User Guide

**Control Plane, User Plane Gateways;**

**LTE EPC;**

**Installation Guide;**

**Troubleshooting Guide;**

**(Release 1.5)**

Table of Contents

[1 Acronyms and Definitions 4](#_Toc36222841)

[2 Introduction 5](#_Toc36222842)

[3 Background 5](#_Toc36222843)

[3.1 Source Repository 5](#_Toc36222844)

[3.1.1 ngic-rtc 6](#_Toc36222845)

[3.1.2 oss\_util 6](#_Toc36222846)

[3.1.3 Freediameter 6](#_Toc36222847)

[3.2 License 6](#_Toc36222848)

[3.3 Specifications Referred 6](#_Toc36222849)

[3.4 Release Updates 6](#_Toc36222850)

[3.4.1 Release Details 6](#_Toc36222851)

[4 Deployment Options 9](#_Toc36222852)

[4.1 Combined SGW-PGW Architecture 10](#_Toc36222853)

[4.1.1 Combined GW Deployment and Terminology 10](#_Toc36222854)

[4.1.2 Configuration Settings 10](#_Toc36222855)

[4.2 Split GW Architecture 10](#_Toc36222856)

[4.2.1 Split GW Deployment and Terminology 11](#_Toc36222857)

[4.2.2 Configuration Settings 11](#_Toc36222858)

[4.3 Decommission 12](#_Toc36222859)

[5 Downloading, Building and Launching 12](#_Toc36222860)

[5.1 Downloading 12](#_Toc36222861)

[5.2 Installation of Control Plane and Data Plane 12](#_Toc36222862)

[5.2.1 Control Plane Installation 12](#_Toc36222863)

[5.2.2 Data Plane Installation 14](#_Toc36222864)

[5.3 Building 15](#_Toc36222865)

[5.3.1 Runtime and Compile Time Flags 15](#_Toc36222866)

[5.4 Dependencies 16](#_Toc36222867)

[5.5 Configuration 17](#_Toc36222868)

[5.5.1 Editing Control plane Configuration 18](#_Toc36222869)

[5.5.2 Editing Date Plane Configuration 19](#_Toc36222870)

[5.5.3 Editing gx App Configuration 19](#_Toc36222871)

[5.6 Launching 21](#_Toc36222872)

[5.6.1 Running Control Plane, Data Plane and gx Applications 21](#_Toc36222873)

[5.6.2 Bring Up the Data Plane Path 21](#_Toc36222874)

[6 Monitoring 22](#_Toc36222875)

[6.1 Using CLI 22](#_Toc36222876)

[6.2 Logging 22](#_Toc36222877)

[6.2.1 c3pocli for Logging 23](#_Toc36222878)

[7 Troubleshooting 24](#_Toc36222879)

[7.1 Binding PCI Interface to dpdk Driver 24](#_Toc36222880)

[7.2 Setting Environment for Control and Data Planes 24](#_Toc36222881)

[7.3 Data Not Passing Through Data Plane 24](#_Toc36222882)

[7.4 Giving Permission to Shell Script File 25](#_Toc36222883)

[7.5 Memory Issue in Section 4.3 Control Plane build 25](#_Toc36222884)

[8 Backlog 26](#_Toc36222885)

[9 References 26](#_Toc36222886)

# 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 |  |
| 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 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.

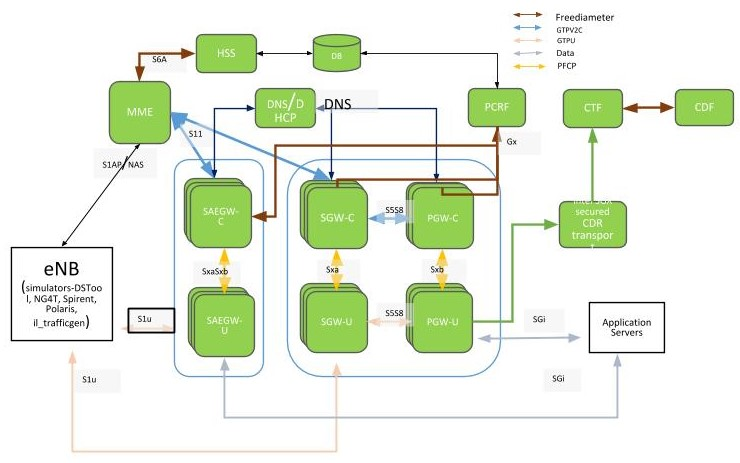


Figure 1. EPC Architecture

## 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>

### 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 1.5. The source code is available at the following source control:

