OCUDU User Manual

Generated 2026-09-05 from 58fa6ca

User Documentation

This section covers everything you need to deploy and operate OCUDU. The User Manual and Tutorials take you from installation through to a running gNB; Integrations covers third-party hardware and cores; Migration covers porting existing srsRAN Project work to OCUDU; Releases tracks current status and what is coming next.

User Manual

Install, configure, and operate OCUDU in a live deployment. Covers installation, running, configuration, outputs, and troubleshooting.

Tutorials

Step-by-step guides for specific use cases: COTS UEs, O-RAN RUs, Near-RT RIC, Kubernetes, and more.

Integrations

Integration guides for third-party 5G cores, O-RAN radio units, switches, and timing devices.

Migrating from srsRAN

Port existing srsRAN Project modifications to OCUDU. Covers rebasing, diffing, and porting changes across subsystems.

Releases & Roadmap

Current release status, supported features, known limitations, and the development roadmap.

Installation

To download and build OCUDU:

1. Install dependencies
2. Install RF driver (only required for Split 8 deployments)
3. Clone the repository
4. Build the codebase

> **NOTE**
>
> OCUDU requires a Linux-based OS with realtime kernel support.

> **COMING FROM SRSRAN PROJECT?**
>
> Complete this installation first, then follow the Migration Guide to port your existing srsRAN Project modifications to OCUDU.

Build Tools and Dependencies

OCUDU uses CMake and C++17. Recommended build tools:

- cmake
- gcc (v15.2.0 or later) **OR** Clang (v14.0.0 or later)

OCUDU requires these dependencies:

- libsctp
- yaml-cpp
- mbedTLS
- A FFT library
- Optional requirement: googletest
  - GoogleTest is only mandatory when building with tests. You can enable test building by using the cmake option -DBUILD_TESTING=On.

Install build tools and dependencies:

```
sudo apt-get install cmake make gcc g++ pkg-config libmbedtls-dev libsctp-dev libyaml-cpp-dev libgtest-dev
```

These packages are optional but recommended:

- Ccache: This will help to speed up re-compilation
- backward-cpp: This library helps to generate more informative backtraces in the stdout if an error occurs during runtime

---

FFT Library

OCUDU requires one of the following libraries for FFT calculation:

- libfftw
- oneMKL
- AOCL-FFTZ
- Arm Performance Libraries

libfftw

```
sudo apt-get install libfftw3-dev
```

oneMKL

```
sudo apt update
sudo apt install -y gpg-agent wget
wget -O- https://apt.repos.intel.com/intel-gpg-keys/GPG-PUB-KEY-INTEL-SW-PRODUCTS.PUB | gpg --dearmor | sudo tee /usr/share/keyrings/oneapi-archive-keyring.gpg > /dev/null
echo "deb [signed-by=/usr/share/keyrings/oneapi-archive-keyring.gpg] https://apt.repos.intel.com/oneapi all main" | sudo tee /etc/apt/sources.list.d/oneAPI.list
sudo apt update
sudo apt install intel-oneapi-mkl-devel libomp-dev
```

AOCL-FFTZ

```
AOCL_FFTZ_VERSION="5.2"
sudo apt update && sudo apt install -y wget autoconf automake make libtool
```

```
cd /tmp
wget --no-check-certificate -O - "https://github.com/amd/aocl-
fftz/archive/refs/tags/${AOCL_FFTZ_VERSION}.tar.gz" | tar -xz
cd aocl-fftz-${AOCL_FFTZ_VERSION}
cmake -B buildFFTZ
cmake --build buildFFTZ --target install -j"${nproc}"
```

ARMPL (for ARM architectures)

```
sudo apt update && sudo apt install -y environment-modules wget
cd /tmp
wget --no-check-certificate -O - https://developer.arm.com/-/cdn-
downloads/permalink/Arm-Performance-Libraries/Version_24.10/arm-performance-
libraries_24.10_deb_gcc.tar | tar -x
cd arm-performance-libraries_24.10_deb/
./arm-performance-libraries_24.10_deb.sh --accept
cd ~ && rm -Rf /tmp/arm-performance-libraries_24.10_deb
source /usr/share/modules/init/bash
export MODULEPATH=$MODULEPATH:/opt/arm/modulefiles
module avail
module load armpl/24.10.0_gcc
```

RF-drivers

> **NOTE**
>
> UHD and/or ZMQ are only required for Split 8 deployments, if you are planning on using a Split 7.2 deployment you may skip this step.

OCUDU uses RF drivers to support different radio types. Currently, only UHD and ZMQ are supported:

- UHD (v4.6.0.0)
- ZMQ

Clone and Build

OCUDU supports several build configurations, controlled via CMake flags. The tabs below cover the available options.

Clone the OCUDU repository:

```
git clone https://gitlab.com/ocudu/ocudu.git
```

Then build the code-base:

```
cd ocudu
mkdir build
cd build
cmake ../
make -j $(nproc)
make test -j $(nproc)
```

The binary is at `ocudu/build/apps/gnb/`. To install system-wide:

```
sudo make install
```

See Running OCUDU for configuration and next steps.

Vector Tests

Most PHY components, as well as a few components from other layers, are tested by injecting vectors of input data and comparing the results with vectors of expected output data. Since data vectors can be quite heavy, vector tests are not included in the main repository and, instead, are offered as an external plugin that is part of the OCUDU MATLAB companion repository. All the details for installing and running the vector tests are explained in our MATLAB integration tutorial.

Running

> **INFO**
>
> This guide outlines how to *manually* run OCUDU applications in a Split 8 deployment with a USRP. For Split 7.2 deployments and/or Kubernetes execution see here.

Baseline Requirements

To successfully run an end-to-end network with OCUDU applications you will need the following:

- A PC with a Linux based OS (Ubuntu 22.04 or later)
- A USRP device
- OCUDU (see the Installation Guide)
- A 3rd-party 5G core (we recommend Open5GS)
- A 3rd-party 5G UE

> **TIP**
>
> If you plan to connect the gNB to a COTS UE best performance is achieved using an external clock source such as an Octoclock or GPSDO that is compatible with your RF-frontend, as the on-board clock within the USRP may not be accurate enough to enable a connection with the UE. This is discussed further in the relevant tutorial.

System Preparation

Before running any of OCUDU applications, we recommend tuning your system for best performance. We provide a script to configure known performance parameters:

- OCUDU performance script

The script does the following:

1. Sets the scaling governor to performance
2. Disables DRM KMS polling
3. Tunes network buffers (Ethernet based USRPs only)

Run the script as follows from the main project folder:

```
sudo ./scripts/ocudu_performance
```

Running OCUDU

If OCUDU has been installed using `sudo make install` or installed from packages then you will be able to run the gNB from anywhere on your machine.

If you have built OCUDU from source and have not installed it, then you can run the gNB from: `ocudu/build/apps/gnb`. In this folder you will find the gNB application binary.

Run the gNB as follows, passing the YAML configuration file:

```
sudo ./gnb -c gnb_rf_b200_tdd_n78_20mhz.yml
```

Run the gNB with `sudo` to ensure threads are configured with the correct priority.

Example configuration files can be found in the `configs/` folder in OCUDU codebase. For more information on the configuration files and the available parameters see the configuration reference.

When running, the gNB should generate the following console output:

```
Available radio types: uhd.

--== OCUDU gNB (commit 050a2bb72e) ==--

[INFO] [UHD] linux; GNU C++ version 9.4.0; Boost_107100; UHD_4.2.0.HEAD-0-g197cdc4f
Making USRP object with args 'type=b200'
Cell pci=1, bw=10 MHz, dl_arfcn=632628 (n78), dl_freq=3489.42 MHz, dl_ssb_arfcn=632640,
ul_freq=3489.42 MHz

==== gNodeB started ===
Type <t> to view trace
```

Entering `t` will enable the console trace, see here for more details.

Configuration parameters can also be passed on the command line. To see the list of options, use:

```
./gnb --help
```

For more information on running OCUDU, and configuring for various use-cases see the full list of tutorials.

Configuration Reference

OCUDU uses a YAML (.yml) configuration file.

The gNB comes with a number of example configuration files, these can be found in the `configs/` folder in the source files.

These configuration file examples provide a basic set-up to get users up and running, users can easily modify these to suit their use-case.

These include sample supplementary configuration files for:

- MIMO with a USRP
- Increasing QAM from 64 to 256
- Specific QOS configurations for voice, video, IMS, live-streaming and buffered video streaming
- Slicing support

OCUDU allows configuration files to be used in concatenation with other configuration files, using the `-c` flag between each file. For example you might want to use both the n310 and MIMO example configs together as follows:

```
sudo ./gnb -c gnb_rf_n310_fdd_n3_20mhz.yml -c mimo.yml
```

Furthermore you may wish to use QAM256 in addition to the above. This can be done as follows:

```
sudo ./gnb -c gnb_rf_n310_fdd_n3_20mhz.yml -c mimo.yml -c qam256.yml
```

Note: list-of-structures sections such as `cells` and `qos` cannot be merged across files due to CLI11 limitations. The last file to define the section takes effect.

Configuration Reference

All configuration parameters are presented here below in the following format:

```
parameter: default_value         # Optional/Required TYPE (default value). Parameter
description. Format: <format description> OR Supported: <supported values>.
```

DOWNLOAD CONFIGURATION FILES

Download the full reference configuration for each deployment mode: gNodeB (gnb.yaml) | CU (cu.yaml) | DU (du.yaml)

