212, TS 29.213 |
| Sxa, Sxb, SxaSxb - Association, establishment, modification, session release. | PFCP support  TS 29.244 |
| Gx support in SAEGW-C and PGW-C for:  CCR-I  CCA-I  CCR-T  CCA-T  RAR  RAA  CCR-U  CCA-U |  |
| libGTPv2C Auto generated code on S11 and S5/S8 Interfaces | TS 29.274 |
| libpfcp Auto generated code integration on SxaSxb, Sxa, Sxb Interfaces | PFCP support  TS 29.244 |
| Restoration support   * Support for FQ-CSID in the Gateways. * Restoration and cleanup for the control plane components (SGW-C, PGW-C, SAEGW-C) failures. | TS 23.007 – Section 17. |
| CLI and statistics parameters | CLI to collect statistical information on various interfaces, protocols and operations. |
| API error with cause values | TS 29.274, TS 29.244 |
| Peer echo and timers | TS 29.274, TS 29.244  T3, N3 timers |
| APN and Distributed Deployment Support   1. Ability of SGW-C to communicate with Multiple PGW-C with each PGW-C supporting different APN. 2. Ability of PGW-C / SAEGW-C to communicate with multiple SGW-C with each SGW-C serving different UE. |  |
| TS 29.303 based DNS routines to implement the agreed upon flows for the selection of UPF (SGW-U, PGW-U, SAEGW-U) located closest to eNB associated with the UE. |  |
| Configuration file modification | Removed hard coded peer configurations for MME, SGWs and PGWs. Other configuration simplification. |

#### Important Bug Fixes and Open Items

Although release 1.6 is the first to be publicly available, prior releases were developed without being released. The bugs fixed in release 1.6 are listed in Table 3. Note that the Bug ID is only meaningful to the developers of release 1.6.

**Table 3. Bugs Fixed**

|  |  |
| --- | --- |
| **Fixed #** | **Description** |
| 1 | Entry not found for UE error in the DP |
| 2 | No space left on device error when running a test case multiple times |
| 3 | Executing a test multiple times without restarting the GW starts producing a repeating error on the control plane |
| 4 | Losing packets on the downlink in combined GW test |
| 5 | The SPGWC process hangs |
| 6 | No Response from SGW-C during MME initiated delete bearer procedure |
| 7 | Create FAR Apply Action should be set to forward |
| 8 | CLI-PGW - test ID #13 - missing PFCP Session Modification Reply |
| 9 | Split Gateway multiple calls with multiple dedicated bearers data being lost in PGWU |
| 10 | Add development tools (TMUX, screen) to development images |
| 11 | Deployment fails filtering for NIC if it is not 10GbE |
| 12 | Enable the automated deployment to take the configuration file as a parameter |
| 13 | Error reporting in log files need to be consistent |
| 14 | TEID RI value in DP configuration takes value which is out of range without any error handling |
| 15 | PFCP association setup happens even when TEIDRI is not provided in dp\_config.cfg file |
| 16 | Echo response from SAEGWU on S1U do not have Recovery: 0 |
| 17 | [SAEGW\_DP\_Error] spgwu ngic\_dataplane - Failed to remove GTPU header |
| 18 | Auto deployment does not check core\_range to appropriate with hardware core mapping |
| 19 | r13 TAC query is delayed too long after the eNB query fails |
| 20 | SGX billing deployment fails |
| 21 | SGWC created new session with SGWU for retransmission of CSR. |

The issues and bugs that have been identified but not fixed in release 1.6 are listed in Table 4. Note that the Bug ID is only meaningful to the developers of release 1.6.

**Table 4. Open Items and Bugs**

|  |  |
| --- | --- |
| **Open #** | **Description** |
| 1 | OMEC user-plane does not respond with error-indication after path failure |
| 2 | OMEC control plane does not retry PFCP Association Setup Request |
| 3 | The SGWC gets into a state that can only be fixed by rebooting the VM |
| 4 | Longevity testing fails after a couple of hours with both the CP and DP aborting and restarting with the following errors |
| 5 | SGW-C Does not respond to Create Session Request when SGW-U is down |
| 6 | OMEC returns Mandatory IE missing for E-UTRAN initial attach with (ipv6) |
| 7 | Consistently getting: Failure to allocate memory for upf list structure error with test case when creating somewhere near 80 sessions |
| 8 | PGWU does not initiate Sx heartbeats to PGWC |
| 9 | SGW doesn't respond to S11 delete-session-request after a path failure on S5 |
| 10 | Implement Network time syncing in the VMs, and set the timezone to UTC |
| 11 | SGW-C Does not respond to Release Access Bearers Request from MME |
| 12 | OMEC does not support create session request during handovers with HO indication flags set during X2-based handover with S-GW change |
| 13 | S1-based handover is failing |
| 14 | OMEC creates new dedicated bearers instead of updating existing bearers |
| 15 | SGWC does not send Modify-Bearer-Response(reject) response in the first attempt at times |
| 16 | Deployment cleanup fails if VM is in shut off state at the target |
| 17 | SGWC does not send MBReq to PGWC during X2-based handover without Serving GW relocation and UE Time Zone changed |
| 18 | no reply from PGWC when a diameter Protocol/Transient/Permanent error is returned by PCRF |
| 19 | During inter MME Tracking Area Update procedure without S-GW change- SGW-C during terminating the session, sends DSReponse to Old MME |
| 20 | Combine the GW service and kni service on the DP nodes |
| 21 | SGWC responds with Delete-Session-Request Cause : Context Not Found (64) for existing session still undergoing replacement |
| 22 | While testing E-UTRAN attach with same imsi and apn is present on sgw-c, The PGW-C erroneously, throws error of default\_eps\_bearer\_qos is missing |
| 23 | OMEC doesn't respond to echo req from MME intermittently |
| 24 | OMEC does not respond to create session request with unknown apn |
| 25 | Need more detail in the Discarding packet due to gtp version is not supported messages |
| 26 | pgwc and sgwc do not respond with CSResponse & cause-code for context replacement and additional pdn scanerios |
| 27 | OMEC doesnt respond to Delete-Session-Request with wrong bearer-id |
| 28 | OMEC PGW sends Bearer-Identifier=35 in CCR-I and CCR-T |
| 29 | OMEC doesn't send create\_session\_response for default bearer creation intermittently |
| 30 | [Deployment] Secure deployment with integration with Vault(hashicorp) |
| 31 | [Deployment] Automate SGX deployment issue |
| 32 | Implementing Deployment using Remote Image repository |
| 33 | No console notifications when peers are down |
| 34 | TEID range returned after CP restart is not correct |
| 35 | Support for Dedicated bearer in Handover scenarios |
| 36 | Over time, the resetsecs and upsecs statistics deviate by quite a bit without resetting the stats |
| 37 | [Deployment] After killing the ngic process, DP itself restart service but manually need to start kni service. |
| 38 | [Deployment] Need support to resume capability in case of failure |
| 39 | GW Alarms and Warnings issues in deployment scripts |
| 40 | [Combined GW] Performance Test issue |
| 41 | [Combined GW] DP Logging Errors During Performance Test |