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

Branch: <ngic\_rtc>

### Release Details

#### New Features

Although release 1.5 is the first to be publicly available, prior releases were developed without being released. The new features included in release 1.5 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 | TS 23.401 - 5.3.2.1-1, 5.10.2  TS 23.214 - 6.3.1.1 |
| Dedicated Bearer Activation | TS 23.214 - 6.3.1.7-1 |
| 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 |
| 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 |
| 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 |
| Configuration file modification | Removed hard coded peer configurations for MME, SGWs and PGWs. Other configuration simplification. |

#### Important Bug Fixes and Open Items

Although release 1.5 is the first to be publicly available, prior releases were developed without being released. The bugs fixed in release 1.5 are listed in Table 3. Note that the Bug ID is only meaningful to the developers of release 1.5.

Table 3. Bugs Fixed

| **Fixed #** | **Description** |
| --- | --- |
| 1 | SGW-C created a new session with SGW-U for retransmission of CSR. |
| 2 | SGX billing deployment fails |
| 3 | Over time, the resetsecs and upsecs statistics deviate by quite a bit without resetting the stats |
| 4 | [SAEGW\_DP\_Error] spgwu ngic\_dataplane - Failed to remove GTP-U header |
| 5 | Echo response from SAEGW-U on S1U do not have Recovery: 0 |
| 6 | Support for Dedicated bearer in Handover scenarios |
| 7 | PFCP association setup happens even when TEIDRI is not provided in dp\_config.cfg file |
| 8 | TEID RI value in DP configuration takes value which is out of range without any error handling |
| 9 | TEID range returned after CP restart is not correct |
| 10 | Error reporting in log files need to be consistent |
| 11 | Enable the automated deployment to take the configuration file as a parameter |
| 12 | Deployment fails filtering for NIC if it is not 10GbE |
| 13 | Add development tools (TMUX, screen) to development images |
| 14 | Split Gw multiple calls with multiple dedicated bearers data being lost in PGW-U |
| 15 | No console notifications when peers are down |
| 16 | CLI-PGW - test ID #13 - missing PFCP Session Modification Reply |
| 17 | OMEC PGW sends Bearer-Identifier=35 in CCR-I and CCR-T |
| 18 | Need more detail in the Discarding packet due to gtp version is not supported messages |
| 19 | Create FAR Apply Action should be set to forward |
| 20 | During inter MME Tracking Area Update procedure without S-GW change- SGW-C during terminating the session, sends DSReponse to Old MME |
| 21 | The SPGWC process hangs |
| 22 | Losing packets on the downlink in combined GW test |
| 23 | OMEC creates new dedicated bearers instead of updating existing bearers |
| 24 | Executing a test multiple times without restarting the GW starts producing a repeating error on the control plane |
| 25 | No space left on device error when running a test case multiple times |

The issues and bugs that have been identified but not fixed in release 1.5 are listed in Table 4. Note that the Bug ID is only meaningful to the developers of release 1.5.