gnb.yaml

```yaml
# SPDX-FileCopyrightText: Copyright (C) 2021-2026 Software Radio Systems Limited
# SPDX-License-Identifier: BSD-3-Clause-Open-MPI

ran_node_name: gnb01            # Optional TEXT (gnb01). Sets the RAN node name.
gnb_id: 411                        # Optional UINT (411). gNodeB identifier.
gnb_id_bit_length: 22              # Optional UINT (22). gNodeB identifier length in bits. Supported: [22 - 32].
gnb_cu_up_id: 0                    # Optional UINT (0). Sets the gNB CU-UP ID. Supported: [0 - 68719476735].
gnb_du_id: 0                       # Optional UINT (0). Sets the gNB DU ID. Supported: [0 - 68719476735].
dryrun: false                      # Optional BOOLEAN (false). Enable application dry run mode. Supported: [false, true].

cells:                             # Optional TEXT. Sets the cell configuration on a per cell basis, overwriting the default configuration defined by cell_cfg.
  - pci: 1                         # Optional UINT. Override the PCI defined by the default cell in cell_cfg.
    dl_arfcn: 536020               # Optional UINT. Set the ARFCN for the above PCI. Note, this must match the value used in the cell_cfg section.
  - pci: 2                         # Optional UINT. Set the PCI of the second cell.
    dl_arfcn: 532020               # Optional UINT. Set the ARFCN of the second cell.

qos:                  # Optional TEXT. Configures RLC and PDCP radio bearers on a per 5QI basis. Each list entry should begin with "-".
- five_qi: 9 # Optional UINT (9). 5QI. Supported: [0 - 255].
  rlc:
    mode: am # Optional TEXT (am). RLC mode.
    um-bidir:
      tx:
        sn: 0          # Optional UINT (0). RLC UM TX SN.
        queue-size: 0  # Optional UINT (0). RLC UM TX SDU queue size.
        queue-bytes: 0 # Optional UINT (0). RLC UM TX SDU queue limit in bytes.
      rx:
        sn: 0           # Optional UINT (0). RLC UM RX SN.
        t-reassembly: 0 # Optional INT (0). RLC UM t-Reassembly.
    am:
      tx:
        sn: 0                 # Optional UINT (0). RLC AM TX SN size.
        t-poll-retransmit: 0  # Optional INT (0). RLC AM TX t-PollRetransmit (ms).
        max-retx-threshold: 0 # Optional UINT (0). RLC AM max retx threshold.
        poll-pdu: 0           # Optional INT (0). RLC AM TX PollPdu.
        poll-byte: 0          # Optional INT (0). RLC AM TX PollByte.
        max_window: 0         # Optional UINT (0). Non-standard parameter that limits the tx window size. Can be used for limiting memory usage with large windows. 0 means no limits other than the SN size (i.e. 2^[sn_size-1]).
```

```
        queue-size: 4096      # Optional UINT (4096). RLC AM TX SDU queue size.
        queue-bytes: 6172672  # Optional UINT (6172672). RLC AM TX SDU queue size
in bytes.
      rx:
        sn: 0                 # Optional UINT (0). RLC AM RX SN.
        t-reassembly: 0       # Optional INT (0). RLC AM RX t-Reassembly.
        t-status-prohibit: 0 # Optional INT (0). RLC AM RX t-StatusProhibit.
        max_sn_per_status: 0 # Optional UINT (0). RLC AM RX status SN limit.
  pdcp:
    rohc:
      rohc_type: none      # Optional TEXT (none). ROHC type
(none/rohc/ul_only_rohc). Values: {none, rohc, ul_only_rohc}. Default: none.
Supported: [none, rohc, uplink_only_rohc].
      max_cid: 15          # Optional UINT (15). Maximum CID.
      profile0x0001: false # Optional BOOLEAN (false). Configure profile0x0001
(ROHCv1 RTP/UDP/IP). Supported: [false, true].
      profile0x0002: false # Optional BOOLEAN (false). Configure profile0x0002
(ROHCv1 UDP/IP). Supported: [false, true].
      profile0x0003: false # Optional BOOLEAN (false). Configure profile0x0003
(ROHCv1 ESP/IP). Supported: [false, true].
      profile0x0004: false # Optional BOOLEAN (false). Configure profile0x0004
(ROHCv1 IP). Supported: [false, true].
      profile0x0006: false # Optional BOOLEAN (false). Configure profile0x0006
(ROHCv1 TCP/IP). Supported: [false, true].
      profile0x0101: false # Optional BOOLEAN (false). Configure profile0x0101
(ROHCv2 RTP/UDP/IP). Supported: [false, true].
      profile0x0102: false # Optional BOOLEAN (false). Configure profile0x0102
(ROHCv2 UDP/IP). Supported: [false, true].
      profile0x0103: false # Optional BOOLEAN (false). Configure profile0x0103
(ROHCv2 ESP/IP). Supported: [false, true].
      profile0x0104: false # Optional BOOLEAN (false). Configure profile0x0104
(ROHCv2 IP). Supported: [false, true].
    tx:
      sn: 0                       # Optional UINT (0). PDCP TX SN size.
      discard_timer: 0            # Optional INT (0). PDCP TX discard timer
(ms).
      status_report_required: false # Optional BOOLEAN (false). PDCP TX status
report required. Supported: [false, true].
    rx:
      sn: 0                      # Optional UINT (0). PDCP RX SN size.
      t_reordering: 0            # Optional INT (0). PDCP RX t-Reordering (ms).
      out_of_order_delivery: false # Optional BOOLEAN (false). PDCP RX enable out-
of-order delivery. Supported: [false, true].
  f1u_cu_up:
    backoff_timer: 5 # Optional INT (5). F1-U backoff timer (ms).
    queue_size: 8192 # Optional UINT (8192). F1-U backoff timer (ms).
    batch_size: 256  # Optional UINT (256). F1-U backoff timer (ms).
  f1u_du:
    backoff_timer: 0  # Optional INT (0). F1-U backoff timer (ms).
    ul_buffer_size: 0 # Optional UINT (0). F1-U handover buffer size.

srbs:                             # Optional TEXT. Configures signaling radio
bearers. Each list entry should begin with "-".
  - srb_id: 0 # Optional UINT (0). SRB Id. Format: legal values: {1, 2}.
    rlc:
      tx:
        sn: 0                 # Optional UINT (0). RLC AM TX SN size.
        t-poll-retransmit: 0  # Optional INT (0). RLC AM TX t-PollRetransmit
(ms).
        max-retx-threshold: 0 # Optional UINT (0). RLC AM max retx threshold.
```

```
        poll-pdu: 0           # Optional INT (0). RLC AM TX PollPdu.
        poll-byte: 0          # Optional INT (0). RLC AM TX PollByte.
        max_window: 0         # Optional UINT (0). Non-standard parameter that limits the tx window size. Can be used for limiting memory usage with large windows. 0 means no limits other than the SN size (i.e. 2^[sn_size-1]).
        queue-size: 4096      # Optional UINT (4096). RLC AM TX SDU queue size in PDUs.
        queue-bytes: 6172672  # Optional UINT (6172672). RLC AM TX SDU queue size in bytes.
      rx:
        sn: 0               # Optional UINT (0). RLC AM RX SN.
        t-reassembly: 0      # Optional INT (0). RLC AM RX t-Reassembly.
        t-status-prohibit: 0 # Optional INT (0). RLC AM RX t-StatusProhibit.
        max_sn_per_status: 0 # Optional UINT (0). RLC AM RX status SN limit.

cu_cp:
  max_nof_dus: 6                        # Optional UINT (6). Sets the maximum number of DU connections that the CU-CP may accept.
  max_nof_cu_ups: 6                     # Optional UINT (6). Sets the maximum number of CU-UP connections that the CU-CP may accept.
  max_nof_ues: 8192                     # Optional UINT (8192). Sets the maximum number of UEs that the CU-CP may accept.
  max_nof_drbs_per_ue: 8                # Optional UINT (8). Sets the maximum number of DRBs per UE. Supported: [1 - 29].
  inactivity_timer: 120                 # Optional INT (120). Sets the UE/PDU Session/DRB inactivity timer in seconds. Supported: [1 - 7200].
  request_pdu_session_timeout: 3        # Optional INT (3). Timeout for the setup of a PDU session after an InitialUeMessage was sent to the core, in seconds. The timeout must be larger
                                        # than T310. If the value is reached, the UE will be released.
  enable_rrc_inactive: false            # Optional BOOLEAN (false). Enables the RRC INACTIVE state for UEs. Supported: [false, true].
  ran_paging_cycle: 32                  # Optional UINT (32). Sets the default RAN paging cycle for INACTIVE UEs. Supported: [32, 64, 128, 256].
  nof_i_rnti_ue_bits: 13                # Optional UINT (13). Sets the number of bits used for the I-RNTI UE identifier. Supported: [1 - 18].
  t380: 10                              # Optional INT (10). Sets the RRC inactivity timer T380 in minutes. Supported: [5, 10, 20, 30, 60, 120, 360, 720].
  extra_amfs:                           # Optional TEXT. Sets the list of extra AMFs for the CU-CP to connect to.

  amf:
    no_core: false                       # Optional BOOLEAN (false). Setting to true allows the gNB to run without a core. Supported: [false, true].
    amf_reconnection_retry_time: 1000    # Optional UINT (1000). Sets the time in ms to wait before retrying to connect to the AMF after a failed connection attempt.
    addrs: 127.0.1.100                   # Required TEXT. Sets the AMF IP addresses for N2 interface. Multiple addresses supported for SCTP multi-homing. Format: IPV4 or IPV6 IP address.
    port: 38412                          # Optional UINT (38412). Sets the AMF port. Supported: [20000 - 40000].
    bind_addrs: 127.0.0.1                # Optional TEXT (127.0.0.1). Sets the local bind addresses for N2 interface. Multiple addresses supported for SCTP multi-homing.
    bind_interface: auto                 # Optional TEXT (auto). Sets network device to bind for N2 interface.
    sctp_rto_initial: 120                # Optional INT (120). Sets the initial retransmission timeout when creating the SCTP connection.
```

```
    sctp_rto_min: 120                    # Optional INT (120). Sets the minimum retransmission timeout for the SCTP connection.
    sctp_rto_max: 500                    # Optional INT (500). Sets the maximum retransmission timeout for the SCTP connection.
    sctp_init_max_attempts: 3            # Optional INT (3). Sets the maximum retransmission attempts for the initial SCTP connection.
    sctp_max_init_timeo: 500             # Optional INT (500). Sets the maximum retransmission timeout for the initial SCTP connection.
    sctp_hb_interval: 30000              # Optional INT (30000). SCTP heartbeat interval in milliseconds. Supported: [-1 - inf] (-1 to use system default).
    sctp_assoc_max_retx: 10              # Optional INT (10). SCTP association max retransmissions. Supported: [-1 - inf] (-1 to use system default).
    sctp_nodelay: false                  # Optional BOOLEAN (false). Setting to true allows the gNB to send SCTP messages as soon as possible without any Nagle-like algorithm. Supported: [false, true].
    procedure_timeout: 5000              # Optional UINT (5000). Sets the AMF NGAP procedure timeout in milliseconds.
    supported_tracking_areas:            # Optional TEXT. Sets the list of tracking areas supported by this AMF.
      -                                    # Supported TAC #1
        tac: 7                             # Required UINT. Sets the Tracking Area Code.
        plmn_list:                         # Required TEXT. Sets the list of PLMN items supported for this Tracking Area.
          -                                  # Supported PLMN #1
            plmn: "00101"                    # Required TEXT. Sets the Public Land Mobile Network code. Format: 7-digit PLMN code containing MCC & MNC.
            tai_slice_support_list:          # Required TEXT. Sets the list of TAI slices for this PLMN item.
              -                                # Supported TAI Slice(s) for PLMN
                sst: 1                         # Optional UINT (0). Sets the Slice Service Type. Supported: [0 - 255].
                sd: 16777215                   # Optional UINT (16777215). Sets the Service Differentiator. Supported: [0 - 16777215].

  xnap:
    procedure_timeout: 5000              # Optional UINT (5000). Sets the Xn-AP procedure timeout in milliseconds.
    reconnect_timer: 10000               # Optional UINT (10000). Sets the Xn-AP reconnect timer in milliseconds.
    no_connection_init: false            # Optional BOOLEAN (false). Disables Xn-AP connection initiation. Supported: [false, true].
    gateways:                            # Optional TEXT. Sets the list of Xn-AP gateway addresses. Each list entry should begin with "-".
      -
        bind_addrs: x.x.x.x              # Required TEXT. Sets the gateway IP addresses.
        sctp:
          sctp_rto_initial: 120
          sctp_rto_min: 120
          sctp_rto_max: 500
          sctp_init_max_attempts: 3
          sctp_max_init_timeo: 500
          sctp_hb_interval: 30000
          sctp_assoc_max_retx: 10
          sctp_nodelay: false
        connections:                 # Optional TEXT. Sets the list of Xn-C peer connections reachable via this gateway.
          -
```

```
150             peer_addrs:             # Required TEXT. Sets the peer gNB IP addresses
    for this XnAP connection.
151
152   mobility:
153     trigger_handover_from_measurements: false   # Optional BOOLEAN (false). Sets
    whether or not to start HO if neighbor cells become stronger. Supported: [false,
    true].
154     trigger_cho_on_ue_setup: false              # Optional BOOLEAN (false). Sets
    whether to trigger Conditional Handover on UE setup. Supported: [false, true].
155     cho_timeout_ms: 10000                       # Optional UINT (10000). Sets the
    Conditional Handover timeout in milliseconds. Supported: [1 - 600000].
156     cells:                                      # Optional TEXT. Sets the list of
    cells known to the CU-CP, their configs (if not provided over F1) and their
    respective neighbor cells.
157       -                                         # Define Cell 1
158         nr_cell_id:                               # Required UINT. The ID of a cell
    that belongs to this gNB.
159         periodic_report_cfg_id:                   # Optional UINT. The periodical
    report configuration to use for this cell.
160         ncells:                                   # Required TEXT. List of neighbor
    cells.
161           -                                       # Define neighbour cell(s) of
    Cell 1.
162             nr_cell_id:                           # Required UINT. The ID of this
    neighbor cell.
163             report_configs:                       # Required TEXT. List of report
    configurations to use for measurements of this neighbor cell.
164       -                                         # Define Cell 2
165         nr_cell_id:                               # Required UINT. The ID of a cell
    that does not belong to this gNB.
166         ncells:                                   # Required TEXT. List of neighbor
    cells.
167           -                                       # Define neighbour cell(s) of
    Cell 2.
168             nr_cell_id:                           # Required UINT. The ID of this
    neighbor cell.
169             report_configs:                       # Required TEXT. List of report
    configurations to use for measurements of this neighbor cell.
170         gnb_id_bit_length:                        # Optional UINT. The ID of this
    cell's gNB. Must be present if the cell does not belong to this gNB. Supported: [22
    - 32].
171         pci:                                      # Optional UINT. The PCI of this
    cell. Must be present if the cell does not belong to this gNB. Supported: [0 -
    1007].
172         band:                                     # Optional TEXT (enum). The NR
    band of this cell. Must be present if the cell does not belong to this gNB.
173         ssb_arfcn:                                # Optional UINT. Sets the SSB
    ARFCN of this cell.
174         plmn:                                     # Optional TEXT. Sets the PLMN of
    this cell.
175         tac:                                      # Optional UINT. Sets the
    Tracking Area Code of this cell. Supported: [0 - 16777215].
176         ssb_scs:                                  # Optional UINT. The SSB
    subcarrier spacing of this cell in kHz. Must be present if cell does not belong to
    this gNB. Supported: [15, 30, 60, 120, 240].
177         ssb_period:                               # Optional UINT. The SSB period
    of this cell in ms. Must be present if cell does not belong to this gNB. Supported:
    [5, 10, 20, 40, 80, 160].
178         ssb_offset:                               # Optional UINT. The SSB offset
    of this cell. Must be present if cell does not belong to this gNB.
```

```
        ssb_duration:                           # Optional UINT. The SSB duration of this cell in subframes. Must be present if cell does not belong to this gNB. Supported: [1, 2, 3, 4, 5].
    report_configs:                             # Optional TEXT. Sets the list of report configurations to dynamically build a measurement configuration sent to UEs.
      -                                         # Define report configs to be used by the UEs
        report_cfg_id:                          # Required UINT. The ID of this report configuration.
        report_type:                            # Required TEXT. The type of the report. Supported: [event_triggered, periodical, cond_trigger].
        event_triggered_report_type:            # Optional TEXT. Sets the type of event triggered report. Supported: [a1, a2, a3, a4, a5, a6, d1, t1, d2].
        report_interval_ms:                      # Optional UINT. The report interval in ms. Supported: [120, 240, 480, 640, 1024, 2048, 5120, 10240, 20480, 40960, 60000, 360000, 720000, 1800000].
        meas_trigger_quantity:                  # Optional TEXT. Sets the quantity to be measured. Supported: [rsrp, rsrq, sinr].
        meas_trigger_quantity_threshold_db:     # Optional INT. Sets the measurement trigger quantity threshold in dB for A1/A2/A4/A5 events. Valid ranges: RSRP [-156..-31] dBm, RSRQ [-43..20] dB, SINR [-23..40] dB. Supported: [-156 - 40].
        meas_trigger_quantity_threshold_2_db:   # Optional INT. Sets the measurement trigger quantity threshold 2 in dB for A5 events. Valid ranges: RSRP [-156..-31] dBm, RSRQ [-43..20] dB, SINR [-23..40] dB. Supported: [-156 - 40].
        meas_trigger_quantity_offset_db:        # Optional INT. Sets the measurement trigger quantity offset in dB for A3/A6 events. Supported: [-15 - 15].
        hysteresis_db:                          # Optional UINT. Sets the hysteresis in dB for measurement report trigger. Supported: [0 - 15].
        time_to_trigger_ms:                     # Optional UINT. Sets the time in ms during which a condition must be met before measurement report trigger. Supported: [0, 40, 64, 80, 100, 128, 160, 256, 320, 480, 512, 640, 1024, 1280, 2560, 5120].
        periodic_ho_rsrp_offset_db: -1          # Optional INT (-1). Sets the RSRP offset for periodic HO triggering in dB. -1 disables. Note the actual value is field value * 0.5 dB. Supported: [-1 - 30].
        t312:                                   # Optional UINT. Sets the T312 timer in ms. Supported: [0, 50, 100, 200, 300, 400, 500, 1000].
        distance_thresh_from_ref1_km:           # Optional FLOAT. Sets the D1/D2 distance threshold 1 in km. Supported: [0.0 - 3276.75].
        distance_thresh_from_ref2_km:           # Optional FLOAT. Sets the D1/D2 distance threshold 2 in km. Supported: [0.0 - 3276.75].
        hysteresis_location_km:                 # Optional FLOAT. Sets the D1/D2 location hysteresis in km. Supported: [0.0 - 327.68].
        ref_location1:
          latitude:                             # Optional FLOAT. Sets D1 reference location 1 latitude in degrees. Supported: [-90.0 - 90.0].
          longitude:                            # Optional FLOAT. Sets D1 reference location 1 longitude in degrees. Supported: [-180.0 - 180.0].
        ref_location2:
          latitude:                             # Optional FLOAT. Sets D1 reference location 2 latitude in degrees. Supported: [-90.0 - 90.0].
          longitude:                            # Optional FLOAT. Sets D1 reference location 2 longitude in degrees. Supported: [-180.0 - 180.0].
        t1_thres:                               # Optional TEXT. Sets the T1 time threshold. Format: Unix ms integer or YYYY-MM-DDTHH:MM:SS[.mmm].
        duration_s:                             # Optional FLOAT. Sets the T1 duration in seconds. Supported: [0.1 - 600.0].
```

```yaml

  rrc:
    force_reestablishment_fallback: false    # Optional BOOLEAN (false). Forces RRC re-establishment fallback to RRC setup. Supported: [false, true].
    force_resume_fallback: false             # Optional BOOLEAN (false). Forces RRC Resume to fall back to RRC Setup. Supported: [false, true].