# Deployment Options

The ngic-rtc software can be configured to run in two deployment modes, Combined SGW-PGW Architecture and Split GW Architecture, as described below.

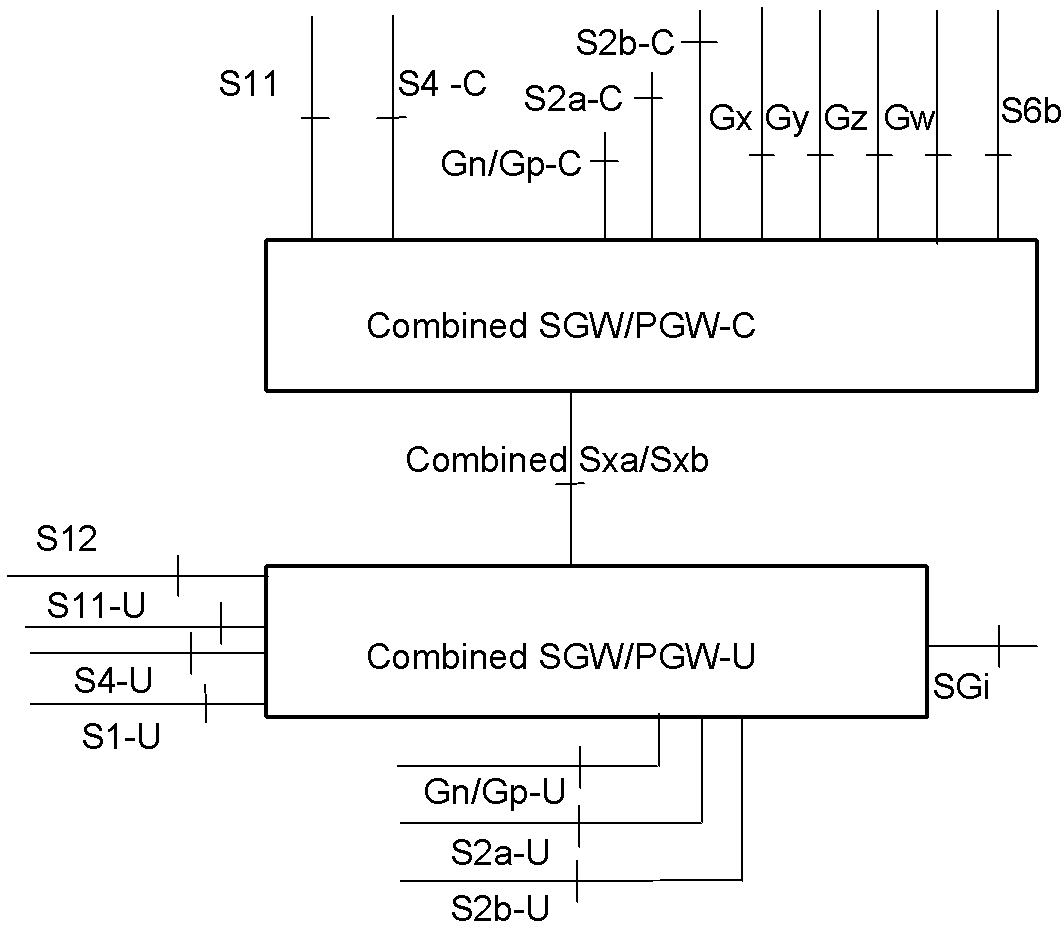
## Combined SGW-PGW Architecture

In the Combined SGW-PGW Architecture, a single instance of the control gateway operates as a combination of SGW and PGW for the control plane, and a single instance of user gateway operates as a combination of SGW and PGW for the user plane.

TS 23.214 section 4.2.2 describes the combined gateway architecture.

### Combined GW Deployment and Terminology

The deployment diagram, taken from TS 23.214, Section 4.2.2, is shown in Figure 2. The control plane shown as “Combined SGW/PGW-C” in the diagram is referred to as “SAEGW-C” throughout this document. The user plane shown as “Combined SGW/PGW-U” in the diagram is referred to as “SAEGW-U” throughout this document.

****

**Figure 2. Deployment diagram**

### Configuration Settings

The following configuration runtime setting is used to run ngic-rtc as SAEGW-C.

CP\_TYPE = 03

The following configuration settings are used to run ngic-rtc as SAEGW-U.

SPGW\_CFG=03

For the details of the setup configuration refer to Section 5.5.1.