Table 4. Open Items and Bugs

| **Open #** | **Description** |
| --- | --- |
| 1 | CP - Received delete session on non-existent EBI & Dropping packet" |
| 2 | [VM Performance Degradation] - Unexcepted packet loss running NG40 test at higher PPS |
| 3 | Packet loss in uplink path |
| 4 | SPGW-C becomes unavailable |
| 5 | UE Requested Dedicated Bearer Resource Allocation fails |
| 6 | UE Requested Bearer Resource Modification with TFT change fails |
| 7 | Some CDR files are missing aftre load run (got 49 CDR for 25 UE sessions) |
| 8 | [Split Gateway] GTPV2C\_CAUSE\_CONTEXT\_NOT\_FOUND when using unexpected GTP Message ModifyBearer Request |
| 9 | [Split Gateway] GTPV2C\_CAUSE\_CONTEXT\_NO when using Unexpected GTP Message DeleteSession Request |
| 10 | [Split-GW] Uplink Packet Loss |
| 11 | [Combined GW] DP Logging Errors During Performance Test |
| 12 | [Combined GW] Performance Test |
| 13 | [Split Gateway] Packet Loss during Performance Test |
| 14 | GW Alarms and Warnings |
| 15 | [Deployment] Need support to resume capability in case of failure |
| 16 | r13 TAC query is delayed too long after the eNB query fails |
| 17 | [Deployment] After killing the ngic process, DP itself restart service but manually need to start kni service. |
| 18 | Auto deployment does not check core\_range to appropriate with hardware core mapping |
| 19 | Implementing Deployment using Remote Image repository |
| 20 | [Deployment] Automate SGX deployment |
| 21 | [Deployment] Secure deployment with integration with Vault(hashicorp) |
| 22 | OMEC doesn't send create\_session\_response for default bearer creation intermittently |
| 23 | OMEC doesnt respond to Delete-Session-Request with wrong bearer-id |
| 24 | pgwc and sgwc do not respond with CSResponse & cause-code for context replacement and additional pdn scanerios |
| 25 | OMEC does not respond to create session request with unknown apn |
| 26 | OMEC doesn't respond to echo req from MME intermittently |
| 27 | While testing E-UTRAN attach with same imsi and apn is present present on sgw-c, The PGW-C erroneously, throws error of default\_eps\_bearer\_qos is missing |
| 28 | SGW-C responds with Delete-Session-Request Cause : Context Not Found (64) for existing session still undergoing replacement |
| 29 | Combine the GW service and kni service on the DP nodes |
| 30 | no reply from PGW-C when a diameter Protocol/Transient/Permanent error is returned by PCRF |
| 31 | No Response from SGW-C during MME initiated delete bearer procedure |
| 32 | SGW-C does not send MBReq to PGW-C during X2-based handover without Serving GW relocation and UE Time Zone changed |
| 33 | Deployment cleanup fails if VM is in shut off state at the target |
| 34 | SGW-C does not send Modify-Bearer-Response(reject) response in the first attempt at times |
| 35 | S1-based handover is failing |
| 36 | OMEC does not support create session request during handovers with HO indication flags set during X2-based handover with S-GW change |
| 37 | SGW-C Does not respond to Release Access Bearers Request from MME |
| 38 | Implement Network time syncing in the VMs, and set the timezone to UTC |
| 39 | SGW doesn't respond to S11 delete-session-request after a path failure on S5 |
| 40 | PGWU does not initiate Sx heartbeats to PGW-C |
| 41 | Consistently getting: Failure to allocate memory for upf list structure error with test case when creating somewhere near 80 sessions |
| 42 | OMEC returns Mandatory IE missing for E-UTRAN initial attach with (ipv6) |
| 43 | SGW-C Does not respond to Create Session Request when SGW-U is down |
| 44 | Longevity testing fails after a couple of hours with both the CP and DP aborting and restarting with the following errors |
| 45 | The SGW-C gets into a state that can only be fixed by rebooting the VM |
| 46 | Entry not found for UE error in the DP |
| 47 | OMEC control plane does not retry PFCP Association Setup Request |
| 48 | OMEC user-plane does not respond with error-ind after path failure |