    rrc_procedure_guard_time_ms: 1000        # Optional UINT (1000). Sets the guard time in ms used for RRC message exchange with UE. This is added to the RRC procedure timeout.

  security:
    integrity: not_needed                    # Optional TEXT (not_needed). Sets the default integrity protection indication for DRBs. Supported: [not_needed, preferred, required].
    confidentiality: required                # Optional TEXT (required). Sets the default confidentiality protection indication for DRBs. Supported: [not_needed, preferred, required].
    nea_pref_list: nea0,nea2,nea1,nea3       # Optional TEXT (nea0,nea2,nea1,nea3). Sets the ordered preference list for the selection of encryption algorithm (NEA).
    nia_pref_list: nia2,nia1,nia3            # Optional TEXT (nia2,nia1,nia3). Sets the ordered preference list for the selection of integrity algorithm (NIA).

  f1ap:
    procedure_timeout: 1000                  # Optional UINT (1000). Sets the time the F1AP waits for a DU response in milliseconds.

  e1ap:
    procedure_timeout: 1000                  # Optional UINT (1000). Sets the time the E1AP waits for a CU-UP response in milliseconds.

cu_up:
  max_nof_ues: 16384                      # Optional UINT (16384). Sets the maximum number of UE bearer contexts supported by the CU-UP. Supported: [1 - 65536].
  warn_on_drop: false                     # Optional BOOLEAN (false). Enables logging a warning for dropped packets in GTP-U and PDCP due to full queues. Supported: [false, true].
  plmn_list:                              # Optional TEXT. Sets the list of supported PLMN IDs. Each entry is a 5 or 6 digit string (MCC+MNC). Supported: up to 12 PLMNs.

  ngu:
    no_core: false                         # Optional BOOLEAN (false). Setting to true allows the gNB to run without a core. Supported: [false, true].
    gtpu:
      queue_size: 2046                     # Optional UINT (2046). Sets the GTP-U queue size, in PDUs.
      batch_size: 256                      # Optional UINT (256). Sets the maximum number of GTP-U PDUs processed in a batch.
      reordering_timer: 0                  # Optional UINT (0). Sets the GTP-U RX reordering timer in milliseconds. 0 disables reordering.
      rate_limiter_period: 100             # Optional UINT (100). Sets the GTP-U RX rate limiter period in milliseconds.
      ignore_ue_ambr: true                 # Optional BOOLEAN (true). Ignores GTP-U DL UE-AMBR rate limiter. Supported: [false, true].
      teid_release_linger_time: 100        # Optional UINT (100). Sets the TEID linger time after release in milliseconds.
    socket:                                # Optional TEXT. Defines socket(s) for NG-U interface. Each list entry should begin with "-".
      -                                    # Socket 1
```

```
        bind_addr: 127.0.0.1                # Optional TEXT (127.0.0.1). Sets local IP address to bind for N3 interface. Format: IPV4 or IPV6 IP address.
        bind_interface: auto                # Optional TEXT (auto). Sets network device to bind for N3 interface.
        ext_addr: auto                      # Optional TEXT (auto). Sets the external IP address for N3 traffic. "auto" uses bind_addr.
        udp:
          max_rx_msgs: 256                  # Optional UINT (256). Sets the maximum number of messages RX from a socket in a single syscall.
          pool_threshold: 0.9               # Optional FLOAT (0.9). Sets the pool occupancy threshold above which packets are dropped. Supported: [0.0 - 1.0].
          tx_qsize: 4096                    # Optional UINT (4096). Sets the size of TX queue used for batching SDUs.
          max_tx_msgs: 256                  # Optional UINT (256). Sets the maximum amount of messages TX in a single syscall.
          max_tx_segments: 256              # Optional UINT (256). Sets the maximum amount of segments TX in a single SDU.
          reuse_addr: false                 # Optional BOOLEAN (false). Allow multiple sockets to bind to the same port. Supported: [false, true].
          dscp:                             # Optional UINT. Sets the Differentiated Services Code Point value. Supported: [0 - 63].

  test_mode:
    enable: false                           # Optional BOOLEAN (false). Enables or disables CU-UP test mode. Supported: [false, true].
    integrity_enable: true                  # Optional BOOLEAN (true). Enables or disables PDCP integrity testing. Supported: [false, true].
    ciphering_enable: true                  # Optional BOOLEAN (true). Enables or disables PDCP ciphering testing. Supported: [false, true].
    nea_algo: 2                             # Optional UINT (2). Sets the NEA algorithm index for test mode. Supported: [0 - 3].
    nia_algo: 2                             # Optional UINT (2). Sets the NIA algorithm index for test mode. Supported: [1 - 3].
    ue_ambr: 40000000000                    # Optional UINT (40000000000). Sets the DL UE-AMBR used for testing in bps.
    attach_detach_period: 0                 # Optional UINT (0). Sets the attach/detach period for test mode in milliseconds. 0 means always attached.
    reestablish_period: 0                   # Optional UINT (0). Sets the reestablish period for test mode in milliseconds. 0 means no re-establishment.
    f1u_peer_address: 127.0.10.2            # Optional TEXT (127.0.10.2). Sets the F1-U peer address for test mode.
    nof_ues: 1                              # Optional UINT (1). Sets the number of test UEs.

du:
  warn_on_drop: false                     # Optional BOOLEAN (false). Enables logging a warning for dropped packets in F1-U, RLC and MAC due to full queues. Supported: [false, true].

cell_cfg:
  pci: 1                                  # Optional UINT (1). Sets the Physical Cell ID. Supported: [0 - 1007].
  sector_id:                               # Optional UINT. Sets the Sector ID (4-14 bits). Concatenated with gNB Id to form NR Cell Identity. Auto-derived if not specified. Supported: [0 - 16383].
  dl_arfcn: 536020                        # Required UINT (536020). Sets the Downlink ARFCN.
  band: auto                              # Optional TEXT (auto). Sets the NR band being used for the cell. If not specified, will be set automatically based on ARFCN.
```

```
  common_scs: 15                        # Required UINT (15). Sets the subcarrier spacing in kHz. Supported: [15, 30, 60, 120].
  channel_bandwidth_MHz: 20             # Required UINT (20). Sets the channel bandwidth in MHz. Supported: [5, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100].
  nof_antennas_dl: 1                    # Optional UINT (1). Sets the number of antennas for downlink transmission. Supported: [1, 2, 4].
  nof_antennas_ul: 1                    # Optional UINT (1). Sets the number of antennas for uplink transmission. Supported: [1, 2, 4].
  plmn: "00101"                         # Required TEXT (00101). Sets the Public Land Mobile Network code. Format: 7-digit PLMN code containing MCC & MNC.
  tac: 7                                # Required UINT (7). Sets the Tracking Area Code.
  enabled: true                         # Optional BOOLEAN (true). Sets whether the cell is enabled on start-up or not. Supported: [false, true].
  cell_barred: false                    # Optional BOOLEAN (false). Sets whether the cell is broadcast as barred for new access. Supported: [false, true].
  intra_freq_reselection: true          # Optional BOOLEAN (true). Enables intra-frequency cell reselection. Supported: [false, true].
  q_rx_lev_min: -70                     # Optional INT (-70). Sets the required minimum received RSRP level for cell selection/re-selection, in dBm. Supported: [-70 - -22].
  q_qual_min: -20                       # Optional INT (-20). Sets the required minimum received RSRQ level for cell selection/re-selection, in dB. Supported: [-43 - -12].
  pcg_p_nr_fr1:                         # Optional INT. Sets the maximum total TX power for the UE in this NR cell group in FR1. Supported: [-30 - +23].
  additional_plmns:                     # Optional TEXT. Sets additional PLMN list for the cell.
  slicing:                              # Optional TEXT. Configures network slicing options. This can only be set via the configuration file.
    -                                     # Configure Slice 1, for further slices, replicate below format.
      sst: 1                                # Optional UINT (1). Sets the Slice Service Type. Supported: [0 - 255].
      sd: 0                                 # Optional UINT (0). Sets the Service Differentiator. Supported: [0 - 16777215].
      sched_cfg:                            # Scheduler configuration for the slice.
        min_prb_policy_ratio: 0               # Optional UINT (0). Sets the minimum percentage of PRBs to be allocated to the slice. Supported: [0 - 100].
        max_prb_policy_ratio: 100             # Optional UINT (100). Sets the maximum percentage of PRBs to be allocated to the slice. Supported: [1 - 100].
        ded_prb_policy_ratio: 1               # Optional UINT. Sets the dedicated percentage of PRBs to be allocated to the slice. Supported: [1 - 100].
        priority: 0                           # Optional UINT (0). Sets the slice scheduler priority. Supported: [0 - 254].
        policy:
          qos_sched:
            combine_function: gbr_prioritized     # Optional TEXT (gbr_prioritized). Sets the QoS-aware scheduler weight combining function. Supported: [gbr_prioritized, geometric_mean].
            pf_fairness_coeff: 2                  # Optional FLOAT (2). Sets the fairness coefficient for the Proportional Fair policy scheduler.
            prio_enabled: true                    # Optional BOOLEAN (true). Enables the QoS Flow priority in QoS-aware scheduling. Supported: [false, true].
            pdb_enabled: true                     # Optional BOOLEAN (true). Enables the Packet Delay Budget (PDB) in QoS-aware scheduling. Supported: [false, true].
```

```
            gbr_enabled: true                    # Optional BOOLEAN (true). Enables the Guaranteed Bit Rate (GBR) in QoS-aware scheduling. Supported: [false, true].
          rr_sched:                              # Optional. Round-robin scheduler policy configuration.

  mac_cell_group:
    bsr_cfg:
      periodic_bsr_timer: 10                     # Optional UINT (10). Sets the periodic Buffer Status Report Timer value in nof. subframes. Supported: [1, 5, 10, 16, 20, 32, 40, 64, 80, 128, 160, 320, 640, 1280, 2560, 0].
      retx_bsr_timer: 80                         # Optional UINT (80). Sets the retransmission Buffer Status Report Timer value in nof. subframes. Supported: [10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120, 10240].
      lc_sr_delay_timer:                         # Optional UINT. Sets the Logical Channel SR delay timer in nof. subframes. Supported: [20, 40, 64, 128, 512, 1024, 2560].
    phr_cfg:
      phr_prohibit_timer: 10                     # Optional UINT (10). Sets the PHR prohibit timer in nof. subframes. Supported: [0, 10, 20, 50, 100, 200, 500, 1000].
    sr_cfg:
      sr_trans_max: 64                           # Optional UINT (64). Sets the maximum number of SR transmissions. Supported: [4, 8, 16, 32, 64].
      sr_prohibit_timer:                         # Optional TEXT. Sets the timer for SR transmission on PUCCH in ms. Supported: [1, 2, 4, 8, 16, 32, 64, 128].

  ssb:
    ssb_period: 10                               # Optional UINT (10). Sets the period of SSB scheduling in milliseconds. Supported: [5, 10, 20].
    ssb_block_power_dbm: -16                     # Optional INT (-16). Sets the SS PBCH block power in dBm. Supported: [-60 - +50].
    pss_to_sss_epre_db: 0                        # Optional UINT (0). Sets the PSS to SSS EPRE ratio in dB. Supported: [0, 3].


  sib:
    si_window_length: 160                        # Optional UINT (160). Sets the length of the SI scheduling window in slots. Supported: [5, 10, 20, 40, 80, 160, 320, 640, 1280].
    si_sched_info:                               # Optional TEXT. Configures the scheduling for each of the SI-messages broadcast by the gNB.
    t300: 1000                                   # Optional UINT (1000). Sets the RRC Connection Establishment timer in ms. Supported: [100, 200, 300, 400, 600, 1000, 1500, 2000].
    t301: 1000                                   # Optional UINT (1000). Sets the RRC Connection Re-establishment timer in ms. Supported: [100, 200, 300, 400, 600, 1000, 1500, 2000].
    t310: 1000                                   # Optional UINT (1000). Sets the Out-of-sync timer in ms. Supported: [0, 50, 100, 200, 500, 1000, 2000].
    n310: 1                                      # Optional UINT (1). Sets the Out-of-sync counter. Supported: [1, 2, 3, 4, 6, 8, 10, 20].
    t311: 3000                                   # Optional UINT (3000). Sets the RRC Connection Re-establishment procedure timer in ms. Supported: [1000, 3000, 5000, 10000, 15000, 20000, 30000].
    n311: 1                                      # Optional UINT (1). Sets the In-sync counter. Supported: [1, 2, 3, 4, 5, 6, 8, 10].
    t319: 1000                                   # Optional UINT (1000). Sets the RRC Connection Resume timer in ms. Supported: [100, 200, 300, 400, 600, 1000, 1500, 2000].
```

```
    sib2:
      q_hyst: 3                          # Optional UINT (3). Sets the hysteresis value for ranking criteria. Supported: [0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24].
      thresh_serving_low_p: 0            # Optional UINT (0). Sets the Rx level threshold for serving cell when reselecting to lower priority RAT/frequency. Supported: [0 - 31].
      cell_reselection_priority: 6       # Optional UINT (6). Sets the cell reselection priority for this cell's frequency. Supported: [0 - 7].
      q_rx_lev_min: -140                 # Optional INT (-140). Sets the minimum required Rx level in the cell in dBm. Supported: even values in [-140, -44].
      s_intra_search_p: 62               # Optional UINT (62). Sets the Rx level threshold for intra-frequency measurements in dB. Supported: even values in [0, 62].
      t_reselection_nr: 1                # Optional UINT (1). Sets the cell reselection timer value in seconds. Supported: [0 - 7].
    sib3:
      intra_freq_neigh_cell_list:        # Optional TEXT. Sets the intra-frequency neighbor cell list.
        - pci: 0                           # Optional UINT (0). Sets the PCI of the neighbour cell. Supported: [0 - 1007].
          q_offset_cell: 0                 # Optional INT (0). Sets the cell-specific offset for this neighbour. Format: legal values: {-24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24}.
      intra_freq_excluded_cell_list:     # Optional TEXT. Sets the intra-frequency excluded cell list.
        - start: 0                         # Required UINT (0). Sets the range start. Supported: [0 - 1007].
          size: 1                          # Required UINT (1). Sets the range size. Format: legal values: {1, 4, 8, 12, 16, 24, 32, 48, 64, 84, 96, 128, 168, 252, 504, 1008}.
    sib4:
      inter_freq_carrier_freq_list:      # Optional TEXT. Sets the inter-frequency carrier frequency list.
        - arfcn: 0                         # Optional UINT (0). Sets the carrier ARFCN.
          ssb_scs: 15kHz                   # Optional TEXT (15kHz). Sets the SSB subcarrier spacing. Format: accepts SCS strings (e.g. "15kHz", "30kHz").
          derive_ssb_index_from_cell: false  # Optional BOOLEAN (false). Derives SSB index from cell. Supported: [false, true].
          q_rx_lev_min: -140               # Optional INT (-140). Sets the minimum required Rx level in dBm. Format: must be an even value within [-140, -44].
          thresh_x_high_p: 0               # Optional UINT (0). Sets the Rx level threshold in dB for reselection to a higher priority RAT/frequency. Format: must be an even value within [0, 62].
          thresh_x_low_p: 0                # Optional UINT (0). Sets the Rx level threshold in dB for reselection to a lower priority RAT/frequency. Format: must be an even value within [0, 62].
          q_offset_freq: 0                 # Optional INT (0). Sets the frequency-specific offset in dB for equal priority NR frequencies. Format: legal values: {-24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24}.
    sib5:
      t_reselection_eutra:               # Optional UINT. Sets the EUTRA cell reselection timer value in seconds. Supported: [0 - 7].
      carrier_freq_list_eutra:           # Optional TEXT. Sets the EUTRA carrier frequency list.
```

```
355         - earfcn: 0                          # Optional UINT (0). Sets the EUTRA
    EARFCN.
356           allowed_meas_bandwidth: 0          # Optional UINT (0). Sets the allowed
    measurement bandwidth. Format: legal values: {6, 15, 25, 50, 75, 100}.
357           presence_antenna_port1: false      # Optional BOOLEAN (false). Sets
    whether all neighbour cells use Antenna Port 1. Supported: [false, true].
358           cell_reselection_priority:         # Required UINT. Sets the cell
    reselection priority for this frequency. Supported: [0 - 7].
359           thresh_x_high: 0                   # Optional UINT (0). Sets the Rx
    level threshold in dB for reselection to a higher priority RAT/frequency. Format:
    must be an even value within [0, 62].
360           thresh_x_low: 0                    # Optional UINT (0). Sets the Rx
    level threshold in dB for reselection to a lower priority RAT/frequency. Format:
    must be an even value within [0, 62].
361           q_rx_lev_min: -140                 # Optional INT (-140). Sets the
    minimum required Rx level in dBm. Format: must be an even value within [-140,
    -44].
362           q_qual_min: -34                    # Optional INT (-34). Sets the
    minimum required quality level in dB. Supported: [-34 - -3].
363           p_max_eutra: 33                    # Optional INT (33). Sets the maximum
    allowed transmission power in dBm. Supported: [-30 - 33].
364       sib16:
365         freq_prio_list_slicing:              # Optional TEXT. Sets the frequency
    priority slicing list entries.
366         - dl_implicit_carrier_freq: 0        # Optional UINT (0). Sets the DL
    implicit carrier frequency index. Supported: [0 - 8].
367           slice_info_list:                   # Optional TEXT. Sets the slice info
    list entries.
368             - nsag_id: 0                     # Optional UINT (0). Sets the NSAG
    ID. Supported: [0 - 255].
369               allowed: true                  # Optional BOOLEAN (true). Sets
    whether the cell list is allowed (true) or excluded (false). Supported: [false,
    true].
370               reselection_priority: 0        # Optional FLOAT (0). Sets the
    priority for this cell reselection slice. Format: must be a multiple of 0.2 within
    [0.0, 7.8].
371               cells_allowed:                 # Optional TEXT. Sets the slice
    cell list entries.
372                 - start: 0                   # Required UINT (0). Sets the
    range start. Supported: [0 - 1007].
373                   size: 1                    # Required UINT (1). Sets the
    range size. Format: legal values: {1, 4, 8, 12, 16, 24, 32, 48, 64, 84, 96, 128,
    168, 252, 504, 1008}.
374       etws:
375         message_id: 4356                     # Optional UINT (4356). Sets the ETWS
    message ID. Supported: [0 - 65535].
376         serial_num: 12288                    # Optional UINT (12288). Sets the ETWS
    message serial number. Supported: [0 - 65535].
377         warning_type: 2432                   # Optional UINT (2432). Sets the ETWS
    warning type. Supported: [0 - 65535].
378         data_coding_scheme: 0                # Optional UINT (0). Sets the ETWS CBS
    data coding scheme. Supported: [0 - 255].
379         warning_message:                     # Optional TEXT. Sets the ETWS warning
    message string.
380       cmas:
381         message_id: 4370                     # Optional UINT (4370). Sets the CMAS
    message ID. Supported: [0 - 65535].
382         serial_num: 12291                    # Optional UINT (12291). Sets the CMAS
    message serial number. Supported: [0 - 65535].
```

```
      data_coding_scheme: 0                 # Optional UINT (0). Sets the CMAS CBS data coding scheme. Supported: [0 - 255].
      warning_message:                      # Optional TEXT. Sets the CMAS warning message string.

  ul_common:
    p_max:                                  # Optional INT. Sets maximum transmit power allowed in this serving cell. Supported: [-30 - +23].