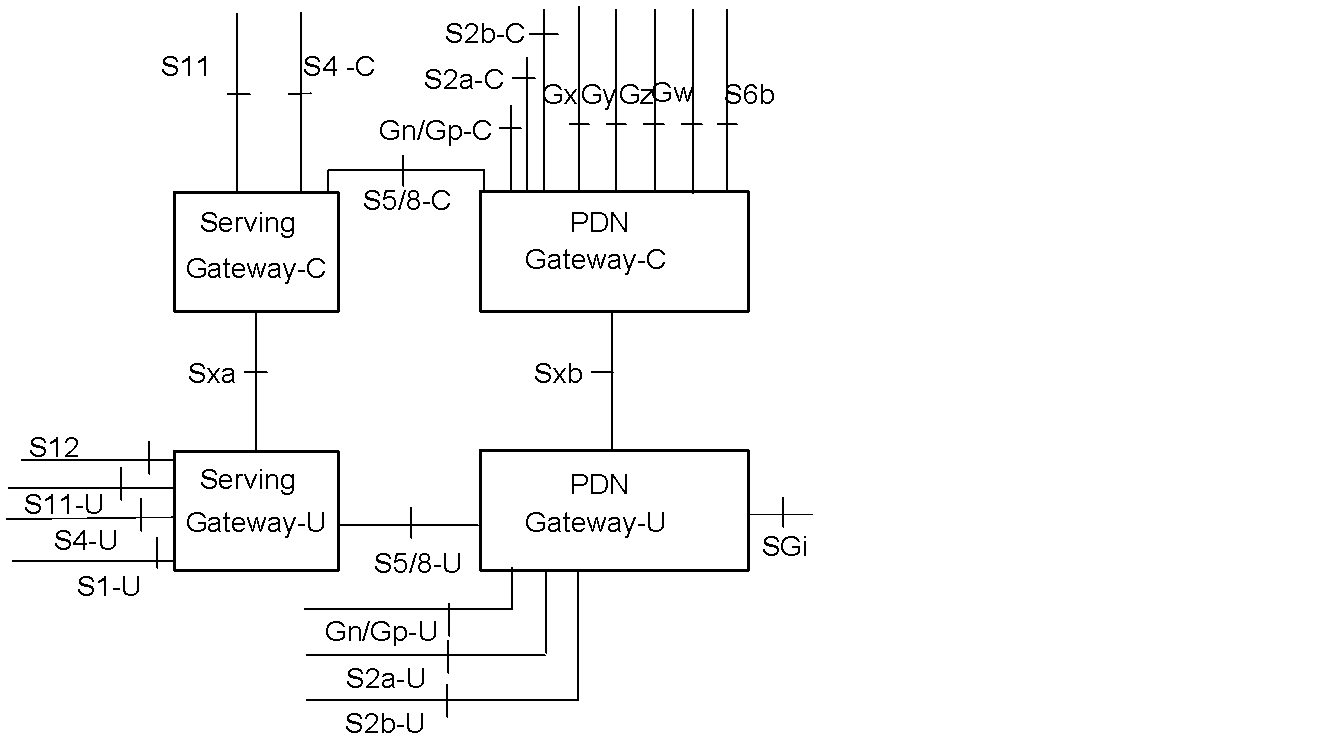
## Split GW Architecture

Operating as a split GW in the CUPS architecture, the ngic-rtc software can be deployed to run as four different types of instances: SGW-C, PGW-C, SGW-U and PGW-U.

TS 23.214 section 4.2.1 describes the split gateway architecture.

### Split GW Deployment and Terminology

The deployment diagram, taken from TS 23.214, Section 4.2.1, is shown in Figure 3. The control plane shown as “Serving Gateway-C” in the diagram is referred to as “SGW-C” throughout this document. The user plane shown as “Serving Gateway-U” in the diagram is referred to as “SGW-U” throughout this document. The control plane shown as “PDN Gateway-C” in the diagram is referred to as “PGW-C” throughout this document. The user plane shown as “PDN Gateway-U” in the diagram is referred to as “PGW-U” throughout this document.

****

**Figure 3 Deployment diagram**

### Configuration Settings

The following runtime configuration setting is used to run ngic-rtc as SGW-C.

CP\_TYPE = 01

The following configuration setting is used to run ngic-rtc as PGW-C.

CP\_TYPE = 02

The following configuration setting is used to run ngic-rtc as SGW-U.

SPGW\_CFG=01

The following configuration setting is used to run ngic-rtc as PGW-U.

SPGW\_CFG=02

Please refer to Section 5 for the details of overall configuration.

## Decommission

<TBD>

# Downloading, Building and Launching

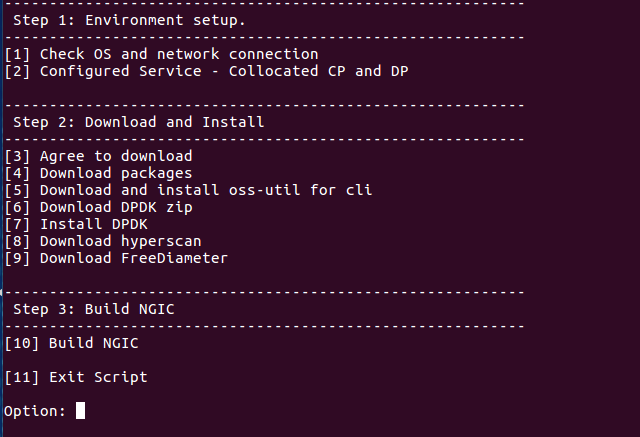
## Downloading

To download from github, execute the following.

*git clone <Git repo URL>*

## Installation of Control Plane and Data Plane

To install a control plane or user plane, run the *install.sh* script from the ngic-rtc folder. The *install.sh* script will provide the menu shown in Figure 4.

****

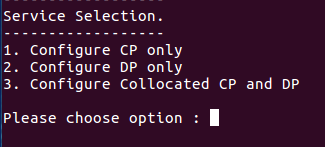
**Figure 4. Install.sh Menu**

### Control Plane Installation

Follow the steps below to install the control plane.

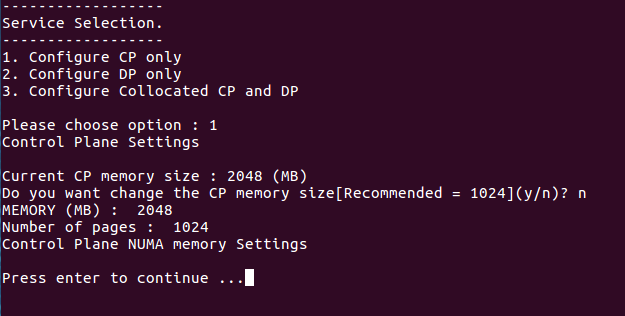
1. Enter “1” to obtain information about network connectivity and OS information
2. Enter “2” to configure control plane

After selecting option 2, the new selection menu will appear as shown in Figure 5.