# 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.

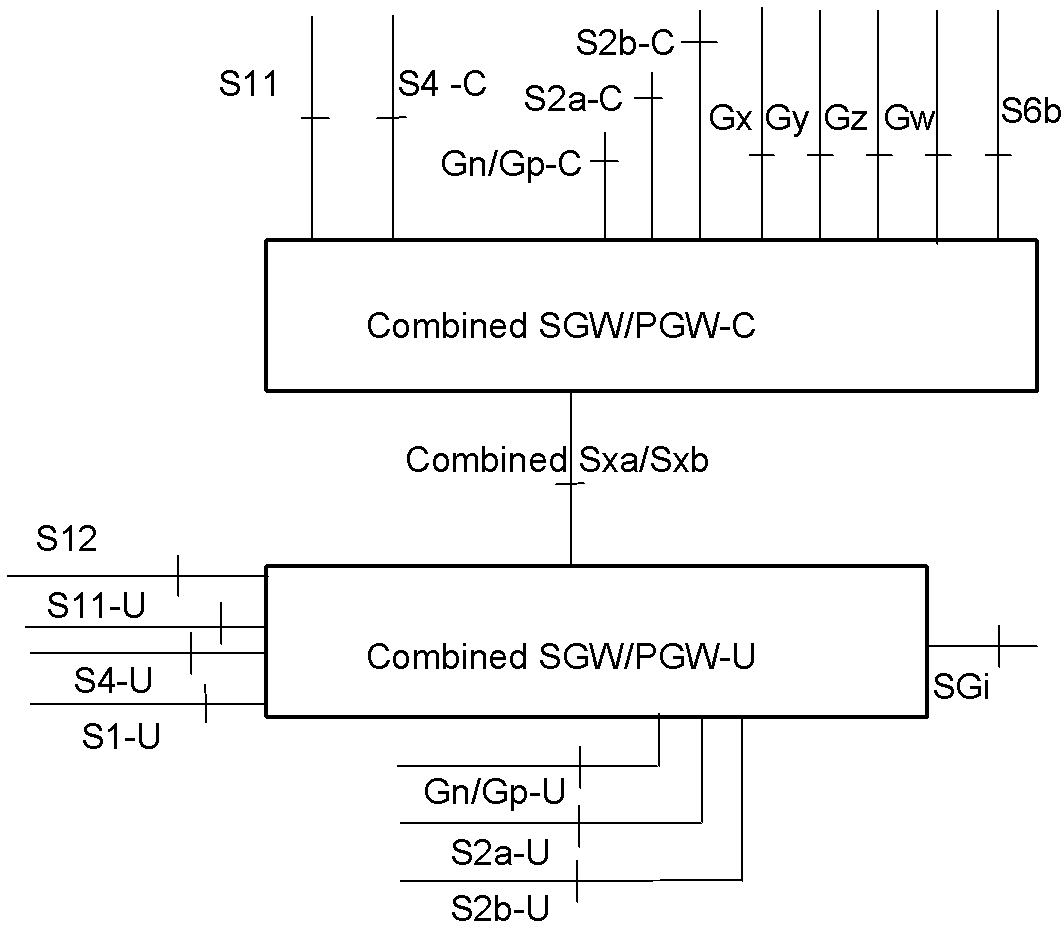


Figure 2. Deployment diagram

### Configuration Settings

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

CP\_TYPE = 03

The following configuration settings is 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.