    max_pucchs_per_slot: 31                 # Optional UINT (31). Sets the maximum number of PUCCH grants that can be allocated per slot. Supported: [1 - 128].
    max_ul_grants_per_slot: 32              # Optional UINT (32). Sets the maximum number of UL grants that can be allocated per slot. Supported: [1 - 144].
    min_pucch_pusch_prb_distance: 1         # Optional UINT (1). Sets the minimum PRB distance between PUCCH and UE-dedicated PUSCH grants. Supported: [0 - 137].

  pdcch:
    common:
      coreset0_index:                           # Optional INT. Sets the CORESET 0 index. Supported: [0 - 15].
      ss1_n_candidates: {0,0,2,0,0}             # Optional UINT ({0,0,2,0,0}). Sets the number of PDCCH candidates per aggregation level for SearchSpace#1. Supported: any 5-value array of [0 - 8].
      ss0_index: 0                              # Optional UINT (0). Sets the SearchSpace#0 index. Supported: [0 - 15].
      max_coreset0_duration:                    # Optional INT. Sets the maximum CORESET#0 duration in OFDM symbols. Supported: [1 - 2].
    dedicated:
      coreset1_rb_start:                        # Optional UINT. Sets the starting common resource block for CORESET 1, relative to CRB0. Default: derived automatically. Supported: [0 - 275].
      coreset1_l_crb:                           # Optional INT. Sets the length of CORESET 1 in number of CRBs. Supported: [0 - 275].
      coreset1_duration:                        # Optional INT. Sets the duration of CORESET 1 in number of OFDM symbols. Supported: [1 - 2].
      ss2_n_candidates: [0, 0, 0, 0, 0]         # Optional UINT ([0, 0, 0, 0, 0]). Sets the number of PDCCH candidates per aggregation level for SearchSpace#2. Supported: any 5-value array of [0 - 8].
      dci_format_0_1_and_1_1: true              # Optional BOOLEAN (true). Sets whether to use non-fallback or fallback DCI format in UE SearchSpace#2. Supported: [false, true].
      ss2_type: ue_dedicated                    # Optional TEXT (ue_dedicated). Sets the SearchSpace type for UE dedicated SearchSpace#2. Supported: [common, ue_dedicated].
      al_cqi_offset: 0                          # Optional FLOAT (0). Sets the offset applied to the CQI value in PDCCH aggregation level calculation. Supported: [-15 - 15].

  pdsch:
    min_ue_mcs: 0                           # Optional UINT (0). Sets a minimum PDSCH MCS value for all UEs. Supported: [0 - 28].
    max_ue_mcs: 28                          # Optional UINT (28). Sets a maximum PDSCH MCS value for all UEs. Supported: [0 - 28].
    fixed_rar_mcs: 0                        # Optional UINT (0). Sets a fixed RAR MCS value for all UEs. Supported: [0 - 28].
    fixed_sib1_mcs: 5                       # Optional UINT (5). Sets a fixed SIB1 MCS for all UEs. Supported: [0 - 28].
    nof_harqs: 16                           # Optional UINT (16). Sets the number of DL HARQ processes. Supported: [2, 4, 6, 8, 10, 12, 16, 32].
    max_nof_harq_retxs: 4                   # Optional UINT (4). Sets the maximum number of times a DL HARQ can be retransmitted before it is discarded. Supported:
```

```
    [0 - 64].
414     harq_retx_timeout: 100                  # Optional UINT (100). Sets the maximum
    time in ms between a HARQ NACK and the scheduler allocating the HARQ for
    retransmission. Supported: [10 - 500].
415     max_consecutive_kos: 100                # Optional UINT (100). Sets the maximum
    number of consecutive HARQ-ACK KOs before an RLF is reported. Supported: [0 -
    inf].
416     rv_sequence: [0,2,3,1]                  # Optional UINT ([0, 2, 3, 1]). Sets
    the redundancy version sequence to use for PDSCH. Supported: any combination of [0,
    1, 2, 3].
417     mcs_table: qam256                       # Optional ENUM (qam256). Sets the MCS
    table to use for PDSCH. Supported: [qam64, qam256, qam64lowse].
418     harq_feedback_disabled: false           # Optional TEXT (false). Sets whether
    to disable DL HARQ Feedback (NTN only). Supported: [true, false] or 32-bit
    bitmask.
419     interleaving_bundle_size: 0             # Optional UINT (0). Sets the PDSCH
    interleaving bundle size. Supported: [0, 2, 4].
420     max_rank:                               # Optional UINT. Sets the maximum
    number of PDSCH transmission layers. Supported: non-negative integer (0 or
    greater).
421     enable_csi_rs_pdsch_multiplexing: true  # Optional BOOLEAN (true). Enables
    multiplexing of CSI-RS and PDSCH. Supported: [false, true].
422     min_rb_size: 1                          # Optional UINT (1). Sets the minimum
    RB size for UE PDSCH resource allocation. Supported: [1 - 275].
423     max_rb_size: 275                        # Optional UINT (275). Sets the maximum
    RB size for UE PDSCH resource allocation. Supported: [1 - 275].
424     start_rb: 0                             # Optional UINT (0). Sets the start RB
    for resource allocation of UE PDSCHs. Supported: [0 - 275].
425     end_rb: 275                             # Optional UINT (275). Sets the end RB
    for resource allocation of UE PDSCHs. Supported: [0 - 275].
426     max_pdschs_per_slot: 35                 # Optional UINT (35). Sets the maximum
    number of PDSCH grants per slot, including SIB, RAR, Paging and UE data grants.
    Supported: [1 - 35].
427     max_alloc_attempts: 35                  # Optional UINT (35). Sets the maximum
    number of DL or UL PDCCH grant allocation attempts per slot before the scheduler
    skips the slot. Supported: [1 - 35].
428     olla_cqi_inc_step: 0.001                # Optional FLOAT (0.001). Sets the
    outer-loop link adaptation (OLLA) increment value. 0 disables OLLA. Supported: [0 -
    1].
429     olla_target_bler: 0.01                  # Optional FLOAT (0.01). Sets the
    target DL BLER in the OLLA algorithm. Supported: [0 - 0.5].
430     olla_max_cqi_offset: 4                  # Optional FLOAT (4). Sets the maximum
    CQI offset that OLLA can apply. Supported: positive float.
431     dc_offset:                              # Optional TEXT. Sets the Direct
    Current offset in number of subcarriers. Supported: [-1650 - 1649] OR [outside,
    undetermined, center].
432     harq_la_cqi_drop_threshold: 3           # Optional UINT (3). Sets the LA
    threshold for CQI drop above which HARQ retransmissions are cancelled. 0 disables.
    Supported: [0 - 15].
433     harq_la_ri_drop_threshold: 1            # Optional UINT (1). Sets the LA
    threshold for nof. layers drop above which HARQ retransmission is cancelled. 0
    disables. Supported: [0 - 4].
434     dmrs_additional_position: 2             # Optional UINT (2). Sets the PDSCH
    DMRS additional position. Supported: [0 - 3].
435
436   pusch:
437     min_ue_mcs: 0                           # Optional UINT (0). Sets a minimum
    PUSCH MCS value for all UEs. Supported: [0 - 28].
438     max_ue_mcs: 28                          # Optional UINT (28). Sets a maximum
    PUSCH MCS value for all UEs. Supported: [0 - 28].
```

```
    max_nof_harq_retxs: 4                   # Optional UINT (4). Sets the maximum number of times a UL HARQ can be retransmitted. Supported: [0 - 64].
    harq_retx_timeout: 100                  # Optional UINT (100). Sets the maximum time in ms between a CRC=KO and the scheduler allocating the HARQ for retransmission. Supported: [10 - 500].
    max_consecutive_kos: 100                # Optional UINT (100). Sets the maximum number of consecutive CRC KOs before an RLF is reported. Supported: [0 - inf].
    rv_sequence: [0]                        # Optional UINT ([0]). Sets the redundancy version sequence to use for PUSCH. Supported: any combination of [0, 1, 2, 3].
    mcs_table: qam256                       # Optional ENUM (qam256). Sets the MCS table to use for PUSCH. Supported: [qam64, qam256, qam64lowse].
    harq_mode_b: false                      # Optional TEXT (false). Sets HARQ Mode B (NTN only). Supported: [true, false] or 32-bit bitmask.
    nof_harqs: 16                           # Optional UINT (16). Sets the number of UL HARQ processes. Supported: [16, 32].
    ema_alpha_cl_pw_control_sinr: 0.5       # Optional FLOAT (0.5). Sets the EMA smoothing factor alpha for PUSCH closed-loop power control SINR. Supported: (0, 1) exclusive.
    max_rank: 4                             # Optional UINT (4). Sets the maximum number of PUSCH transmission layers. Supported: [1 - 4].
    msg3_delta_preamble: 6                  # Optional INT (6). Sets the MSG3 DeltaPreamble power offset between MSG3 and RACH preamble transmission. Supported: [-1 - 6].
    p0_nominal_with_grant: -76              # Optional INT (-76). Sets the P0 value for PUSCH grant (except MSG3), in dBm. Supported: multiples of 2 within the range [-202, 24].
    p0_nominal_without_grant: -76           # Optional INT (-76). Sets the P0 value for PUSCH without grant, in dBm. Supported: multiples of 2 within the range [-202, 24].
    msg3_delta_power: 8                     # Optional INT (8). Sets the target power level at the network receiver side, in dBm. Supported: multiples of 2 within the range [-6, 8].
    max_puschs_per_slot: 16                 # Optional UINT (16). Sets the maximum number of PUSCH grants per slot. Supported: [1 - 16].
    beta_offset_ack_idx_1: 11               # Optional UINT (11). Sets the betaOffsetACK-Index1 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_ack_idx_2: 6                # Optional UINT (6). Sets the betaOffsetACK-Index2 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_ack_idx_3: 4                # Optional UINT (4). Sets the betaOffsetACK-Index3 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_csi_p1_idx_1: 13            # Optional UINT (13). Sets the betaOffsetCSI-Part1-Index1 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_csi_p1_idx_2: 10            # Optional UINT (10). Sets the betaOffsetCSI-Part1-Index2 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_csi_p2_idx_1: 13            # Optional UINT (13). Sets the betaOffsetCSI-Part2-Index1 part of UCI-OnPUSCH. Supported: [0 - 31].
    beta_offset_csi_p2_idx_2: 10            # Optional UINT (10). Sets the betaOffsetCSI-Part2-Index2 part of UCI-OnPUSCH. Supported: [0 - 31].
    min_k2: 4                               # Optional UINT (4). Sets the minimum value of K2 (difference in slots between PDCCH and PUSCH). Supported: [1 - 4].
    dc_offset:                              # Optional TEXT. Sets the Direct Current offset in number of subcarriers. Supported: [-1650 - 1649] OR [outside, undetermined, center].
    olla_snr_inc_step: 0.001                # Optional FLOAT (0.001). Sets the outer-loop link adaptation (OLLA) increment value. 0 disables OLLA. Supported: [0 - 1].
    olla_target_bler: 0.01                  # Optional FLOAT (0.01). Sets the target UL BLER in the OLLA algorithm. Supported: [0 - 0.5].
```

```
    olla_max_snr_offset: 5                  # Optional FLOAT (5). Sets the maximum SNR offset that OLLA can apply. Supported: positive float.
    dmrs_additional_position: 2             # Optional UINT (2). Sets the PUSCH DMRS additional position. Supported: [0 - 3].
    min_rb_size: 1                          # Optional UINT (1). Sets the minimum RB size for UE PUSCH resource allocation. Supported: [1 - 275].
    max_rb_size: 275                        # Optional UINT (275). Sets the maximum RB size for UE PUSCH resource allocation. Supported: [1 - 275].
    start_rb: 0                             # Optional UINT (0). Sets the start RB for resource allocation of UE PUSCHs. Supported: [0 - 275].
    end_rb: 275                             # Optional UINT (275). Sets the end RB for resource allocation of UE PUSCHs. Supported: [0 - 275].
    enable_cl_loop_pw_control: false        # Optional BOOLEAN (false). Enables the closed-loop power control for PUSCH. Supported: [false, true].
    enable_phr_bw_adaptation: false         # Optional BOOLEAN (false). Enables bandwidth adaptation to prevent negative PHR. Supported: [false, true].
    target_sinr: 10                         # Optional FLOAT (10). Sets the target SINR in dB. Supported: [-5 - 30].
    ref_path_loss: 70                       # Optional FLOAT (70). Sets the reference path-loss for target PUSCH SINR in dB. Supported: [50 - 120].
    pl_compensation_factor: 1               # Optional FLOAT (1). Sets the fractional path-loss compensation factor in PUSCH power control. Supported: [0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1].
    enable_transform_precoding: false       # Optional BOOLEAN (false). Enables transform precoding for PUSCH. Supported: [false, true].

  pucch:
    p0_nominal: -90                         # Optional INT (-90). Sets the power control parameter P0 for PUCCH transmissions in dBm. Supported: multiples of 2 within [-202, 24].
    pucch_resource_common:                  # Optional UINT. Sets the index of PUCCH resource set for common configuration. Supported: [0 - 15].
    sr_period_ms: 20                        # Optional FLOAT (20). Sets the SR period in milliseconds. Supported: [1, 2, 2.5, 4, 5, 8, 10, 16, 20, 40, 80, 160, 320].
    formats: f1_and_f2                      # Optional TEXT (f1_and_f2). Sets the PUCCH format combinations to use. Supported: [f0_and_f2, f1_and_f2, f1_and_f3, f1_and_f4].
    resource_set_size: 8                    # Optional UINT (8). Sets the number of PUCCH resources per UE for HARQ for each PUCCH resource set. Supported: [1 - 8].
    nof_cell_res_set_configs: 2             # Optional UINT (2). Sets the number of PUCCH resource set configurations per cell. Supported: [1 - 10].
    nof_cell_sr_res: 8                      # Optional UINT (8). Sets the number of PUCCH F0 or F1 resources per cell for SR. Supported: [1 - 100].
    nof_cell_csi_res: 8                     # Optional UINT (8). Sets the number of PUCCH F2, F3 or F4 resources per cell for CSI. Supported: [0 - 100].
    f0_intraslot_freq_hop: false            # Optional BOOLEAN (false). Enables intra-slot frequency hopping for PUCCH F0. Supported: [false, true].
    f1_enable_occ: false                    # Optional BOOLEAN (false). Enables OCC for PUCCH F1. Supported: [false, true].
    f1_nof_cyclic_shifts: 2                 # Optional UINT (2). Sets the number of possible cyclic shifts for PUCCH F1 resources. Supported: [1, 2, 3, 4, 6, 12].
    f1_intraslot_freq_hop: false            # Optional BOOLEAN (false). Enables intra-slot frequency hopping for PUCCH F1. Supported: [false, true].
    f2_max_nof_rbs: 1                       # Optional UINT (1). Sets the max number of RBs for PUCCH F2 resources. Supported: [1 - 16].
    f2_max_payload:                         # Optional UINT. Sets the min required payload capacity in bits for PUCCH F2 resources. Supported: [4 - 40].
    f2_max_code_rate: dot35                 # Optional TEXT (dot35). Sets the PUCCH F2 max code rate. Supported: [dot08, dot15, dot25, dot35, dot45, dot60, dot80].
```

```
    f2_intraslot_freq_hop: false            # Optional BOOLEAN (false). Enables intra-slot frequency hopping for PUCCH F2. Supported: [false, true].
    f3_max_nof_rbs: 1                       # Optional UINT (1). Sets the max number of RBs for PUCCH F3 resources. Supported: [1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16].
    f3_max_payload:                         # Optional UINT. Sets the min required payload capacity in bits for PUCCH F3 resources. Supported: [4 - 40].
    f3_max_code_rate: dot35                 # Optional TEXT (dot35). Sets the PUCCH F3 max code rate. Supported: [dot08, dot15, dot25, dot35, dot45, dot60, dot80].
    f3_intraslot_freq_hop: false            # Optional BOOLEAN (false). Enables intra-slot frequency hopping for PUCCH F3. Supported: [false, true].
    f3_additional_dmrs: false               # Optional BOOLEAN (false). Enables additional DMRS for PUCCH F3. Supported: [false, true].
    f3_pi2_bpsk: false                      # Optional BOOLEAN (false). Enables pi/2 BPSK for PUCCH F3. Supported: [false, true].
    f4_max_code_rate: dot35                 # Optional TEXT (dot35). Sets the PUCCH F4 max code rate. Supported: [dot08, dot15, dot25, dot35, dot45, dot60, dot80].
    f4_intraslot_freq_hop: false            # Optional BOOLEAN (false). Enables intra-slot frequency hopping for PUCCH F4. Supported: [false, true].
    f4_additional_dmrs: false               # Optional BOOLEAN (false). Enables additional DMRS for PUCCH F4. Supported: [false, true].
    f4_pi2_bpsk: false                      # Optional BOOLEAN (false). Enables pi/2 BPSK for PUCCH F4. Supported: [false, true].
    f4_occ_length: 2                        # Optional UINT (2). Sets the OCC length for PUCCH F4. Supported: [2, 4].
    ema_alpha_cl_pw_control_sinr: 0.5       # Optional FLOAT (0.5). Sets the EMA smoothing factor alpha for PUCCH closed-loop power control SINR. Supported: (0, 1) exclusive.
    repetition_sinr_thresholds:             # Optional FLOAT. Sets the SINR thresholds (dB) mapping UE SINR to PUCCH repetition factor. Up to 3 entries.
    repetition_factors:                     # Optional UINT. Sets the repetition factor for each PUCCH resource in a HARQ-ACK resource set. Supported: [1, 2, 4, 8].
    f4_enable_occ: false                    # Optional BOOLEAN (false). Enables OCC multiplexing for PUCCH F4. Supported: [false, true].
    min_k1: 4                               # Optional UINT (4). Sets the minimum value of K1 (difference in slots between PDSCH and HARQ-ACK). Supported: [1 - 4].
    max_consecutive_kos: 100                # Optional UINT (100). Sets the maximum number of consecutive undecoded PUCCH F2 for CSI before an RLF is reported.
    enable_cl_loop_pw_control: false        # Optional BOOLEAN (false). Enables the closed-loop power control for PUCCH. Supported: [false, true].
    target_sinr_f0: 10                      # Optional FLOAT (10). Sets the target SINR for PUCCH F0 in dB. Supported: [-10 - 20].
    target_sinr_f2: 6                       # Optional FLOAT (6). Sets the target SINR for PUCCH F2 in dB. Supported: [-10 - 20].
    target_sinr_f3: 1                       # Optional FLOAT (1). Sets the target SINR for PUCCH F3 in dB. Supported: [-15 - 10].

  srs:
    type_enabled: disabled                  # Optional TEXT (disabled). Enables/disables SRS and sets resource type. Supported: [disabled, periodic, aperiodic].
    period_ms: 20                           # Optional FLOAT (20). Sets the periodic SRS period in ms. Supported: [1, 2, 2.5, 4, 5, 8, 10, 16, 20, 32, 40, 64, 80, 160, 320, 640, 1280, 2560].
    max_nof_sym_per_slot: 2                 # Optional UINT (2). Sets the number of symbols for UL slot reserved for SRS cell resources. Supported: [1 - 6].
    nof_sym_per_resource: 1                 # Optional UINT (1). Sets the number of symbols per SRS resource. Supported: [1, 2, 4].
```

```
    c_srs:                                  # Optional UINT. Sets the C_SRS parameter for SRS. If not set, computed automatically. Supported: [0 - 63].
    freq_domain_shift: 0                    # Optional UINT (0). Sets the SRS frequency domain shift. Supported: [0 - 268].
    tx_comb: 4                              # Optional UINT (4). Sets the SRS TX comb size. Supported: [2, 4].
    cyclic_shift_reuse: 1                   # Optional UINT (1). Sets the SRS cyclic shift reuse factor. Supported: [1, 2, 3, 4, 6].
    sequence_id_reuse: 1                    # Optional UINT (1). Sets the reuse factor for the SRS sequence ID. Supported: [1, 2, 3, 5, 6, 10, 15, 30].
    p0: -84                                 # Optional INT (-84). Sets the P0 value for SRS in dBm. Supported: multiples of 2 within [-202, 24].

  prach:
    prach_config_index:                     # Optional UINT. Sets the PRACH configuration index. If not set, derived automatically. Supported: [0 - 255].
    prach_root_sequence_index: 1            # Optional UINT (1). Sets the PRACH Root Sequence Index (RSI). Supported: [0 - 837].
    zero_correlation_zone: 0                # Optional UINT (0). Sets the Zero Correlation Zone. Supported: [0 - 15].
    fixed_msg3_mcs: 0                       # Optional UINT (0). Sets a fixed Msg3 MCS. Supported: [0 - 28].
    max_msg3_harq_retx: 4                   # Optional UINT (4). Sets the maximum number of Msg3 HARQ retransmissions. Supported: [0 - 4].
    total_nof_ra_preambles: 64              # Optional UINT (64). Sets the number of different contention-based PRACH preambles per occasion. Supported: [1 - 64].