****

**Figure 5. Service Selection Menu**

After entering “1” for control plane installation, the selection menu shown in Figure 6 will appear. Note that “n” has already been selected in the figure. If you want to change memory then enter “y”, otherwise enter “n”.

****

**Figure 6. Control Plane Setting Selection**

Note: The configuration of the combined SGW-PGW Architecture described in Section 4.1 is not supporting in release 1.6.

1. Enter “3” and agree to download dependent libraries or packages. Enter “y” after selecting option “3”.
2. Enter “4” to download packages.
3. Enter “5” to be redirected to the oss-util installation.

Follow each step sequentially.

1. Enter “6” to download the DPDK zip file.
2. Enter “7” to install the Data Plane Development Kit.
3. Enter “8” to download the FreeDiameter open source library.
4. Enter “9” to Build NGIC.

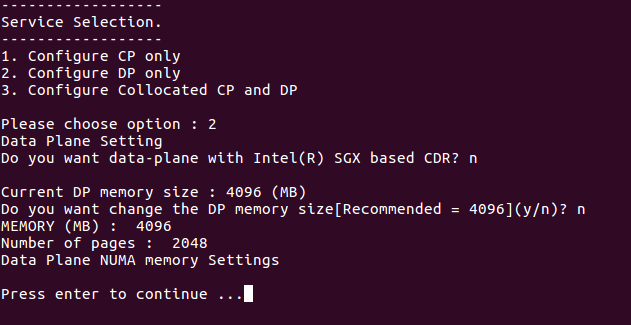
This option builds and links all the libraries used to make the final binary.

### Data Plane Installation

Follow the steps below to install the user plane.

1. Enter “1” to obtain information about network connectivity and OS.
2. Enter “2” to configure the data plane server.

After selecting option 2, the new selection menu will appear as shown in Figure 5. Enter “2” for the data plane.

****

**Figure 7. Data Plane Setting Selection**

After entering “1” for control plane installation, the selection menu shown in Figure 6 will appear. Note that “n” has already been selected in the figure. If you want to change memory then enter “y”, otherwise enter “n”.

After entering “2” for data plane installation, the selection menu shown in Figure 7 will appear. Note that “n” has already been selected in the figure. If you want to change memory then enter “y”, otherwise enter “n”.

Note: The configuration of the combined SGW-PGW Architecture described in Section 4.1 is not supporting in release 1.6.

1. Enter “3” and agree to download dependent libraries or packages. Enter “y” after selecting option “3”.
2. Enter “4” to download packages.
3. Enter “5” to be redirected to the oss-util installation.

Follow each step sequentially.

1. Enter “6” to download the DPDK zip file.
2. Enter “7” to install the Data Plane Development Kit.
3. Enter “8” to download hyperscan packages.
4. Enter “9” to Build NGIC.

This option builds and links all the libraries used to make the final binary.

## Building

The build options below help the user build the code manually without using *./install.sh* script every time. Once setup is complete as section 4.2, the following code changes and build steps can be followed.

1. Control plane

Path: *ngic-rtc/cp*

Command: *make clean; make;*

1. Data plane

Path: *ngic-rtc/dp*

Command: *make clean; make;*

1. Oss-util

Path: *ngic-rtc/oss\_adapter/c3po\_oss/oss-util/*

Command: *make clean; make; make install;*

Note: After this step, go to step 1 or 2.

1. Build libpfcp library

Path: *ngic-rtc/libpfcp*

Command: *make clean; make;*

Note: After this step, go to step 1 or 2 according to the requirements of the server.

1. Build libgtpv2c library

Path: *ngic-rtc/libgtpv2c*

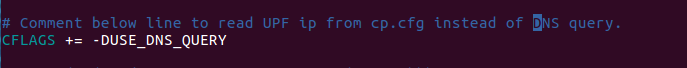
Command: *make clean; make;*

Note: After this step, go to step 1 or 2 according to the requirements of the server.

### Runtime and Compile Time Flags

1. DNS FLAG

Path: *ngic-rtc/cp/Makefile*

****

Function: Disable DNS flag to discover Data Plane Server statically

Flag Type: Compile Time

1. CP AND DP LOGGING FLAG

Path: *ngic-rtc/config/cp.cfg*

Path: *ngic-rtc/config/dp\_config.cfg*

****

****

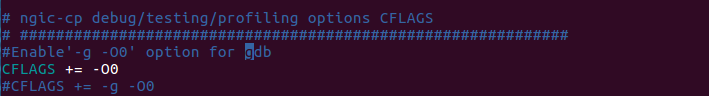
Function: Activate Logging

Flag Type: Run Time

1. GDB FLAG

Path: *ngic-rtc/cp/Makefile*

Path: *ngic-rtc/dp/Makefile*

****

Function: uncomment the flag to enable debug mode and use ./run.sh debug command to activate debugging when run the Control Plane Server and Data Plane Server.

Flag Type: Compile Time

1. GX FLAG

Path: *ngic-rtc/cp/Makefile*

****

Function: Remove Gx functionality if comment the flag

Flag Type: Compile Time

1. PERFORMANCE FLAG

Path: *ngic-rtc/dp/Makefile*

****

Function: Remove Log Levels if uncomment the flag

Flag Type: Compile Time

## Dependencies

1. **Ubuntu 16.04 LTS** should be installed. The Ubuntu 16.04 LTS image can be downloaded from <https://www.ubuntu.com/download/alternative-downloads>
2. The User account on the server must have **root privileges**.
3. Peer components, MME, DNS, PCRF, eNB, and SGi-AS, or their simulators must be installed and configured.
4. The following packages must be installed by install.sh:

* DPDK version 16.11.4
* build-essential
* linux-headers-generic
* git
* unzip
* libpcap-dev
* make
* hyperscan
* curl
* openssl-dev
* freediameter
* Pistache
* rapidjson
* spdlog
* cpp-driver
* c-ares
* and any other library dependencies

## Configuration

The interfaces configured are listed in Table 5.