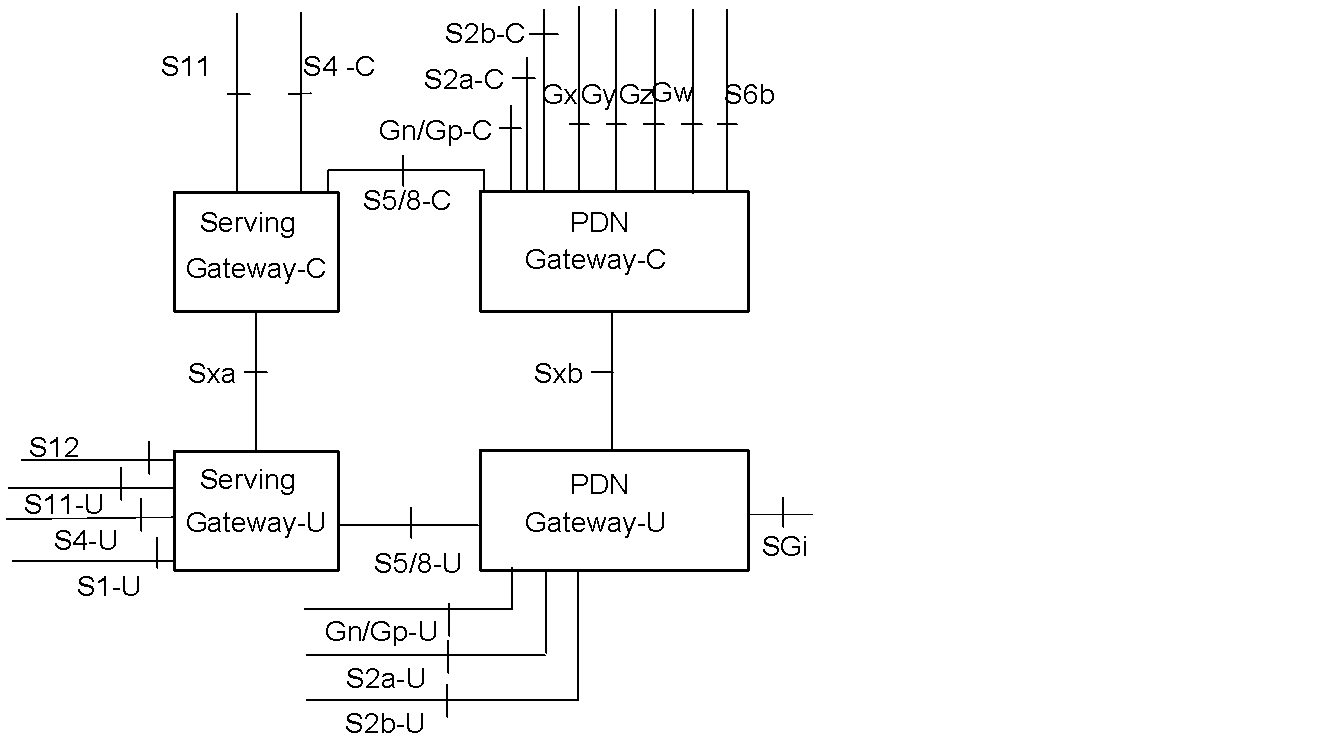


Figure 3 Deployment diagram

### 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-folder. The install.sh script will provide the menu shown in Figure 4.

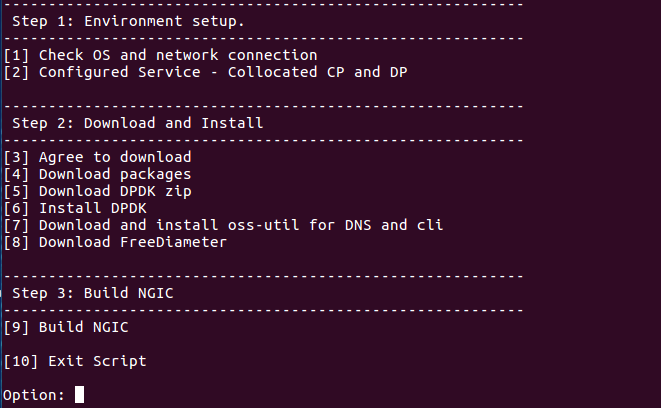


Figure 4. Install.sh Menu

### 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.

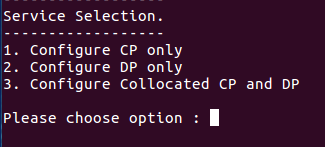


Figure 5. Service Selection Menu

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”.

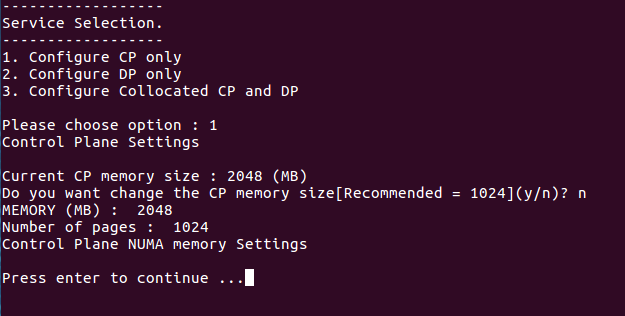


Figure 6. Control Plane Setting Selection

Note: The configuration of the combined SGW-PGW Architecture described in Section 4.1 is not supporting in release 1.5.