    cfra_enabled: false                     # Optional BOOLEAN (false). Enables Contention Free Random Access (CFRA). Supported: [false, true].
    prach_frequency_start:                  # Optional INT. Sets the offset of lowest PRACH transmission occasion in frequency domain in PRBs. Supported: [0 - (MAX_NOF_PRB - 1)].
    preamble_rx_target_pw: -100             # Optional INT (-100). Sets the target power level at the network receiver side, in dBm. Supported: multiples of 2 within [-202, -60].
    preamble_trans_max: 7                   # Optional UINT (7). Sets the max number of RA preamble transmissions before declaring a failure. Supported: [3, 4, 5, 6, 7, 8, 10, 20, 50, 100, 200].
    power_ramping_step_db: 4                # Optional UINT (4). Sets the power ramping step for PRACH. Supported: [0, 2, 4, 6].
    ports: 0                                # Optional UINT (0). Sets the list of antenna ports.
    nof_ssb_per_ro: 1                       # Optional UINT (1). Sets the number of SSBs per RACH occasion. Supported: [1].
    nof_cb_preambles_per_ssb: 64            # Optional UINT (64). Sets the number of contention based preambles per SSB. Supported: [1 - 64].
    nof_prach_guardbands_rbs: 3             # Optional UINT (3). Sets the number of RBs used as guardband on each side of the PRACH RBs interval. Supported: [1 - 10].
    slice_based_ra_prioritization:          # Optional TEXT. Sets the list of configurations for slice-based RA prioritization.
    ra_resp_window:                         # Optional UINT. Sets the RA-response window length in number of slots. Supported: [1, 2, 4, 8, 10, 20, 40, 80].
    two_step:
      cb_preambles_per_ssb_per_shared_ro: 4   # Optional UINT (4). Sets the number of CB preambles per SSB per shared RACH occasion for 2-step RA. Supported: [1 - 60].
      msgA_rsrp_thres_dbm: -100               # Optional INT (-100). Sets the RSRP threshold in dBm for UE to select 2-step RA over 4-step RA. Supported: [-156 - -29].
```

```
      msgB_response_window_slots: 40          # Optional UINT (40). Sets the MsgB response window length in slots. Supported: [1, 2, 4, 8, 10, 20, 40, 80, 160, 320].
      td_offset: 1                            # Optional UINT (1). Sets the time-domain offset in slots from PRACH slot to MsgA PUSCH slot. Supported: [1 - 32].
      pusch_td_res_index: 0                   # Optional UINT (0). Sets the index into the PUSCH-TimeDomainAllocationResource table for MsgA PUSCH scheduling.
      mcs: 0                                  # Optional UINT (0). Sets the MCS index for MsgA PUSCH transmission. Supported: [0 - 28].
      nof_prbs_per_msgA_po: 3                 # Optional UINT (3). Sets the number of PRBs per MsgA PUSCH occasion. Supported: [1 - 32].
      prb_start: 0                            # Optional UINT (0). Sets the frequency offset in PRBs of the lowest MsgA PUSCH occasion from PRB 0.
      po_fdm: 1                               # Optional UINT (1). Sets the number of MsgA PUSCH occasions FDMed in one time instance. Supported: [1, 2, 4, 8].

  tdd_ul_dl_cfg:
    dl_ul_tx_period: 10                     # Optional INT (10). Sets the TDD pattern periodicity in slots. Supported: [2 - 80].
    nof_dl_slots: 6                         # Optional INT (6). Sets the number of consecutive full Downlink slots. Supported: [0 - 80].
    nof_dl_symbols: 8                       # Optional INT (8). Sets the number of Downlink symbols at the beginning of the slot following full Downlink slots. Supported: [0 - 13].
    nof_ul_slots: 3                         # Optional INT (3). Sets the number of consecutive full Uplink slots. Supported: [0 - 80].
    nof_ul_symbols: 0                       # Optional INT (0). Sets the number of Uplink symbols at the end of the slot preceding the first full Uplink slot. Supported: [0 - 13].
    pattern2:
      dl_ul_tx_period: 10                       # Optional INT (10). Sets the second TDD pattern periodicity in slots. Supported: [2 - 80].
      nof_dl_slots: 6                           # Optional INT (6). Sets the number of consecutive full Downlink slots in the second pattern. Supported: [0 - 80].
      nof_dl_symbols: 8                         # Optional INT (8). Sets the number of Downlink symbols in the special slot of the second pattern. Supported: [0 - 13].
      nof_ul_slots: 3                           # Optional INT (3). Sets the number of consecutive full Uplink slots in the second pattern. Supported: [0 - 80].
      nof_ul_symbols: 0                         # Optional INT (0). Sets the number of Uplink symbols in the special slot of the second pattern. Supported: [0 - 13].

  paging:
    pg_search_space_id: 1                   # Optional UINT (1). Sets the SearchSpace to use for Paging.
    default_pg_cycle_in_rf: 128             # Optional UINT (128). Sets the default Paging cycle in number of Radio Frames. Supported: [32, 64, 128, 256].
    nof_pf_per_paging_cycle: oneT           # Optional TEXT (oneT). Sets the number of paging frames per DRX cycle. Supported: [oneT, halfT, quarterT, oneEighthT, oneSixteethT].
    pf_offset: 0                            # Optional UINT (0). Sets the paging frame offset.
    nof_po_per_pf: 1                        # Optional UINT (1). Sets the number of paging occasions per paging frame. Supported: [1, 2, 4].
    edrx_enabled: false                     # Optional BOOLEAN (false). Enables eDRX. Supported: [false, true].

  csi:
    csi_rs_enabled: true                    # Optional BOOLEAN (true). Enables CSI-RS resources and CSI reporting. Supported: [false, true].
```

```
    csi_rs_period: 20                       # Optional UINT (20). Sets the CSI-RS period in milliseconds. Supported: [10, 20, 40, 80].
    report_type: periodic                   # Optional TEXT (periodic). Sets the type of CSI reporting configuration. Supported: [periodic, aperiodic].
    meas_csi_rs_slot_offset:                # Optional UINT. Sets the slot offset of first CSI-RS resource used for measurement.
    tracking_csi_rs_slot_offset:            # Optional UINT. Sets the slot offset of the first CSI-RS resource used for tracking.
    zp_csi_rs_slot_offset:                  # Optional UINT. Sets the slot offset of the ZP CSI-RS resources.
    pwr_ctrl_offset: 0                      # Optional INT (0). Sets the power offset of PDSCH RE to NZP CSI-RS RE in dB. Supported: [-8 - 15].

  scheduler:
    nof_preselected_newtx_ues: 1024         # Optional UINT (1024). Sets the number of UEs pre-selected for potential newTx allocations per slot. Supported: [1 - 8192].
    policy:
      qos_sched:
        combine_function: gbr_prioritized   # Optional TEXT (gbr_prioritized). Sets the QoS-aware scheduler weight combining function. Supported: [gbr_prioritized, geometric_mean].
        pf_fairness_coeff: 2                # Optional FLOAT (2). Sets the fairness coefficient for the Proportional Fair policy scheduler.
        prio_enabled: true                  # Optional BOOLEAN (true). Enables the QoS Flow priority in QoS-aware scheduling. Supported: [false, true].
        pdb_enabled: true                   # Optional BOOLEAN (true). Enables the Packet Delay Budget (PDB) in QoS-aware scheduling. Supported: [false, true].
        gbr_enabled: true                   # Optional BOOLEAN (true). Enables the Guaranteed Bit Rate (GBR) in QoS-aware scheduling. Supported: [false, true].
      rr_sched:                             # Optional. Round-robin scheduler policy configuration.

  ta:
    ta_measurement_slot_period: 80              # Optional UINT (80). Sets the measurements periodicity in nof. slots over which the new Timing Advance Command is computed.
    ta_measurement_slot_prohibit_period: 0      # Optional UINT (0). Sets the delay in number of slots between issuing the TA_CMD and starting TA measurements. Supported: [0 - 10000].
    ta_cmd_offset_threshold: 1                  # Optional INT (1). Sets the TA offset threshold above which a Timing Advance Command is triggered. -1 disables. Supported: [-1 - 31].
    ta_target: 1.0                              # Optional FLOAT (1.0). Sets the Timing Advance target in units of TA. Supported: [-30.0 - 30.0].
    ta_update_measurement_ul_sinr_threshold: 0  # Optional FLOAT (0). Sets the UL SINR threshold in dB above which reported N_TA update measurements are considered valid.
    ta_outlier_detection_zscore_threshold: 1.75 # Optional FLOAT (1.75). Sets the Z-score threshold for outlier detection in N_TA measurements. Supported: [0.0 - 5.0].

  rlm:
    rlm_resource_type: default_type         # Optional TEXT. Sets the Radio Link Monitoring resource detection type. Supported: [default_type, ssb, csi_rs, ssb_and_csi_rs].

  drx:
    on_duration_timer: 10                   # Optional UINT (10). Sets the minimum duration in ms that the UE stays in active mode when DRX is configured. Supported:
```

```
    [1, 2, 3, 4, 5, 6, 8, 10, 20, 30, 40, 50, 60, 80, 100, 200, 300, 400, 500, 600,
    800, 1000, 1200, 1600].
611     inactivity_timer: 0                     # Optional UINT (0). Sets the duration
    in ms that the UE stays active after PDCCH reception when DRX is configured.
    Supported: [0, 1, 2, 3, 4, 5, 6, 8, 10, 20, 30, 40, 50, 60, 80, 100, 200, 300, 500,
    750, 1280, 1920, 2560].
612     retx_timer_dl: 0                        # Optional UINT (0). Sets the maximum
    duration in slots until a DL ReTX is received by the UE when DRX is configured.
    Supported: [0, 1, 2, 4, 6, 8, 16, 24, 33, 40, 64, 80, 96, 112, 128, 160, 320].
613     retx_timer_ul: 0                        # Optional UINT (0). Sets the maximum
    duration in slots until a grant for UL ReTX is received by the UE when DRX is
    configured. Supported: [0, 1, 2, 4, 6, 8, 16, 24, 33, 40, 64, 80, 96, 112, 128,
    160, 320].
614     long_cycle: 0                           # Optional UINT (0). Sets the duration
    in ms between UE DRX long cycles. 0 disables DRX. Supported: [10, 20, 32, 40, 60,
    64, 70, 80, 128, 160, 256, 320, 512, 640, 1024, 1280, 2048, 2560, 5120, 10240] OR
    [0].
615
616   ntn:
617     cell_specific_koffset: 1                # Optional UINT (1). Sets the cell-
    specific K-offset for NTN timing advance in ms. Supported: [1 - 1023].
618     ntn_ul_sync_validity_dur: 60            # Optional UINT (60). Sets the UL sync
    validity duration in seconds. Supported: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50,
    55, 60, 120, 180, 240, 900].
619     propagator_type: rk4                    # Optional TEXT (rk4). Sets the orbit
    propagator type. Supported: [rk4, keplerian].
620     ta_report: false                        # Optional BOOLEAN (false). Enables TA
    reporting in SIB19. Supported: [false, true].
621     use_state_vector: true                  # Optional BOOLEAN (true). Broadcasts
    ephemeris as ECEF state vectors (true) or orbital parameters (false). Supported:
    [false, true].
622     epoch_timestamp:                        # Optional TEXT. Sets the epoch
    timestamp for NTN assistance info. Format: Unix time in ms or YYYY-MM-
    DDTHH:MM:SS[.mmm].
623     epoch_sfn_offset: 0                     # Optional UINT (0). Sets the optional
    SFN offset between SIB19 tx slot and epoch time.
624     distance_threshold: 0                   # Optional UINT (0). Sets the distance
    from serving cell reference location for location-based measurement in meters.
    Supported: [0 - 3276250].
625     t_service:                              # Optional TEXT. Sets the end of
    service time for the current cell. Format: Unix time in ms or YYYY-MM-
    DDTHH:MM:SS[.mmm].
626     epoch_time:
627       sfn: 0                                # Optional UINT (0). Sets the SFN part
    of the epoch time. Supported: [0 - 1023].
628       subframe_number: 0                    # Optional UINT (0). Sets the sub-frame
    number of the epoch time. Supported: [0 - 9].
629     ta_info:
630       ta_common: 0                          # Optional FLOAT (0). Sets the common
    Timing Advance value in microseconds. Supported: [0 - 270730].
631       ta_common_drift: 0                    # Optional FLOAT (0). Sets the drift
    rate of common TA. Supported: [-51.4606 - 51.4606].
632       ta_common_drift_variant: 0            # Optional FLOAT (0). Sets the drift
    rate variation of common TA. Supported: [0 - 0.57898].
633       ta_common_offset: 0                   # Optional FLOAT (0). Sets the constant
    offset added to TA common. Supported: [0 - 10000].
634     feeder_link:
635       enable_doppler_compensation: false    # Optional BOOLEAN (false). Enables
    feeder link Doppler compensation. Supported: [false, true].
```

```
      dl_freq: 0                          # Optional FLOAT (0). Sets the downlink
feeder link frequency (gNB to satellite) in Hz. Supported: [0 - 100e9].
      ul_freq: 0                          # Optional FLOAT (0). Sets the uplink
feeder link frequency (satellite to gNB) in Hz. Supported: [0 - 100e9].
    ephemeris_info_ecef:
      pos_x: 0                            # Optional FLOAT (0). Sets the X
position of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
      pos_y: 0                            # Optional FLOAT (0). Sets the Y
position of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
      pos_z: 0                            # Optional FLOAT (0). Sets the Z
position of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
      vel_x: 0                            # Optional FLOAT (0). Sets the X
velocity of the satellite in m/s. Supported: [-7864.32 - 7864.26].
      vel_y: 0                            # Optional FLOAT (0). Sets the Y
velocity of the satellite in m/s. Supported: [-7864.32 - 7864.26].
      vel_z: 0                            # Optional FLOAT (0). Sets the Z
velocity of the satellite in m/s. Supported: [-7864.32 - 7864.26].
    ephemeris_orbital:
      semi_major_axis: 0                  # Optional FLOAT (0). Sets the semi-
major axis of the satellite in meters. Supported: [6500000 - 42998632.07].
      eccentricity: 0                     # Optional FLOAT (0). Sets the
eccentricity of the satellite. Supported: [0 - 0.01500510825].
      periapsis: 0                        # Optional FLOAT (0). Sets the argument
of periapsis in radians. Supported: [0 - 6.28407400155].
      longitude: 0                        # Optional FLOAT (0). Sets the
longitude of ascending node in radians. Supported: [0 - 6.28407400155].
      inclination: 0                      # Optional FLOAT (0). Sets the orbital
inclination in radians. Supported: [-1.57101850624 - 1.57101848283].
      mean_anomaly: 0                     # Optional FLOAT (0). Sets the mean
anomaly in radians. Supported: [0 - 6.28407400155].
    gateway_location:
      latitude: 0                         # Optional FLOAT (0). Sets the gateway
latitude in degrees. Supported: [-90 - 90].
      longitude: 0                        # Optional FLOAT (0). Sets the gateway
longitude in degrees. Supported: [-180 - 180].
      altitude: 0                         # Optional FLOAT (0). Sets the gateway
altitude in meters. Supported: [-1000 - 20000].
    reference_location:
      latitude: 0                         # Optional FLOAT (0). Sets the cell
reference latitude in degrees. Supported: [-90 - 90].
      longitude: 0                        # Optional FLOAT (0). Sets the cell
reference longitude in degrees. Supported: [-180 - 180].
    moving_ref_location:
      latitude: 0                         # Optional FLOAT (0). Sets the moving
reference latitude in degrees. Supported: [-90 - 90].
      longitude: 0                        # Optional FLOAT (0). Sets the moving
reference longitude in degrees. Supported: [-180 - 180].
    polarization:
      dl: linear                          # Optional TEXT (linear). Sets the DL
service link polarization. Supported: [lhcp, rhcp, linear].
      ul: linear                          # Optional TEXT (linear). Sets the UL
service link polarization. Supported: [lhcp, rhcp, linear].
    sat_switch_with_resync:
      epoch_timestamp:                    # Optional TEXT. Sets the epoch
timestamp for satellite switch. Format: Unix time in ms or YYYY-MM-
DDTHH:MM:SS[.mmm].
      t_service_start:                    # Optional TEXT. Sets the time when the
target satellite starts serving. Format: Unix time in ms or YYYY-MM-
DDTHH:MM:SS[.mmm].
```

```
      ssb_time_offset_sf: 0                 # Optional UINT (0). Sets the SSB time offset in subframes. Supported: [0 - 159].
      ntn_gateway_location:
        latitude: 0                         # Optional FLOAT (0). Sets the gateway latitude. Supported: [-90 - 90].
        longitude: 0                        # Optional FLOAT (0). Sets the gateway longitude. Supported: [-180 - 180].
        altitude: 0                         # Optional FLOAT (0). Sets the gateway altitude in meters. Supported: [-1000 - 20000].

e2:
  enable_cu_cp_e2: false                # Optional BOOLEAN (false). Enables the E2 agent for CU-CP. Supported: [false, true].
  enable_cu_up_e2: false                # Optional BOOLEAN (false). Enables the E2 agent for CU-UP. Supported: [false, true].
  enable_du_e2: false                   # Optional BOOLEAN (false). Enables the E2 agent for DU. Supported: [false, true].
  addrs: 127.0.0.1                      # Optional TEXT (127.0.0.1). Sets the RIC IP addresses. Multiple addresses can be specified for SCTP multi-homing. Format: IPV4 or IPV6 IP address.
  port: 36421                           # Optional UINT (36421). Sets the RIC port. Supported: [20000 - 40000].
  bind_addrs: 127.0.0.1                 # Optional TEXT (127.0.0.1). Sets the local bind addresses for the E2 interface. Multiple addresses can be specified for SCTP multi-homing. Format: IPV4 or IPV6 IP address.
  sctp_rto_initial: 120                 # Optional INT (120). Sets the initial retransmission timeout when creating the SCTP connection.
  sctp_rto_min: 120                     # Optional INT (120). Sets the minimum retransmission timeout for the SCTP connection.
  sctp_rto_max: 500                     # Optional INT (500). Sets the maximum retransmission timeout for the SCTP connection.
  sctp_init_max_attempts: 3             # Optional INT (3). Sets the maximum retransmission attempts for the initial SCTP connection.
  sctp_max_init_timeo: 500              # Optional INT (500). Sets the maximum retransmission timeout for the initial SCTP connection.
  sctp_hb_interval: 30000               # Optional INT (30000). Sets the SCTP heartbeat interval in milliseconds. -1 uses system default.
  sctp_assoc_max_retx: 10               # Optional INT (10). Sets the SCTP association max retransmissions. -1 uses system default.
  sctp_nodelay: false                   # Optional BOOLEAN (false). Enables sending SCTP messages without Nagle-like delay. Supported: [false, true].
  e2sm_kpm_enabled: false               # Optional BOOLEAN (false). Enables the E2SM KPM service module. Supported: [false, true].
  e2sm_rc_enabled: false                # Optional BOOLEAN (false). Enables the E2SM RC service module. Supported: [false, true].
  e2sm_ccc_enabled: false               # Optional BOOLEAN (false). Enables the E2SM CCC service module. Supported: [false, true].

ntn:
  cell_specific_koffset:                 # Optional UINT. Sets the cell-specific k-offset for NTN timing advance in ms. Supported: [1 - 1023].
  ntn_ul_sync_validity_dur:              # Optional UINT. Sets the NTN UL sync validity duration in seconds. Supported: [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 120, 180, 240, 900].
  epoch_time:
    sfn: 0                               # Optional UINT (0). Sets the SFN part. Supported: [0 - 1023].
    subframe_number: 0                   # Optional UINT (0). Sets the sub-frame number part. Supported: [0 - 9].
  ta_info:
```

```
    ta_common: 0                        # Optional FLOAT (0). Sets the common
Timing Advance value in microseconds. Supported: [0 - 270730].
    ta_common_drift: 0                  # Optional FLOAT (0). Sets the drift rate
of common TA. Supported: [-51.4606 - 51.4606].
    ta_common_drift_variant: 0          # Optional FLOAT (0). Sets the drift rate
variation of common TA. Supported: [0 - 0.57898].
    ta_common_offset: 0                 # Optional FLOAT (0). Sets the constant
offset added to TA common. Supported: [0 - 10000].
  ephemeris_info_ecef:
    pos_x: 0                            # Optional FLOAT (0). Sets the X position
of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
    pos_y: 0                            # Optional FLOAT (0). Sets the Y position
of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
    pos_z: 0                            # Optional FLOAT (0). Sets the Z position
of the satellite in meters. Supported: [-43620761.6 - 43620759.3].
    vel_x: 0                            # Optional FLOAT (0). Sets the X velocity
of the satellite in m/s. Supported: [-7864.32 - 7864.26].
    vel_y: 0                            # Optional FLOAT (0). Sets the Y velocity
of the satellite in m/s. Supported: [-7864.32 - 7864.26].
    vel_z: 0                            # Optional FLOAT (0). Sets the Z velocity
of the satellite in m/s. Supported: [-7864.32 - 7864.26].
  ephemeris_orbital:
    semi_major_axis: 0                  # Optional FLOAT (0). Sets the semi-major
axis of the satellite in meters. Supported: [6500000 - 42998632.07].
    eccentricity: 0                     # Optional FLOAT (0). Sets the
eccentricity of the satellite. Supported: [0 - 0.01500510825].
    periapsis: 0                        # Optional FLOAT (0). Sets the argument
of periapsis in radians. Supported: [0 - 6.28407400155].
    longitude: 0                        # Optional FLOAT (0). Sets the longitude
of the satellite's ascending node in radians. Supported: [0 - 6.28407400155].
    inclination: 0                      # Optional FLOAT (0). Sets the orbital
inclination in radians. Supported: [-1.57101850624 - 1.57101848283].
    mean_anomaly: 0                     # Optional FLOAT (0). Sets the mean
anomaly in radians. Supported: [0 - 6.28407400155].
  polarization:
    dl:                                 # Optional TEXT. Sets the DL service link
polarization. Supported: [lhcp, rhcp, linear].
    ul:                                 # Optional TEXT. Sets the UL service link
polarization. Supported: [lhcp, rhcp, linear].
  ta_report:                            # Optional BOOLEAN. Enables TA reporting
in SIB19. Supported: [false, true].

custom_freq_bands:                    # Optional TEXT. Defines non-standard NR
frequency bands for use with dl_arfcn. Each list entry should begin with "-".
  -
    band_nr: 999                        # Required UINT. Sets the NR band number.
    dl_freq_min: 3400                   # Required FLOAT. Sets the DL frequency
range start in MHz.
    dl_freq_max: 3600                   # Required FLOAT. Sets the DL frequency
range end in MHz.
    ul_freq_min: 3400                   # Optional FLOAT. Sets the UL frequency
range start in MHz. Omit for TDD.
    ul_freq_max: 3600                   # Optional FLOAT. Sets the UL frequency
range end in MHz. Omit for TDD.
    f_raster: 15                        # Required UINT. Sets the ARFCN raster in
kHz. Supported: [15, 30, 60, 100, 120].
    ssb_scs: 30                         # Required UINT. Sets the SSB subcarrier
spacing in kHz. Supported: [15, 30, 120, 240].
    ssb_case_c: false                   # Optional BOOLEAN (false). Sets SSB
pattern case C. Use true for n77/n78/n79-like TDD bands. Supported: [false,
```

```
true].
733     min_40mhz_bw: false                 # Optional BOOLEAN (false). Sets the
minimum 40 MHz bandwidth constraint. Supported: [false, true].
734     delta_gscn: 1                       # Optional UINT (1). Sets the GSCN step for
SSB candidate search. Supported: [1, 3, 7, 16].
735     ntn: false                          # Optional BOOLEAN (false). Sets whether
this is a Non-Terrestrial Network band. Supported: [false, true].
736
737 ru_ofh:
738 # Many of the following values are optional as they have default values. In
practice, all of the following parameters should be defined by the user, as they
will need
739 # to be configured specifically for the RU being used. Failing to configure these
parameters correctly may result in the RU failing to connect to the DU.
740   gps_alpha: 0                                          # Optional UINT (0).
Sets the GPS alpha. Supported: [0 - 1.2288e+07].
741   gps_beta: 0                                           # Optional INT (0). Sets
the GPS beta. Supported: [-32768 - +32767].
742   ru_bandwidth_MHz: <cell bandwidth>                      # Optional TEXT (<cell
bandwidth>). Sets the channel bandwidth in MHz. Supported: [5, 10, 15, 20, 25, 30,
40, 50, 60, 70, 80, 90, 100].
743   t1a_max_cp_dl: 500                                    # Optional UINT (500).
Sets the T1a maximum value for downlink control-plane in microseconds. Supported:
[0 - 1960].
744   t1a_min_cp_dl: 258                                    # Optional UINT (258).
Sets the T1a minimum value for downlink control-plane in microseconds. Supported:
[0 - 1960].
745   t1a_max_cp_ul: 500                                    # Optional UINT (500).
Sets the T1a maximum value for uplink control-plane in microseconds. Supported: [0
- 1960].
746   t1a_min_cp_ul: 285                                    # Optional UINT (285).
Sets the T1a minimum value for uplink control-plane in microseconds. Supported: [0
- 1960].
747   t1a_max_up: 300                                       # Optional UINT (300).
Sets the T1a maximum value for user-plane in microseconds. Supported: [0 - 1960].
748   t1a_min_up: 85                                        # Optional UINT (85).
Sets the T1a minimum value for user-plane in microseconds. Supported: [0 - 1960].
749   ta4_max: 500                                          # Optional UINT (500).
Sets the Ta4 maximum value for User-Plane in microseconds. Supported: [0 - 1960].
750   ta4_min: 85                                           # Optional UINT (85).
Sets the Ta4 minimum value for User-Plane in microseconds. Supported: [0 - 1960].
751   is_prach_cp_enabled: true                             # Optional BOOLEAN
(true). Sets the PRACH control-plane enabled flag. Supported: [false, true].
752   ignore_ecpri_seq_id: false                            # Optional BOOLEAN
(false). Sets whether to ignore the eCPRI sequence ID field value. Supported:
[false, true].
753   ignore_ecpri_payload_size: false                      # Optional BOOLEAN
(false). Sets whether to ignore the eCPRI payload size field value. Supported:
[false, true].
754   ignore_prach_start_symbol: false                      # Optional BOOLEAN
(false). Sets whether to ignore the PRACH start symbol in PRACH U-Plane packets.
Supported: [false, true].
755   log_lates_as_warnings: true                           # Optional BOOLEAN
(true). Sets late Open Fronthaul events to be logged as warnings. Supported:
[false, true].
756   warn_unreceived_ru_frames: after_traffic_detection    # Optional TEXT
(after_traffic_detection). Sets the policy for warning about unreceived Radio Unit
frames. Supported: [never, always, after_traffic_detection].
757   compr_method_ul: bfp                                  # Optional TEXT (bfp).
Sets the uplink compression method. Supported: [none, bfp, bfp selective, block
```

```
    scaling, mu law, modulation, modulation selective].
758   compr_bitwidth_ul: 9                                  # Optional UINT (9).
    Sets the uplink compression bit width. Supported: [1 - 16].
759   compr_method_dl: bfp                                  # Optional TEXT (bfp).
    Sets the downlink compression method. Supported: [none, bfp, bfp selective, block
    scaling, mu law, modulation, modulation selective].
760   compr_bitwidth_dl: 9                                  # Optional UINT (9).
    Sets the downlink compression bit width. Supported: [1 - 16].
761   compr_method_prach: bfp                               # Optional TEXT (bfp).
    Sets the PRACH compression method. Supported: [none, bfp, bfp selective, block
    scaling, mu law, modulation, modulation selective].
762   compr_bitwidth_prach: 9                               # Optional UINT (9).
    Sets the PRACH compression bit width. Supported: [1 - 16].
763   enable_ul_static_compr_hdr: true                      # Optional BOOLEAN
    (true). Enables the uplink static compression header. Supported: [false, true].
764   enable_dl_static_compr_hdr: true                      # Optional BOOLEAN
    (true). Enables the downlink static compression header. Supported: [false, true].
765   ru_reference_level_dBFS: -12                          # Optional FLOAT (-12).
    Sets the RU IQ reference level mapped to the maximum RF power in dBFS. Supported:
    (-inf - 0].
766   subcarrier_rms_backoff_dB: auto                       # Optional TEXT (auto).
    Sets the per-subcarrier power back-off relative to the RU reference level in dB.
    Accepts a numeric value >= 0 or 'auto'. Supported: [0 - inf, auto].
767   iq_scaling: 0.1                                       # Optional FLOAT (0.1).
    Sets the IQ scaling factor. This is an alternative to ru_reference_level_dBFS.
    Supported: [0 - 100].
768   cplane_prach_fft_len: 4096                            # Optional UINT (4096).
    Sets the PRACH FFT length used in C-Plane Type-3 messages. Supported: [128, 256,
    512, 1024, 1536, 2048, 3072, 4096].
769   cells:                                                # Optional TEXT. Sets
    the hardware-specific cell configuration on a per cell basis.
770     -                                                   # Cell 1 - set custom
    configuration for each cell
771       network_interface: enp1s0f0                       # Optional TEXT
    (enp1s0f0). Sets the ethernet network interface name for the RU or PCIe identifier
    when using DPDK.
772       enable_promiscuous: false                         # Optional BOOLEAN
    (false). Enables promiscuous mode on the network interface. Supported: [false,
    true].
773       check_link_status: false                          # Optional BOOLEAN
    (false). Enables Ethernet link status checking at startup. Supported: [false,
    true].
774       mtu: 9000                                         # Optional UINT (9000).
    Sets the NIC interface MTU size in bytes. Supported: [1500 - 9600].
775       ru_mac_addr: 70:b3:d5:e1:5b:06                    # Optional TEXT
    (70:b3:d5:e1:5b:06). Sets the Radio Unit MAC address. Format: a string, e.g.
    [AA:BB:CC:DD:11:22].
776       du_mac_addr: 00:11:22:33:00:77                    # Optional TEXT
    (00:11:22:33:00:77). Sets the Distributed Unit MAC address. Format: a string, e.g.
    [AA:BB:CC:DD:11:22].
777       vlan_tag_cp: 1                                    # Optional UINT. Sets
    the VLAN tag value for C-Plane. Supported: [1 - 4094].
778       vlan_tag_up: 1                                    # Optional UINT. Sets
    the VLAN tag value for U-Plane. Supported: [1 - 4094].
779       vlan_pcp_cp:                                      # Optional UINT. Sets
    the C-Plane VLAN PCP. Requires vlan_tag_cp. Supported: [0 - 7].
780       vlan_pcp_up:                                      # Optional UINT. Sets
    the U-Plane VLAN PCP. Requires vlan_tag_up. Supported: [0 - 7].
781       prach_port_id: [4]                                # Optional UINT ([4]).
    Sets the RU PRACH eAxC port ID(s).
```

```
      dl_port_id: [0]                                  # Optional UINT ([0]).
Sets the RU downlink eAxC port ID(s). Format: vector containing all DL eAxC ports,
e.g. [0, ..., N].
      ul_port_id: [0]                                  # Optional UINT ([0]).
Sets the RU uplink eAxC port ID(s).
      ru_bandwidth_MHz: <cell bandwidth>                 # Optional TEXT. Sets
the channel bandwidth in MHz. Supported: [5, 10, 15, 20, 25, 30, 40, 50, 60, 70,
80, 90, 100].
      t1a_max_cp_dl: 500                               # Optional UINT (500).
Sets T1a maximum value for downlink control-plane. Supported: [0 - 1960].
      t1a_min_cp_dl: 258                               # Optional UINT (258).
Sets T1a minimum value for downlink control-plane. Supported: [0 - 1960].
      t1a_max_cp_ul: 500                               # Optional UINT (500).
Sets T1a maximum value for uplink control-plane. Supported: [0 - 1960].
      t1a_min_cp_ul: 285                               # Optional UINT (285).
Sets T1a minimum value for uplink control-plane. Supported: [0 - 1960].
      t1a_max_up: 300                                  # Optional UINT (300).
Sets T1a maximum value for user-plane. Supported: [0 - 1960].
      t1a_min_up: 85                                   # Optional UINT (85).
Sets T1a minimum value for user-plane. Supported: [0 - 1960].
      ta4_max: 500                                     # Optional UINT (500).
Sets the Ta4 maximum value for User-Plane. Supported: [0 - 1960].
      ta4_min: 85                                      # Optional UINT (85).
Sets the Ta4 minimum value for User-Plane. Supported: [0 - 1960].
      is_prach_cp_enabled: true                        # Optional BOOLEAN
(true). Sets the PRACH control-plane enabled flag. Supported: [false, true].
      ignore_ecpri_seq_id: false                       # Optional BOOLEAN
(false). Sets whether to ignore the eCPRI sequence ID field value. Supported:
[false, true].
      ignore_ecpri_payload_size: false                 # Optional BOOLEAN
(false). Sets whether to ignore the eCPRI payload size field value. Supported:
[false, true].
      ignore_prach_start_symbol: false                 # Optional BOOLEAN
(false). Sets whether to ignore the PRACH start symbol in PRACH U-Plane Packets.
Supported: [false, true].
      log_lates_as_warnings: true                      # Optional BOOLEAN
(true). Sets late events to be logged as warnings. Supported: [false, true].
      warn_unreceived_ru_frames: after_traffic_detection   # Optional TEXT
(after_traffic_detection). Sets the policy for warning about unreceived RU frames.
Supported: [never, always, after_traffic_detection].
      compr_method_ul: bfp                             # Optional TEXT (bfp).
Sets the uplink compression method. Supported: [none, bfp, bfp selective, block
scaling, mu law, modulation, modulation selective].
      compr_bitwidth_ul: 9                             # Optional UINT (9).
Sets the uplink compression bit width. Supported: [1 - 16].
      compr_method_dl: bfp                             # Optional TEXT (bfp).
Sets the downlink compression method. Supported: [none, bfp, bfp selective, block
scaling, mu law, modulation, modulation selective].
      compr_bitwidth_dl: 9                             # Optional UINT (9).
Sets the downlink compression bit width. Supported: [1 - 16].
      compr_method_prach: bfp                          # Optional TEXT (bfp).
Sets the PRACH compression method. Supported: [none, bfp, bfp selective, block
scaling, mu law, modulation, modulation selective].
      compr_bitwidth_prach: 9                          # Optional UINT (9).
Sets the PRACH compression bit width. Supported: [1 - 16].
      enable_ul_static_compr_hdr: true                 # Optional BOOLEAN
(true). Enables the uplink static compression header. Supported: [false, true].
      enable_dl_static_compr_hdr: true                 # Optional BOOLEAN
(true). Enables the downlink static compression header. Supported: [false, true].
```

```
      ru_reference_level_dBFS: -12                    # Optional FLOAT (-12). Sets the RU IQ reference level mapped to the maximum RF power. Supported: (-inf - 0].
      subcarrier_rms_backoff_dB: auto                 # Optional TEXT (auto). Sets the per-subcarrier power back-off relative to the RU reference level. Accepts a numeric value >= 0 or 'auto'. Supported: [0 - inf, auto].
      iq_scaling: 0.1                                 # Optional FLOAT (0.1). Sets the IQ scaling factor. Supported: [0 - 100].
      cplane_prach_fft_len: 4096                      # Optional UINT (4096). Sets the PRACH FFT length used in C-Plane Type-3 messages. Supported: [128, 256, 512, 1024, 1536, 2048, 3072, 4096].

ru_sdr:
  srate: 61.44                          # Optional FLOAT (61.44). Sets the sampling rate of the RF-frontend in MHz.
  device_driver: uhd                    # Optional TEXT (uhd). Sets the RF device driver name.
  device_args:                          # Optional TEXT. Sets an argument that gets passed to the selected RF driver.
  tx_gain: 50.0                         # Optional FLOAT (50.0). Sets the transmit gain in dB.
  rx_gain: 60.0                         # Optional FLOAT (60.0). Sets the receive gain in dB.
  freq_offset: 0                        # Optional FLOAT (0). Sets the center frequency offset in Hertz applied to all channels.
  clock_ppm: 0                          # Optional FLOAT (0). Sets the clock calibration in Parts Per Million (PPM).
  lo_offset: 0                          # Optional FLOAT (0). Shifts the local oscillator frequency in MHz away from the center frequency to move LO leakage out of the channel.
  clock: default                        # Optional TEXT (default). Sets the RF device clock source.
  sync: default                         # Optional TEXT (default). Sets the RF device time synchronization source.
  otw_format: default                   # Optional TEXT (default). Sets the over-the-wire sample format.
  time_alignment_calibration: auto      # Optional TEXT (auto). Compensates for any reception and transmission time misalignment inherent to the RF device. Positive values reduce
                                        # the RF transmission delay with respect to RF reception. Negative values have the opposite effect.
  start_time: auto                      # Optional TEXT (auto). Sets the radio streaming start time. Format: YYYY-MM-DD HH:MM:SS. Use 'auto' or leave empty to let the radio start immediately.
  dl_freq_Hz:                           # Optional FLOAT. Sets the downlink frequency override in Hz. Overrides dl_arfcn-derived frequency.
  ul_freq_Hz:                           # Optional FLOAT. Sets the uplink frequency override in Hz. Overrides ul_arfcn-derived frequency.

  amplitude_control:
    tx_gain_backoff: 12.0                   # Optional FLOAT (12.0). Sets the baseband gain back-off in dB to account for signal PAPR. Format: positive float.
    enable_clipping: false                  # Optional BOOLEAN (false). Enables clipping of baseband samples that exceed the power ceiling. Supported: [false, true].
    ceiling: -0.1                           # Optional FLOAT (-0.1). Sets the power ceiling in dBFS relative to full scale amplitude. Format: negative float or 0.

  expert_cfg:
```

```
    low_phy_dl_throttling: 0               # Optional FLOAT (0). Throttles the lower PHY DL baseband generation. 0 disables throttling.
    tx_mode: continuous                    # Optional TEXT (continuous). Selects a radio transmission mode. Supported: [continuous, discontinuous, same-port].
    power_ramping_time_us: 0               # Optional FLOAT (0). Sets the power ramping time in microseconds, which proactively initiates transmission and mitigates transient effects.

ru_dummy:
  dl_processing_delay: 1  # Optional UINT (1). Sets the DL processing delay in slots.
  time_scaling: 1.0  # Optional FLOAT (1.0). Sets the time scaling factor applied to the slot duration. Must be greater than zero. A value greater than one slows down the RU, while a value between zero and one speeds it up. Supported: [0 - inf].

hal:
  eal_args:                            # Optional TEXT. Sets the EAL configuration parameters used to initialize DPDK.
  bbdev_hwacc:
    hwacc_type:                        # Optional TEXT. Sets the type of BBDEV hardware accelerator. Supported: [acc100, acc200, vrb1].
    id:                                # Optional UINT. Sets the ID of the BBDEV-based hardware-accelerator. Supported: [0 - 65535].
    pdsch_enc:
      nof_hwacc:                        # Optional UINT. Sets the number of hardware-accelerated PDSCH encoding functions. Supported: [0 - 64].
      cb_mode: false                    # Optional BOOLEAN (false). Sets the operation mode of the PDSCH encoder (CB mode = true, TB mode = false). Supported: [false, true].
      max_buffer_size:                  # Optional UINT. Sets the maximum supported buffer size in bytes (CB mode will be forced for larger TBs).
      dedicated_queue: true             # Optional BOOLEAN (true). Sets whether to use a dedicated hardware queue for the PDSCH encoder. Supported: [false, true].
    pusch_dec:
      nof_hwacc:                        # Optional UINT. Sets the number of hardware-accelerated PUSCH decoding functions. Supported: [0 - 64].