**Table 5. Interfaces**

|  |  |
| --- | --- |
| **Interface** | **Description** |
| MGMT | mgmt network address space |
| DNS | DNS network address space |
| S1MME | s1mme network address space |
| S6A | s6a network address space |
| DB | Database network address space |
| S11 | network address space |
| SGWC\_S5S8 | s5s8 control plane (sgw) network address space |
| PGWC\_S5S8 | s5s8 control plane (pgw) network address space |
| FPCNB | fpc nb network address space |
| FPCSB | fpc sb network address space |
| S1U | s1u network address space |
| SGWU\_S5S8 | s5s8 user plane (sgw) network address space |
| PGWU\_S5S8 | s5s8 user plane (pgw) network address space |
| SGI | sgi network address space |
| CTF | ctf network address space |
| CDF | cdf network address space |

An example default configuration for the ngic interfaces is given below. This configuration leverages the networks or IPs defined in the configuration.

[NETWORKS]

EX\_MGMT="10.31.14.0/24"

MGMT="192.168.124.0/24"

S1MME="10.2.1.0/24"

S11="10.2.2.0/24"

DNS="192.168.122.0/24"

S6A="10.2.3.0/24"

DB="10.2.4.0/24"

S5S8\_SGWC="10.2.5.0/24"

S5S8\_SPGWC="10.2.5.0/24"

S5S8\_PGWC="10.2.5.0/24"

S5S8\_SGWU="10.2.6.0/24"

S5S8\_PGWU="10.2.6.0/24"

FPCNB="10.2.7.0/24"

SXA="10.2.8.0/24"

SXB="10.2.8.0/24"

FPCSB="10.2.8.0/24"

GX="10.2.10.0/24"

S1U="11.9.1.0/24"

SGI="13.9.1.0/24"

CTF="10.214.93.0/24"

CDF="10.214.93.0/24"

### Editing Control plane Configuration

The parameters in the *ngic-rtc/config/cp.cfg* file for the control plane are shown in Table 6.

**Table 6. Control Plane Parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Value** |
| CP\_TYPE | This value is per gateway It defines type of deployment for the gateway | SGWC = 01, PGWC = 02, SAEGWC = 03 |
| S11\_IP | SGW-C S11 interface IP | IPV4 |
| S11\_PORT | GTPV2 Port number | 2123 |
| S5S8\_IP | SGW-C s5s8 interface ip if SGW and PGW-C s5s8 interface ip if PGW | IPV4 |
| S5S8\_PORT | GTPV2 Port number | 2123 |
| PFCP\_IP | SX interface ip of SGW-C/SAEGW-C | IPV4 |
| PFCP\_PORT | PFCP protocol port | 8805 |
| MME\_S11\_IP | MME S11 interface ip | IPV4 |
| MME\_S11\_PORT | GTPV2 Port number | 2123 |
| UPF\_PFCP\_IP | Need to select data plane path when DNS is disable | IPV4 |
| UPF\_PFCP\_PORT | PFCP protocol port number | 8805 |
| REQUEST\_TIMEOUT | Wait time for response from peer | Value in milliseconds 1 to 1800000 |
| REQUEST\_TRIES | Max Request Tries if peer is not responding | 1 to 20 |
| CP\_LOGGER | Used to activate logging | 0 or 1 |
| APN\_CONFIG | Add apn Configuration value | apn |
| APP nameserver | Ip address of app server for DNS query | IPV4 |
| OPS nameserver | Ip address of ops server for tac and apn based query | IPV4 |
| IP\_POOL\_CONFIG | Add ip pool configuration for UE ip allocation | IPV4 |

### Editing Date Plane Configuration

The data plane configuration files can be edited using the following.

1. Edit the *ngic-rtc/config/dp\_config.cfg* file

* Change the value of SPGW\_CFG as per gateway deployment mode
* Change the value of ip, mac address, and port according to the configured interface on the machine for flow of data packets

1. Edit the *ngic-rtc/config/interface.cfg* file

* Change the value of dp\_comm\_ip parameter and cp\_comm\_ip parameter in *ngic-rtc/config/interface.cfg*

Note: You must bind the data plane interfaces to the dpdk driver before running the server. See <https://doc.dpdk.org/guides/tools/devbind.html>.

### Editing Gx App Configuration

Edit the configuration files to run Gx applications.

1. As shown in Figure 8 below, edit the gx.conf file available at:

*ngic-rtc/cp/gx\_app/gx.conf*

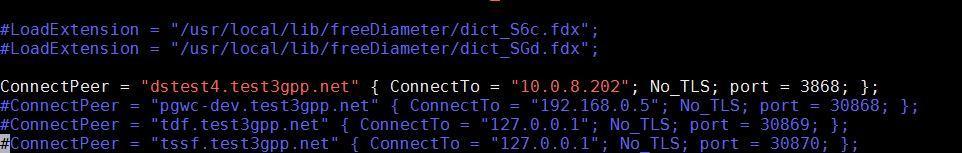
Hostname: pgwc6 (Modify in Identity,TLS\_Cred)

Realm: test3gpp.net (Identity, Realm)

****

**Figure 8. gx.conf File**

1. Modify the PCRF <Hostname>.<Realm> {connectTo= “PCRF IPV4 ”; NO\_TLS,port = 3868} as shown in Figure 9

****

**Figure 9 gx.conf File**

1. As shown in Figure 10, generate the certificate file with command using

*./make\_certs.sh <hostname> <Realm>*

****

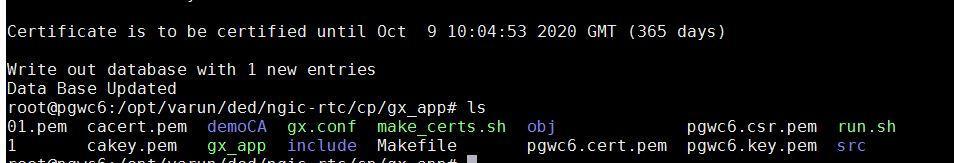
**Figure 10. Certificate Make Command**

The output will be:

<Hostname>.cert.pem

<Hostname>.csr.pem

<Hostname>.key.pem

****

**Figure 11. Certificate Make Console Result**

More information can be found at <http://www.freediameter.net/trac/wiki/Configuration>.