1. Enter “3” and agree to download dependent library or packages. Enter “y” after selecting option “3”.
2. Enter “4” to download packages.
3. Enter “5” to download DPDK zip file.
4. Enter “6” to install the Data Plane Development Kit.
5. Enter “7” to be redirected to the oss-util and dns installation script.

Follow each step sequentially.

1. Enter “8” to download the FreeDiameter open source library.
2. 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 data plane server.

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

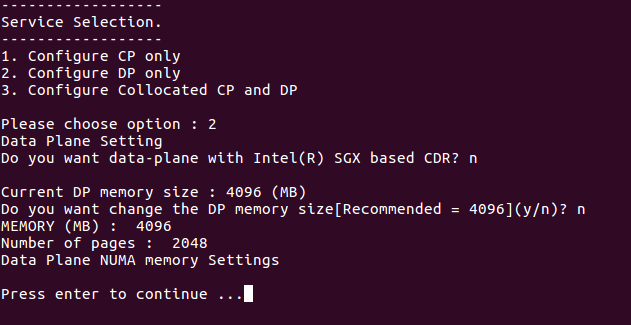


Figure 7. Data Plane Setting Selection

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 1.5.

1. Enter “3” and agree to download dependent library or packages. Enter “y” after selecting option “3”.
2. Enter “4” to download packages.
3. Enter 5 to download the DPDK zip file.
4. Enter 6 to install the Data Plane Development Kit.
5. Enter 7 to download hyperscan packages.
6. Enter 8 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: cd ngic-rtc/cp

Command: make clean; make;

1. Data plane

Path: cd ngic-rtc/dp

Command: make clean; make;

1. Oss-util

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

Command: make clean; make; make install;

Note: After this step, go to step 1.

1. Build libpfcp library

Path: cd 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: cd ngic-rtc/libgtpv2c

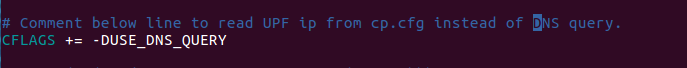
Command: make clean; make;

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 LOGGING FLAG

Path: ngic-rtc/config/cp.cfg

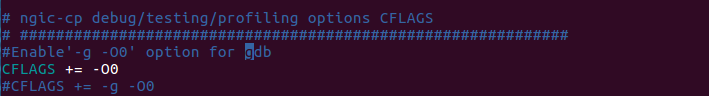


Function: Activate Logging

Flag Type: Run Time

1. GDB FLAG

Path: ngic-rtc/cp/Makefile



Function: uncomment the flag to enable debug mode and use ./run.sh debug command to activate debugging when run the Control 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 | db 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 gatewayy | 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 |
| 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

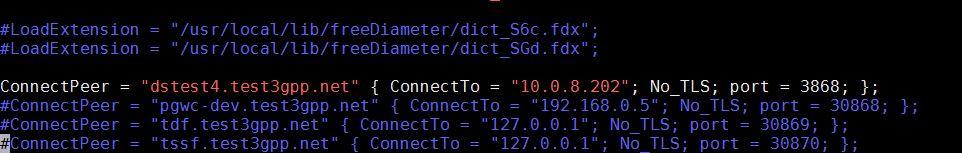


Figure 9 gx.conf File

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

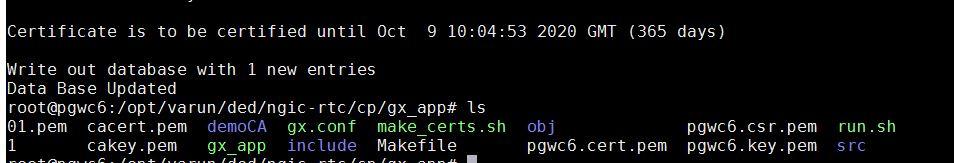


Figure 11. Certificate Make Console Result

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

## 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.

### 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 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 SGW-C, PGW-C, and SAEGW-C 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 setup with 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.