      harq_context_size:                # Optional UINT. Sets the size of the HARQ context repository.
      force_local_harq: false           # Optional BOOLEAN (false). Forces using host memory to implement the HARQ buffer. Supported: [false, true].
      dedicated_queue: true             # Optional BOOLEAN (true). Sets whether to use a dedicated hardware queue for the PUSCH decoder. Supported: [false, true].
    msg_mbuf_size:                      # Optional UINT. Sets the size of the mbufs storing unencoded and unrate-matched messages (in bytes). Supported: [0 - 64000].
    rm_mbuf_size:                       # Optional UINT. Sets the size of the mbufs storing encoded and rate-matched messages (in bytes). Supported: [0 - 64000].
    nof_mbuf:                           # Optional UINT. Sets the number of mbufs in the memory pool.

buffer_pool:
  nof_segments: 1048576                 # Optional UINT (1048576). Sets the number of segments allocated by the buffer pool.
  segment_size: 2048                    # Optional UINT (2048). Sets the size of each buffer pool segment in bytes.

expert_phy:
```

```
  max_proc_delay: 5                              # Optional UINT (5). Sets the maximum allowed DL processing delay in slots. Supported: [1 - 30].
  pusch_dec_max_iterations: 6                    # Optional UINT (6). Sets the maximum number of PUSCH LDPC decoder iterations.
  pusch_dec_enable_early_stop: true              # Optional BOOLEAN (true). Enables the PUSCH LDPC decoder early stop. Supported: [false, true].
  pusch_decoder_force_decoding: false            # Optional BOOLEAN (false). Forces the PUSCH LDPC decoder to decode always. Supported: [false, true].
  pusch_sinr_calc_method: post_equalization      # Optional TEXT (post_equalization). Sets the PUSCH SINR calculation method. Supported: [channel_estimator, post_equalization, evm].
  pusch_channel_estimator_fd_strategy: filter    # Optional TEXT (filter). Sets the PUSCH channel estimator frequency-domain strategy. Supported: [filter, mean, none].
  pusch_channel_estimator_td_strategy: average   # Optional TEXT (average). Sets the PUSCH channel estimator time-domain strategy. Supported: [average, interpolate].
  pusch_channel_estimator_cfo_compensation: true # Optional BOOLEAN (true). Enables the PUSCH channel estimator CFO compensation. Supported: [false, true].
  pusch_channel_equalizer_algorithm: zf          # Optional TEXT (zf). Sets the PUSCH channel equalizer algorithm. Supported: [zf, mmse].
  max_request_headroom_slots: 0                  # Optional UINT (0). Sets the maximum request headroom size in slots. Supported: [0 - 30].
  allow_request_on_empty_uplink_slot: false      # Optional BOOLEAN (false). Generates an uplink request in an uplink slot with no PUCCH/PUSCH/SRS PDUs. Supported: [false, true].
  enable_phy_tap: false                          # Optional BOOLEAN (false). Enables the PHY tap plugin if present. Supported: [false, true].
  phy_tap_arguments:                             # Optional TEXT. Sets the PHY tap plugin argument string passed during construction.

expert_execution:
  affinities:
    main_pool_cpus:                          # Optional TEXT. Sets CPU cores assigned to the main thread pool. No default — the affinity mask is empty by default.
    main_pool_pinning: mask                  # Optional ENUM (mask). Sets the policy used for assigning CPU cores to the main thread pool. Supported: [round-robin, mask].
    ofh:
      timing_cpu:                            # Optional TEXT. Sets the CPU used for timing in the Radio Unit. Format: comma-separated CPU ids or ranges, e.g. "0-3,5".
  threads:
    main_pool:
      nof_threads:                           # Optional UINT. Sets the number of threads for processing upper PHY and upper layers. No default — field is std::optional<unsigned> (nullopt when not set).
      task_queue_size: 2048                  # Optional UINT (2048). Sets the main thread pool task queue size.
      backoff_period: 50                     # Optional UINT (50). Sets the main thread pool back-off period in microseconds.
    upper_phy:
      pdsch_processor_type: auto             # Optional TEXT (auto). Sets the PDSCH processor type. Supported: [auto, generic, flexible].
      pdsch_cb_batch_length: 4               # Optional UINT (4). Sets the PDSCH flexible processor codeblock-batch size. Format: a non-negative integer or the sentinel strings "auto", "default", "synchronous".
      max_pucch_concurrency: 0               # Optional UINT (0). Sets the maximum PUCCH processing concurrency for all cells. 0 means no limit.
```

```
      max_pusch_and_srs_concurrency: 4          # Optional UINT (4). Sets the maximum PUSCH and SRS processing concurrency for all cells. 0 means no limit. If hardware acceleration is enabled, this is set to the number of accelerator queues.
      max_pdsch_concurrency: 0                  # Optional UINT (0). Sets the maximum PDSCH processing concurrency for all cells. 0 means no limit. If hardware acceleration is enabled, this is set to the number of accelerator queues.
    ofh:
      enable_busy_waiting: false                # Optional BOOLEAN (false). Enables busy-waiting of the RU timing worker. Supported: [false, true].
    lower_phy:
      execution_profile: triple                 # Optional TEXT (triple). Sets the lower physical layer executor profile. Supported: [blocking, single, dual, triple].
  queues:
    cu_up_dl_ue_executor_queue_size: 8192     # Optional UINT (8192). Sets the CU-UP DL UE executor queue size.
    cu_up_ul_ue_executor_queue_size: 8192     # Optional UINT (8192). Sets the CU-UP UL UE executor queue size.
    cu_up_ctrl_ue_executor_queue_size: 8192   # Optional UINT (8192). Sets the CU-UP control UE executor queue size.
    cu_up_strand_batch_size: 256              # Optional UINT (256). Sets the CU-UP strand batch size.
    du_ue_data_executor_queue_size: 8192      # Optional UINT (8192). Sets the DU UE executor task queue size for PDU processing.
  cell_affinities:                            # Optional TEXT. Sets the cell CPU affinity configuration on a per cell basis. Each list entry should begin with "-".
    - ru_cpus:                                # Optional TEXT. Sets the CPU cores used for Radio Unit tasks. Format: comma-separated CPU ids or ranges, e.g. "0-3,5".
      ru_pinning: mask                        # Optional TEXT (mask). Sets the policy used for assigning CPU cores to Radio Unit tasks. Format: one of: mask, round-robin.

remote_control:
  enabled: false                          # Optional BOOLEAN (false). Enables the remote control feature. Supported: [false, true].
  bind_addr: 127.0.0.1                    # Optional TEXT (127.0.0.1). Sets the local IP address to bind for remote control. Format: IPV4 or IPV6 IP address.
  port: 8001                              # Optional UINT (8001). Sets the port where the remote control server listens. Supported: [0 - 65535].

test_mode:
  test_ue:
    rnti: 0                               # Optional UINT (0). Sets the C-RNTI of the test UE. Supported: [0 - 65519].
    nof_ues: 1                            # Optional UINT (1). Sets the number of test UEs created. Supported: [0 - 1024].
    ue_creation_stagger_slots: 10         # Optional UINT (10). Sets the number of slots between consecutive test mode UE creations. Supported: [0 - 10240].
    auto_ack_indication_delay:            # Optional TEXT. Sets the delay before UL and DL HARQs are automatically ACKed. Only use if the UL PHY is not operational.
    pdsch_active: true                    # Optional BOOLEAN (true). Enables the PDSCH of the test UE. Supported: [false, true].
    pusch_active: true                    # Optional BOOLEAN (true). Enables the PUSCH of the test UE. Supported: [false, true].
    cqi: 15                               # Optional UINT (15). Sets the Channel Quality Indicator for the test UE. Supported: [1 - 15].
```

```
    pmi: 0                                  # Optional UINT (0). Sets the Precoder Matrix Indicator for the test UE. Supported: [0 - 3].
    ri: 1                                   # Optional UINT (1). Sets the Rank Indicator for the test UE. Supported: [1 - 4].
    i_1_1: 0                                # Optional INT (0). Sets the Precoder Matrix codebook index i_1_1 for the test UE. Supported: [0 - 7].
    i_1_3: 0                                # Optional INT (0). Sets the Precoder Matrix codebook index i_1_3 for the test UE. Supported: [0 - 1].
    i_2: 0                                  # Optional INT (0). Sets the Precoder Matrix codebook index i_2 for the test UE. Supported: [0 - 3].
    attach_detach:
      period_ms: 0                          # Optional UINT (0). Sets the cyclic attach/detach period in ms. 0 disables.
      sleep_ms: 0                           # Optional UINT (0). Sets the sleep time in ms between attach/detach cycles.
    attach_detach_duration_ms:              # Optional UINT. Sets the duration in ms of active traffic before UEs are released and recreated. Supported: [100 - 10000].
    attach_detach_guard_duration_ms: 1000 # Optional UINT (1000). Sets the guard period in ms between a release and next creation cycle. Supported: [100 - 60000].

log:
# All gNB layers and components can be configured independently to output at various levels of detail. Supported levels (from lowest to highest detail):
# none, error, warning, info, debug
  filename: /tmp/gnb.log                    # Optional TEXT (/tmp/gnb.log). Sets the file path for logs. The gnb_appconfig constructor overrides the base logger default (stdout) to /tmp/gnb.log.
  all_level: warning                        # Optional TEXT (warning). Sets a common log level across PHY, MAC, RLC, PDCP, RRC, SDAP, NGAP and GTPU layers. Supported: [none, warning, info, debug, error].
  lib_level: warning                        # Optional TEXT (warning). Sets the log level for external library components. Supported: [none, warning, info, debug, error].
  config_level: none                        # Optional TEXT (none). Sets the configuration parser log level. Supported: [none, info, debug] (error and warning are not accepted).
  ngap_level: warning                       # Optional TEXT (warning). Sets the NGAP log level.
  nrppa_level: warning                      # Optional TEXT (warning). Sets the NRPPa log level.
  e1ap_level: warning                       # Optional TEXT (warning). Sets the E1AP log level.
  e1ap_json_enabled: false                  # Optional BOOLEAN (false). Enables JSON-formatted logging of E1AP PDUs. Supported: [false, true].
  xnap_level: warning                       # Optional TEXT (warning). Sets the Xn-AP log level.
  f1ap_level: warning                       # Optional TEXT (warning). Sets the F1AP log level.
  f1ap_json_enabled: false                  # Optional BOOLEAN (false). Enables JSON-formatted logging of F1AP PDUs. Supported: [false, true].
  e2ap_level: warning                       # Optional TEXT (warning). Sets the E2AP log level. Supported: [none, warning, info, debug, error].
  rrc_level: warning                        # Optional TEXT (warning). Sets the RRC log level.
  sec_level: warning                        # Optional TEXT (warning). Sets the security functions log level.
  cu_level: warning                         # Optional TEXT (warning). Sets the CU management log level.
```

```
  pdcp_level: warning                    # Optional TEXT (warning). Sets the PDCP log level.
  sdap_level: warning                    # Optional TEXT (warning). Sets the SDAP log level.
  rohc_level: warning                    # Optional TEXT (warning). Sets the ROHC log level. Supported: [none, error, warning, info, debug].
  du_level: warning                      # Optional TEXT (warning). Sets the DU management log level.
  mac_level: warning                     # Optional TEXT (warning). Sets the MAC log level.
  rlc_level: warning                     # Optional TEXT (warning). Sets the RLC log level.
  f1u_level: warning                     # Optional TEXT (warning). Sets the F1-U log level.
  gtpu_level: warning                    # Optional TEXT (warning). Sets the GTP-U log level.
  fapi_level: warning                    # Optional TEXT (warning). Sets the FAPI log level (split_7_2 and split_6 only).
  phy_level: warning                     # Optional TEXT (warning). Sets the physical layer log level.
  ofh_level: warning                     # Optional TEXT (warning). Sets the Open Fronthaul log level.
  radio_level: info                      # Optional TEXT (info). Sets the radio driver log level.
  hal_level: warning                     # Optional TEXT (warning). Sets the HAL log level.
  ntn_level: warning                     # Optional TEXT (warning). Sets the NTN log level.
  hex_max_size: 0                        # Optional INT (0). Sets the maximum number of bytes to print in hex. 0 disables hex dumps; -1 for unlimited. Supported: [-1 - 1024].
  broadcast_enabled: false               # Optional BOOLEAN (false). Enables logging of broadcast messages and all PRACH opportunities in the PHY and MAC layers. Supported: [false, true].
  high_latency_diagnostics_enabled: false  # Optional BOOLEAN (false). Enables logging of performance diagnostics when high computational latencies are detected. Supported: [false, true].
  phy_rx_symbols_filename:               # Optional TEXT. Sets the file path to dump received IQ symbols. Leave empty to disable.
  phy_rx_symbols_port: 0                 # Optional TEXT (0). Sets the Rx port number to dump IQ symbols from, or "all" for all UL receive ports. Only active if phy_rx_symbols_filename is set.
  phy_rx_symbols_prach: false            # Optional BOOLEAN (false). Enables dumping of IQ symbols from all PRACH ports. Only active if phy_rx_symbols_filename is set. Supported: [false, true].

pcap:
  ngap_filename: /tmp/cu_ngap.pcap           # Optional TEXT (/tmp/cu_ngap.pcap). Sets the path for NGAP PCAP files.
  ngap_enable: false                         # Optional BOOLEAN (false). Enables NGAP packet capture. Supported: [false, true].
  n3_filename:                               # Optional TEXT. Sets the N3 GTP-U PCAP file output path.
  n3_enable: false                           # Optional BOOLEAN (false). Enables N3 packet capture. Supported: [false, true].
  e1ap_filename: /tmp/cu_e1ap.pcap           # Optional TEXT (/tmp/cu_e1ap.pcap). Sets the E1AP PCAP file output path.
  e1ap_enable: false                         # Optional BOOLEAN (false). Enables E1AP packet capture. Supported: [false, true].
```

```
  e2ap_cu_cp_filename: /tmp/cu_cp_e2ap.pcap   # Optional TEXT (/tmp/cu_cp_e2ap.pcap). Sets the path for E2AP CU-CP PCAP files.
  e2ap_cu_up_filename: /tmp/cu_up_e2ap.pcap   # Optional TEXT (/tmp/cu_up_e2ap.pcap). Sets the path for E2AP CU-UP PCAP files.
  e2ap_du_filename: /tmp/du_e2ap.pcap         # Optional TEXT (/tmp/du_e2ap.pcap). Sets the path for E2AP DU PCAP files.
  e2ap_enable: false                          # Optional BOOLEAN (false). Enables E2AP packet capture. Supported: [false, true].
  f1ap_filename: /tmp/cu_f1ap.pcap            # Optional TEXT (/tmp/cu_f1ap.pcap). Sets the path for F1AP PCAP files.
  f1ap_enable: false                          # Optional BOOLEAN (false). Enables F1AP packet capture. Supported: [false, true].
  f1u_filename: /tmp/du_f1u.pcap              # Optional TEXT (/tmp/du_f1u.pcap). Sets the F1-U GTP-U PCAP file output path.
  f1u_enable: false                           # Optional BOOLEAN (false). Enables F1-U packet capture. Supported: [false, true].
  rlc_filename: /tmp/du_rlc.pcap              # Optional TEXT (/tmp/du_rlc.pcap). Sets the path for RLC PCAP files.
  rlc_rb_type: all                            # Optional TEXT (all). Sets the RLC PCAP RB type filter. Supported: [all, srb, drb].
  rlc_enable: false                           # Optional BOOLEAN (false). Enables RLC packet capture. Supported: [false, true].
  mac_filename: /tmp/du_mac.pcap              # Optional TEXT (/tmp/du_mac.pcap). Sets the path for MAC PCAP files.
  mac_type: udp                               # Optional TEXT (udp). Sets the MAC PCAP format. Supported: [dlt, udp].
  mac_enable: false                           # Optional BOOLEAN (false). Enables MAC packet capture. Supported: [false, true].
  xnap_filename: /tmp/cu_xnap.pcap            # Optional TEXT (/tmp/cu_xnap.pcap). Sets the path for Xn-AP PCAP files.
  xnap_enable: false                          # Optional BOOLEAN (false). Enables Xn-AP packet capture. Supported: [false, true].

metrics:
  autostart_stdout_metrics: false       # Optional BOOLEAN (false). Autostart stdout metrics reporting. Supported: [false, true].
  enable_json: false                    # Optional BOOLEAN (false). Enables JSON metrics reporting. Supported: [false, true].
  enable_log: false                     # Optional BOOLEAN (false). Enables log metrics reporting. Supported: [false, true].
  enable_verbose: false                 # Optional BOOLEAN (false). Enables extended detail metrics reporting. Supported: [false, true].
  layers:
    enable_app_usage: false             # Optional BOOLEAN (false). Enable application usage metrics. Supported: [false, true].
    enable_ngap: false                  # Optional BOOLEAN (false). Enables NGAP metrics. Supported: [false, true].
    enable_e1ap: false                  # Optional BOOLEAN (false). Enables E1AP metrics. Supported: [false, true].
    enable_pdcp: false                  # Optional BOOLEAN (false). Enables PDCP metrics (reported by the CU-UP). Supported: [false, true].
    enable_nrup_cu: false               # Optional BOOLEAN (false). Enables NRUP metrics (CU side). Supported: [false, true].
    enable_rrc: false                   # Optional BOOLEAN (false). Enable CU-CP RRC metrics. Supported: [false, true].
    skip_cu_up_executor: true           # Optional BOOLEAN (true). Whether to skip logging CU-UP executor metrics when executor logging is enabled application wide. Supported: [false, true].
    enable_sched: true                  # Optional BOOLEAN (true). Enables DU scheduler metrics. Supported: [false, true].
```

```
    enable_rlc: false                   # Optional BOOLEAN (false). Enables RLC metrics. Supported: [false, true].
    enable_mac: false                   # Optional BOOLEAN (false). Enables MAC metrics. Supported: [false, true].
    enable_du_proc: false               # Optional BOOLEAN (false). Enables DU management and control procedure metrics. Supported: [false, true].
    enable_executor: false              # Optional BOOLEAN (false). Whether to log application executors metrics. Supported: [false, true].
    enable_du_low: false                # Optional BOOLEAN (false). Enables DU low (upper physical layer) metrics. Supported: [false, true].
    enable_ru: false                    # Optional BOOLEAN (false). Enables Radio Unit metrics. Supported: [false, true].
  periodicity:
    app_usage_report_period: 1000       # Optional UINT (1000). Sets the application resource usage metrics report period in milliseconds.
    cu_cp_report_period: 1000           # Optional UINT (1000). Sets the CU-CP metrics report period in milliseconds.
    cu_up_report_period: 1000           # Optional UINT (1000). Sets the CU-UP metrics report period in milliseconds.
    du_report_period: 1000              # Optional UINT (1000). Sets the DU statistics report period in milliseconds. Supported: [0 - 10485760].
    executors_report_period: 1000       # Optional UINT (1000). Sets the executor metrics report period in milliseconds. Supported: [0 - 10485760].

trace:
  filename:                             # Optional TEXT. Sets the file path for trace output. Leave empty to disable tracing.
  max_tracing_events_per_file: 1000000  # Optional UINT (1000000). Sets the maximum number of events per trace file. 0 means no limit.
  nof_tracing_events_after_severe: 0    # Optional UINT (0). Sets the number of events to write prior to a severe event. Set to zero for writing all events.
  layers:
    cu_up_enable: false                 # Optional BOOLEAN (false). Enables tracing for CU-UP executors. Supported: [false, true].
    du_high_enable: false               # Optional BOOLEAN (false). Enables tracing for DU-high executors. Supported: [false, true].
    phy_enable: false                   # Optional BOOLEAN (false). Enables tracing for physical layer executors. Supported: [false, true].
```