### DNS server setup for UPF selection

#### Basic DNS Server Installation

Execute following steps on the host where DNS is to be installed.

1. Install the required DNS packages

*sudo apt-get install bind9 dnsutils bind9-doc*

1. Set bind to IPV4 mode by update the options parameter in */etc/default/bind9* as follows:

OPTIONS="-4 -u bind"

1. Modify */etc/bind/named.conf*.options to define the DNS servers to forward unresolved DNS request to. For example to forward unresolved DNS requests to Google, */etc/bind/named.conf.options* should look as follows:

options {

directory "/var/cache/bind";

// If there is a firewall between you and nameservers you want

// to talk to, you may need to fix the firewall to allow multiple

// ports to talk. See http://www.kb.cert.org/vuls/id/800113

// If your ISP provided one or more IP addresses for stable

// nameservers, you probably want to use them as forwarders.

// Uncomment the following block, and insert the addresses replacing

// the all-0's placeholder.

forwarders {

8.8.8.8;

};

//========================================================================

// If BIND logs error messages about the root key being expired,

// you will need to update your keys. See https://www.isc.org/bind-keys

//========================================================================

dnssec-validation auto;

auth-nxdomain no; # conform to RFC1035

listen-on-v6 { any; };

};

1. Configure Local File. Modify */etc/bind/named.conf.local* as follows:

* Add the forward zone with the following lines (substitute the zone name with your own):

zone "test3gpp.net" {

type master;

file "/etc/bind/db.test3gpp.net";

};

* Add the reverse zone by adding the following lines (note that the reverse zone name starts with 93.212.10 which is the opposite of 10.212.93.10).

zone "93.212.10in-addr.arpa" {

type master;

notify no;

file "/etc/bind/db.10";

};

1. Create the forward zone file using the file name referenced in */etc/bind/named.conf.local*. In this example the file is named */etc/bind/db.test3gpp.net* and the DNS server (ns1).

; BIND data file for local loopback interface

;

$TTL 604800

@ IN SOA ns1.test3gpp.net. root.ns1.test3gpp.net. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS ns1.test3gpp.net.

@ IN A 127.0.0.1

@ IN AAAA ::1

1. Create the reverse zone file using the file name referenced in etc*/bind/named.conf.local*. In this example the file is named */etc/bind/db.10*

;

; BIND reverse data file for local loopback interface

;

$TTL 604800

@ IN SOA ns1.test3gpp.net. root.ts1.test3gpp.net. (

1 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS ns1.test3gpp.net.

1. Start the DNS server.

*sudo service bind9 start*

1. Verify that the DNS service started successfully by opening /var/log/syslog and look for the following entry toward the end of the file:

Aug 10 13:20:29 ns1 named[29932]: all zones loaded

If this message cannot be located, locate the error message associated with the “named” service and make the necessary corrections.

#### Update /etc/hostname and /etc/hosts

1. Modify */etc/hostname* to reflect the unqualified host name of each server. In this example, the DNS server hostname should be:

ns1

1. Ensure that */etc/hosts* has entries for 127.0.0.1, 127.0.1.1 and the IP address of the host with both the fully qualified hostname and the unqualified hostname.

127.0.0.1 localhost

127.0.1.1 ns1.test3gpp.net ns1

10.212.93.10 ns1.test3gpp.net ns1

## Launching

Prerequisite for running the following component is that EPC setup is up and running with DNS and PCRF as per configured in the Gateway configuration files.

Please refer DNS configuration guide under following repository location for the DNS settings.

<DNS guide path at EPCTools>

### Running Control Plane, Data Plane and Gx Applications

1. Control plane path: *ngic-rtc/cp*
2. Data plane path: *ngic-rtc/dp*
3. Gx app path: *ngic-rtc/cp/gx\_app*

Run the *run.sh* script in each path above to run all servers.

### Bring Up the Data Plane Path

Use the Kni script to bring up the data plane path.

Kni script path: - *ngic-rtc/kni\_ifcfg*

1. *kni-S1Udevcfg.sh* is used to bring up the S1U interface data plane path for data packets.
2. *kni-S5S8devcfg.sh* is used to bring up the S5S8 interface in case of a split gateway for flow of data packets.
3. *kni-SGIdevcfg.sh* is used to bring up the SGI interface data plane path for data packets.

# Monitoring

The Control Plane and Data Plane can be controlled, monitored and tuned through the command line. Appropriate logs are available for each and can be viewed through the command line. Details are provided in the following sections.

## Using CLI

Run the CP or DP as SAEGW-C/U, SGWC/U, PGWC/U using *./run.sh* script in ngic-rtc folder.

Use the readme file to set c3pocli environment *ngic-rtc/oss\_adapter/c3po\_oss/oss-util/cli/README.TXT.*

Once CLI is set up with the above steps, the command *c3pocli* activates the CLI.

The c3pocli command can be used with several arguments as illustrated below:

1. c3pocli[http://127.0.0.1:12997](http://127.0.0.1:12997/) stats describe-stats-live

Gives live statistics of all peers in JSON format.

1. c3pocli http://127.0.0.1:12997 stats describe-stats-all

Shows all messages on each interface.

## Logging

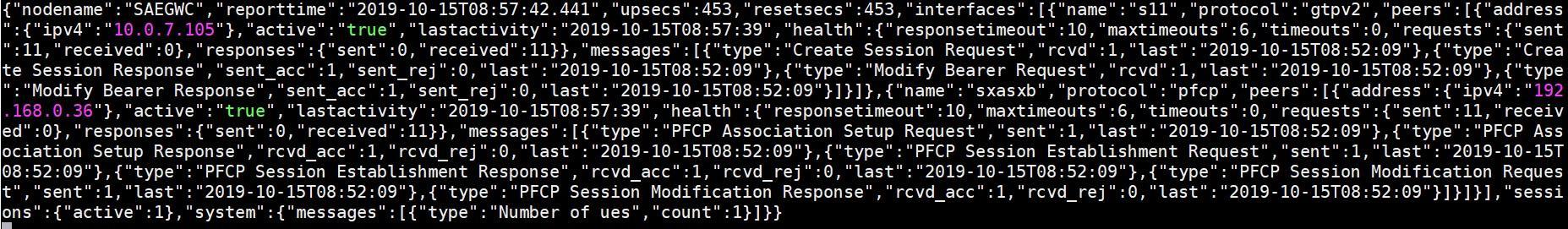
Logs are maintained in the following 3 files under *ngic-rtc/cp/logs* folder or *ngic-rtc/dp/logs*.