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



Figure 12. cp.log file

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

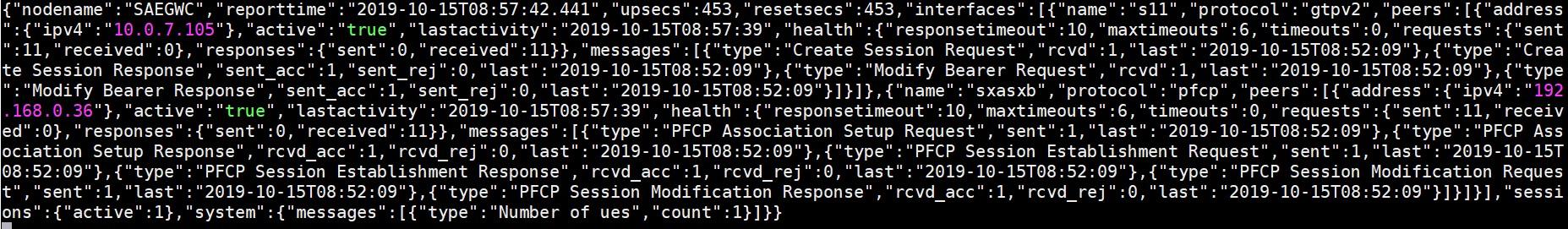


Figure 13. cp\_stat.log File

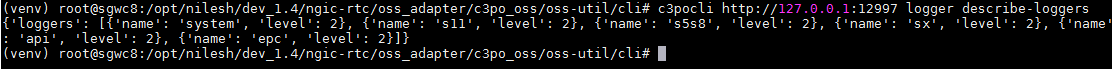
1. Log level with major, minor, and critical are logged into the *cp\_sys.log* file shown in Figure 14.