1. Console messages are logged into the cp.log/dp.log file as shown in Figure 12.

****

**Figure 12. cp.log/dp.log file**

1. Stats after each 5 seconds (stat frequency set to 5 seconds by default) are logged into the *cp\_stat.log/dp\_stat.log* file in JSON object format as shown in Figure 13.

****

**Figure 13. cp\_stat.log/dp\_stat.log File**

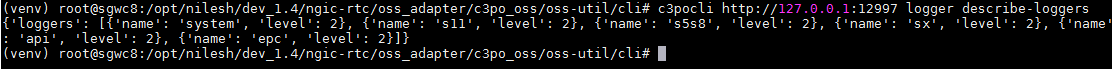
1. Log level with major, minor, and critical are logged into the *cp\_sys.log/dp\_sys.log* file shown in Figure 14.



**Figure 14 cp\_sys.log/dp\_sys.log File**

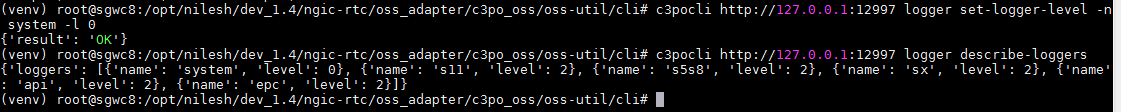
### c3pocli for Logging

1. c3pocli <http://127.0.0.1:12997> *logger describe-loggers* returns a JSON string with the current loggers and their log levels as shown below.



1. c3pocli  [http://127.0.0.1:12997](http://127.0.0.1:12997/) logger set-logger-level -n system -l 0

This command updates the log level for the specified logger name as shown below. Valid log levels are: Trace=0, Debug=1, Info=2, major=3, minor = 4 and critical=5.

****

1. c3pocli http://127.0.0.1:12997 stats set-stats-logging -n all

This command updates the stats logging mode in “all” level, Show all messages on each interface. Interfaces appear based on Gateway type. Ex. If SGW-C then “S11, S5S8 and Sxa” interface. By Default, “suppress” mode is configured.

****

1. c3pocli http://127.0.0.1:12997 stats set-stats-logging -n suppress

This command updates the stats logging mode to suppress mode for save data in file. In “suppress”mode, we show messages which arrive on the interface (S11, S5S8 etc).

1. c3pocli [http://127.0.0.1:12997](http://127.0.0.1:12997/statfreq) stats describe-stats-frequency

This command shows the information about frequency of writing content in the cp\_stat.log/dp\_stat.log file.

1. c3pocli [http://127.0.0.1:12997](http://127.0.0.1:12997/) stats set-stats-frequency -f 2000

This command sets the statistics frequency. The value is in milliseconds.

# Troubleshooting

## Binding PCI Interface to dpdk Driver

Bind the S1u/Sgi port to DPDK drivers

1. Command to get PCI address is:

|  |
| --- |
| root@spgwu:/opt/ngic-rtc/dpdk/usertools# lshw -c network -businfo  Bus info Device Class Description  ========================================================  pci@0000:00:03.0 ens3 network Virtio network device  pci@0000:00:04.0 ens4 network Virtio network device  pci@0000:00:08.0 ens5 network 82599ES 10-Gigabit SFI/SFP+ Network Connection  pci@0000:00:09.0 ens6 network 82599ES 10-Gigabit SFI/SFP+ Network Connection  ... |

1. Bind the port using the PCI id

|  |
| --- |
| cd /opt/ngic-rtc/dpdk/usertools/  ./dpdk-devbind.py -b igb\_uio 00:08.0 |

1. Lists ports

|  |
| --- |
| root@spgwu:/opt/ngic-rtc/dpdk/usertools# ./dpdk-devbind.py --status  Network devices using DPDK-compatible driver  ============================================  0000:00:08.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' drv=igb\_uio unused=ixgbe  Network devices using kernel driver  ===================================  0000:00:03.0 'Virtio network device 1000' if=ens3 drv=virtio-pci unused=igb\_uio \*Active\*  0000:00:04.0 'Virtio network device 1000' if=ens4 drv=virtio-pci unused=igb\_uio \*Active\*  0000:00:09.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection 10fb' if=ens6 drv= ixgbe unused=igb\_uio  … |

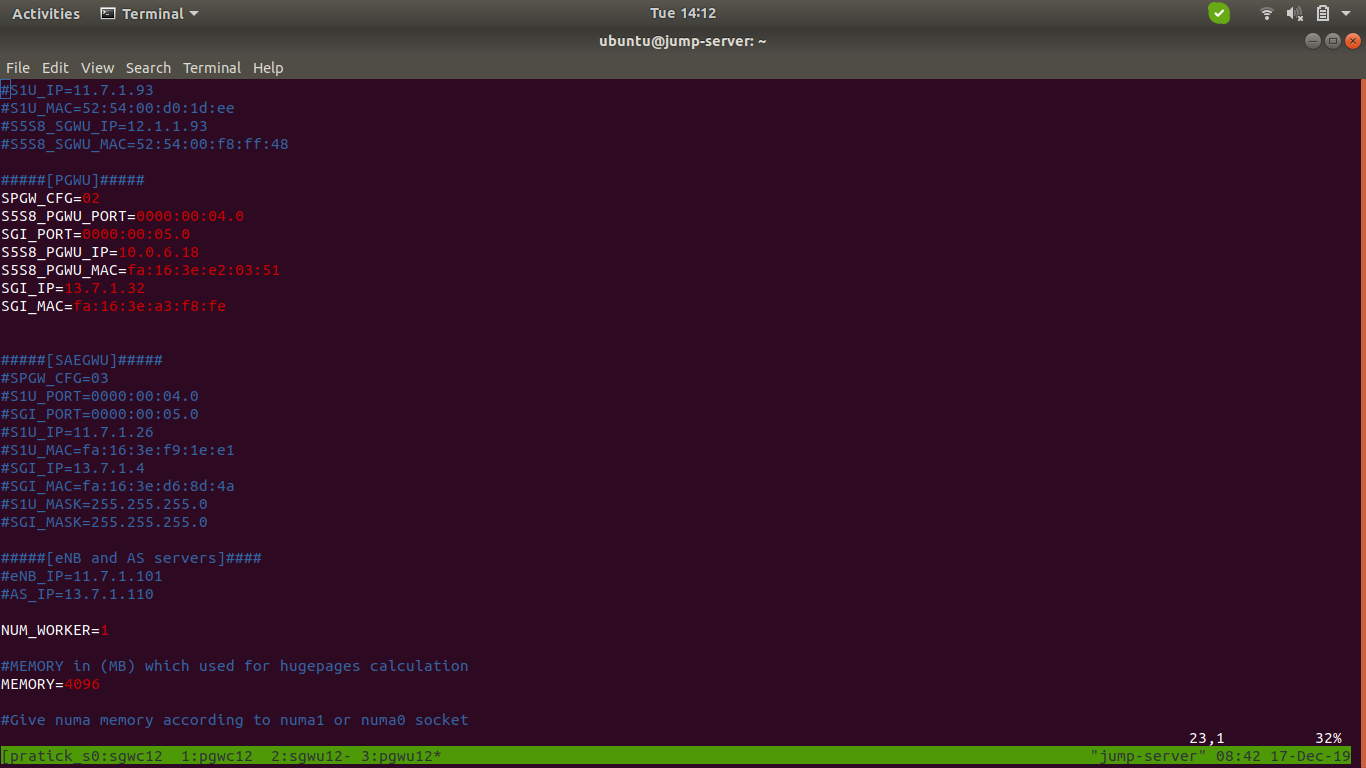
## Setting Environment for Control and Data Planes

Path: - cd *ngic-rtc/*

Command: *source setenv.sh*

## Data Not Passing Through Data Plane

1. Check PGWU/SAEGWU/SGWU for *ngic-rtc/config/dp\_config.cfg* configuration file. See Figure 15.
2. Check the MAC address and IP Configuration

****

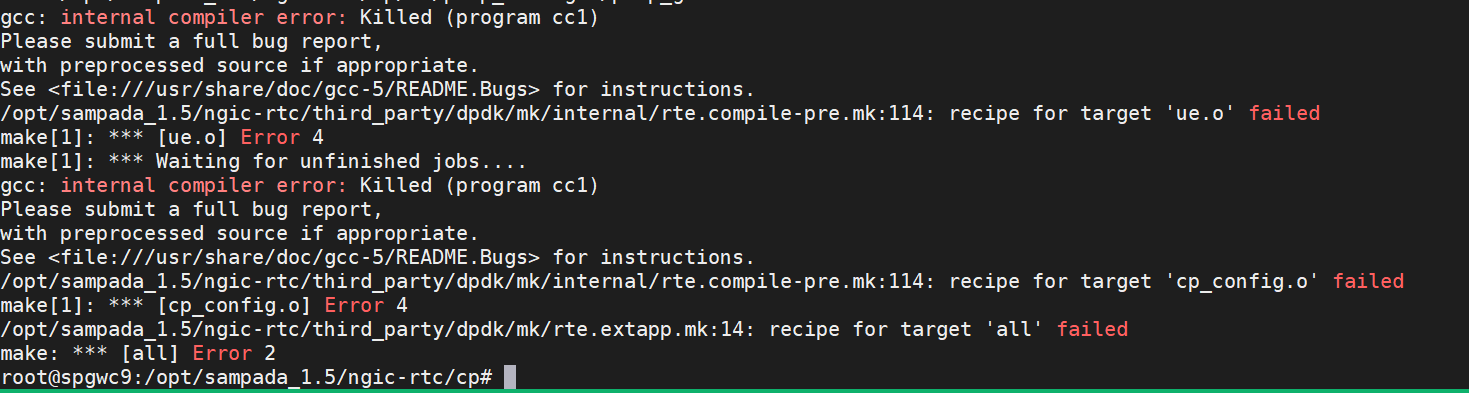
**Figure 15. dp\_config.cfg File**

## Giving Permission to Shell Script File

command: *chmod +x < filename >*

## Memory Issue in Section 4.3 Control Plane build

If memory issues arise during the control plane build described in Section 5.3 (See Figure 16), perform the steps below.

****

**Figure 16. Memory Issue**

1. Go to *ngic-rtc/* path
2. Run the source *setenv.sh* Command
3. *make -j5 build-cp*

# Backlog

Table 7 contains the feature backlog for release. 1.6. Most of the features in the backlog are in the development plan for release 1.7.

**Table 7. Feature Backlog**

|  |  |
| --- | --- |
| **Feature** | **Details** |
| IPv6 support | IPv6 and IPv4v6 dual mode support |
| Gx events |  |
| CDR generation | Usage records information passing to control plane as per specification 29.244. |
| Lawful Interception |  |
| Restoration procedure – for user plane | Handling of Partial or complete failure, recovery as per specification 23.007 |
| 23.401 flows | S1 handover  X2 Handover  S1 Handover  TAU update with/without SGW change  Secondary RAT usage reporting  ERAB Modification  Connection suspend/resume |
| Multiple PDN connections support |  |
| Handover scenarios | In case of SAEGW (combined mode) deployment. |
| SAEGW to PGW Promotion and Demotion for a PDN connection. |  |

# References

1. 3GPP specifications release 15.
   1. TS 23.401
   2. TS 23.214
   3. TS 23.272
   4. TS 23.003
   5. TS 23.007
   6. TS 29.212
   7. TS 29.244
   8. TS 29.303
   9. TS 29.212
   10. TS 29.213
2. Open Networking Foundation <https://www.opennetworking.org/>