Figure 14 cp\_sys.log File

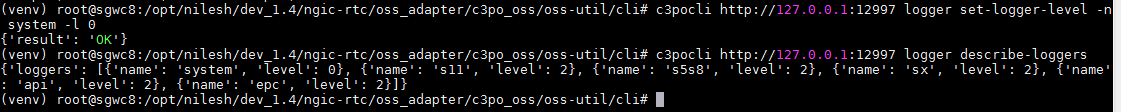
### c3pocli for Logging

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



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

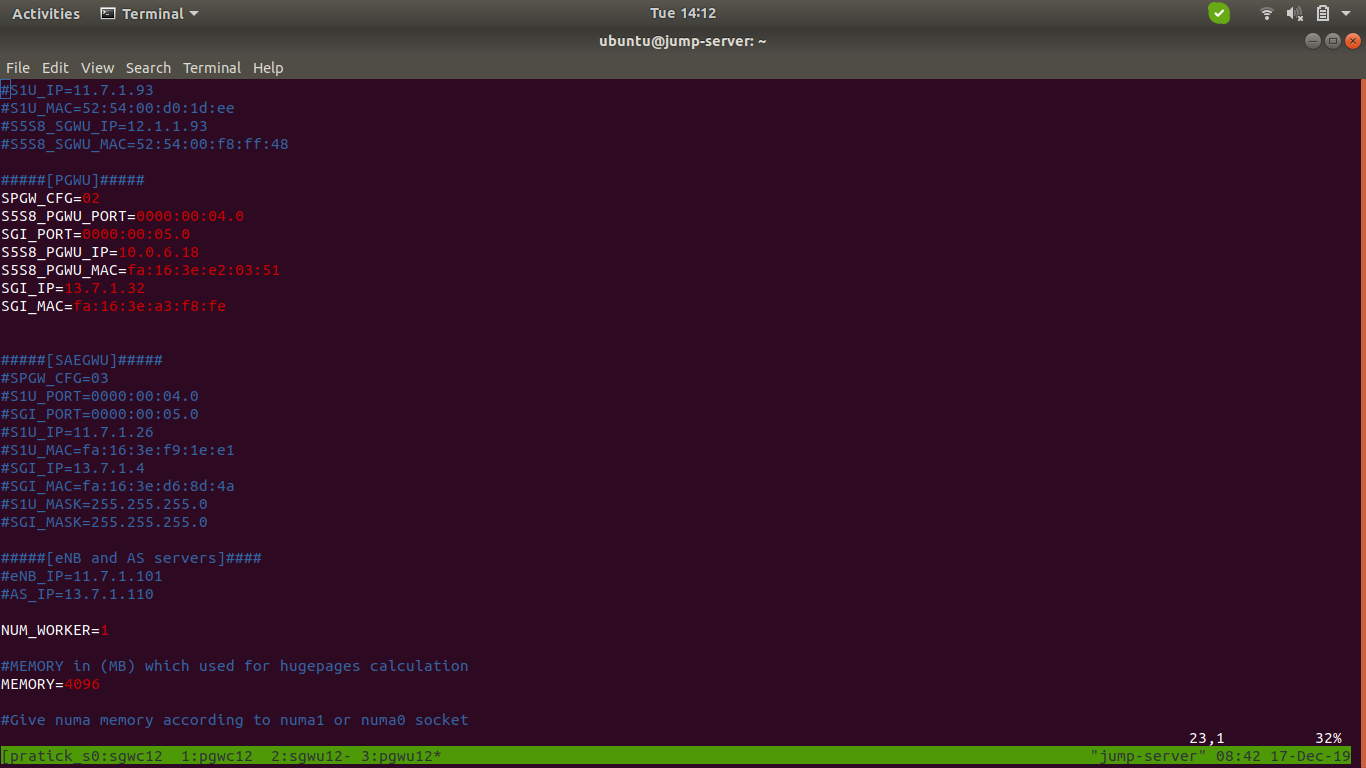


Figure 15. dp\_config.cfg File

## 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.

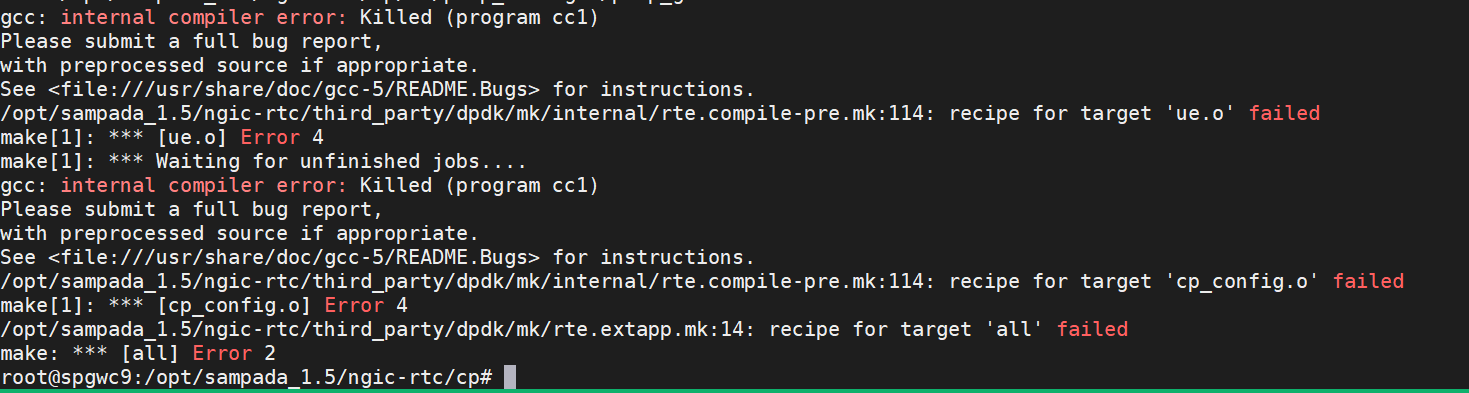


Figure 16. Memory Issue

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. 1.5. Most of the features in the backlog are in the development plan for release 1.6.

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 <> |
| Lawful Interception |  |
| Restoration procedure | 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  Bearer Modification/Deactivation  Connection suspend/resume |
| Multiple PDN connections support |  |
| Diameter messages | CCR-U, CCA-U |
| Handover scenarios |  |
| 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/>