Antenna Configuration

OCUDU supports SISO, codebook-based MIMO, and TX/RX diversity. These options can be configured in the `cell_cfg` section via the `nof_antennas_dl` and `nof_antennas_ul` parameters.

When you configure `nof_antennas_dl` and `nof_antennas_ul`, it informs the gNB about the number of antennas available for downlink (DL) and uplink (UL) transmissions, respectively. By default, a single physical antenna is used for both DL and UL, meaning the values of `nof_antennas_dl` and `nof_antennas_ul` are not additive.

As shown above, an example of a MIMO configuration file can be found in the example configuration files provided with OCUDU source files.

Scheduler Log Reference

The MAC scheduler writes to the `SCHED` logger. Two independent loggers produce its output, both defined in `logging/`:

- **Result logger** (`scheduler_result_logger`) — one `Slot decisions` line per scheduled slot, describing the grants written into the `sched_result` (PDCCH, PDSCH, PUSCH, PUCCH, SRS, ...).
- **Event logger** (`scheduler_event_logger`) — one `Processed slot events` line per slot in which UE/cell events were processed (CRC, HARQ-ACK, BSR, UE creation, PRACH, ...).

Both are per-cell and tag every line with `pci=<physical cell id>`.

Verbosity

The output is controlled by the level of the `SCHED` logger:

Level	Result logger	Event logger
`info`	Summary line + compact one-line entries (`log_info`)	Only events with an info formatter: PRACH, RACH indication, ErrorIndication, cell creation
`debug`	Summary line + one detailed entry per line (`log_debug`)	All events, one detailed entry per line
below `info`	disabled	disabled

`debug` is a superset of `info`: it logs more entry types and more fields per entry.

`log_broadcast`

The result logger is constructed with a `log_broadcast` flag. When `false`, broadcast/common allocations (SSB, SIB, CSI-RS, PRACH occasions, and `SI-RNTI` PDCCH) are omitted. This is normally enabled for a single cell only, so recurring broadcast allocations do not flood the log across all cells.

Log line anatomy

```
2026-07-02T09:29:21.835770 [SCHED   ] [D] [     1.5] Slot decisions pci=0 t=0us (1 PDSCH,
...):
└─ timestamp └─ logger | | └─ message
```

```
|  └─ slot as <SFN>.<slot-in-frame>
└─ level: [I]=info, [D]=debug
```

The `[  SFN.slot]` field is the transmit slot the decisions apply to. Event lines carry the slot in which the events were processed.

Result logger — `Slot decisions`

Summary line

A slot is logged only if it carries something worth reporting: a UE DL/UL grant, RAR, paging, a non-`SI-RNTI` PDCCH, (with `log_broadcast`) a broadcast allocation, or any failed allocation attempt. Empty slots are skipped.

info:

```
Slot decisions pci=0 t=0us (1 PDSCH, 1 PUSCH, 1 PUCCH): <entries...>
```

debug (adds attempted-but-failed allocation counters):

```
Slot decisions pci=1 t=0us (1 PDSCH, 0 PUSCHs, 0 PUCCHs, 0 attempted PDCCHs, 0 attempted
UCIs): <entries...>
```

Field	Meaning
`t=<n>us`	Scheduler decision latency for this slot, microseconds
`N PDSCH`	Number of PDSCH grants (UE grants + RAR + paging, + SIBs when `log_broadcast`)
`N PUSCH`	Number of PUSCH grants
`N PUCCH`	Number of PUCCH grants
`N attempted PDCCH`	Failed DL+UL PDCCH allocation attempts (`debug` only)
`N attempted UCI`	Failed UCI allocation attempts (`debug` only)

At **info** level the entries follow the summary on the *same line*, comma-separated. At **debug** level each entry is on its own line prefixed with `-`.

info-level entries (compact, comma-separated)

Prefix	Format
`SIB1` / `SI-<n>`	`SIB1: rb=<prbs> tbs=<bytes>`
`RAR`	`RAR: ra-rnti=<rnti> rb=<prbs> tbs=<bytes>`
`DL`	`DL: ue=<idx> c-rnti=<rnti> h_id=<harq> ss_id=<ss> rb=<prbs> k1=<k1> newtx=<0/1> rv=<rv> tbs=<bytes>` — on new transmissions appends `ri=<layers> dl_bo=<buffer-occupancy>`; when part of a Rel-16 PDSCH repetition bundle appends `reps=<n> reps_remaining=<n>`
`UL`	`UL: ue=<idx> rnti=<rnti> h_id=<harq> ss_id=<ss> rb=<prbs> newtx=<0/1> rv=<rv> tbs=<bytes>` — then `k2=<k2>`, or `msg3_delay=<n>` for a Msg3 grant (invalid UE index, first tx)
`PG`	`PG: rb=<prbs> tbs=<bytes> ues: <cn|ran>-pg-id=<0x..>, ...`

debug-level entries (one per line, `-` prefix)

Control channels:

```
- DL PDCCH: rnti=0x4601 type=c-rnti cs_id=1 ss_id=2 format=1_1 cce=8 al=2 dci: h_id=0 ndi=1 rv=0 mcs=3 res_ind=0 tdra_idx=3 tpc=1 dai=0
- UL PDCCH: rnti=0x4601 type=c-rnti cs_id=1 ss_id=2 format=0_1 cce=10 al=2 dci: h_id=0 ndi=1 rv=0 mcs=9 tpc=1 dai=0 mimo=0 ant=2
```

Field	Meaning
`type`	RNTI type (`c-rnti`, `tc-rnti`, `si-rnti`, ...)
`cs_id` / `ss_id`	CORESET id / SearchSpace id
`format`	DCI format (`1_0`, `1_1`, `0_0`, `0_1`)
`cce` / `al`	Starting CCE index / aggregation level (number of CCEs)
`dci: ...`	Decoded DCI fields — `h_id` HARQ process, `ndi`, `rv`, `mcs`. DL adds `res_ind` (PUCCH resource indicator), `tdra_idx` (PDSCH time-domain-resource-allocation row index; omitted for paging DCIs short message) and optional `tpc` / `dai` / `vrb_prb_map_used`. UL adds `tpc` and, for format 0_1, `dai` / `mimo` (layers) / `ant` / `csi_req`

DL shared channel:

```
- SSB: ssbIdx=0 crbs=[5..26) symb=[2..6)
- CSI-RS: type=nzp crbs=[0..52) row=1 freq=0010 symb0=4 cdm_type=no_CDM freq_density=three
scramb_id=1
- SIB1 PDSCH: rb=[0..4) symb=[2..14) tbs=120 mcs=5 rv=0
- RAR PDSCH: ra-rnti=0x2 rb=[0..8) symb=[1..14) tbs=32 mcs=0 rv=0 grants (1): tc-
rnti=0x4601: rapid=5 ta=12 time_res=0
- UE PDSCH: ue=0 c-rnti=0x4601 h_id=0 rb=[0..25) symb=[1..14) tbs=309 mcs=9 rv=0 nrtx=0 k1=4
ri=1 <pmi> dl_bo=1024 olla=0.03 grants: lcid=1: size=53, lcid=4: size=256
- PCCH: rb=[0..4) symb=[2..14) tbs=40 mcs=5 rv=0 ues: cn-pg-id=0x1234
```

- `UE PDSCH`: `nrtx` = retransmission count; `ri`/PMI printed only when precoding present; `dl_bo` (DL buffer occupancy) only on new data; `olla` (outer-loop link-adaptation offset) when set; `reps`/`reps_remaining` only when the grant is part of a Rel-16 PDSCH repetition bundle (`reps` = nominal bundle size, `reps_remaining` = repetitions still to be transmitted after this occasion, decrementing to 0 at the last one); per-logical-channel `grants:` list when a TB is built. RAR `result=successRAR|fallbackRAR` appears for 2-step grants.

UL shared channel and control:

```
- UE PUSCH: ue=0 c-rnti=0x4601 h_id=0 rb=[1..52) symb=[0..14) tbs=1121 rv=0 nrtx=0
nof_layers=1 olla=0 k2=4 uci: harq_bits=1 csi-1_bits=0 csi-2_present=No
- PUCCH: c-rnti=0x4601 format=1 prb=[0..1) prb2=50 symb=[0..14) cs=0 occ=0 uci: harq_bits=1
sr=0
- SRS: c-rnti=0x4601 symb=[8..10) tx-comb=(n2 o=0 cs=0) c_srs=0 f_sh=0 seq_id=0 requests=
[ch_mtx=yes pos=no]
- PRACH: pci=1 format=0 nof_occasions=1 nof_preambles=64
```

- `UE PUSCH`: a leading `t` before `c-rnti` marks a temporary-C-RNTI (Msg3) grant; `k2` is replaced by `msg3_delay` for the initial Msg3. `uci:` block present when UCI is multiplexed on PUSCH.
- `PUCCH`: `prb2` shown when frequency hopping; format-specific fields — `cs`/`occ` (F1), `occ=<idx>/<len>` (F4); `csi-1_bits` shown for F2/F3/F4.

Interval notation

`rb`, `symb`, `crbs` are printed as half-open intervals `[start..stop)` — e.g. `rb=[1..52)` is 51 PRBs starting at PRB 1; `symb=[0..14)` is all 14 OFDM symbols.

Event logger — `Processed slot events`

Emitted once per slot in which one or more UE/cell events were processed.

info (space-separated, only events with an info formatter):

```
Processed slot events pci=1: prach(ra-rnti=0x2 preamble=5 tc-rnti=0x4601), RACH Ind
slot_rx=100.3, ErrorIndication slot=100.5
```

debug (one event per line):

```
Processed slot events pci=1:
- UE creation: ue=0 rnti=0x4601
- CRC: ue=0 rnti=0x4601 rx_slot=401.9 h_id=0 crc=true sinr=100dB
- HARQ-ACK: ue=0 rnti=0x4601 slot_rx=401.5 h_id=0 ack=1 tbs=720
- BSR: ue=0 rnti=0x4601 type="Short BSR" report={2: 700} pending_bytes=706
- CSI: ue=0 rnti=0x4601: slot_rx=408.9 cqi=15
- RLC Buffer State: ue=0 lcid=4 pending_bytes=2400
```

Event types

Event	Levels	Notes
Cell creation	info + debug	Emitted once at logger construction
PRACH	info + debug	Detected preamble → RA-RNTI/MsgB-RNTI, `tc-rnti`; debug adds `preamble`, `temp_crnti`, `ta_cmd`
RACH ind	info + debug	debug lists each occasion: `preamble`, `tc-rnti`, `freq_idx`, `start_symbol`, `TA`
ErrorIndication	info + debug	debug lists which channels were discarded (PDCCH / PDSCH / PUSCH+PUCCH)
UE creation / reconfiguration / removal	debug	`ue=<idx> rnti=<rnti>`
UE dedicated config applied / UE deactivation	debug	`ue=<idx> rnti=<rnti>`
SR	debug	Scheduling request
CSI	debug	`cqi` / `ri` / `pmi` when present, else `invalid`
BSR	debug	`type` (Short/Long/...), per-LCG `report={...}`, `pending_bytes`
HARQ-ACK	debug	`ack` status (0/1/2); `tbs` shown on ACK
CRC	debug	`crc=true/false`, `sinr=<n>dB` or `sinr=N/A`
MAC CE	debug	`ue=<idx> lcid=<lcid>`
RLC Buffer State	debug	`ue=<idx> lcid=<lcid> pending_bytes=<n>`
PHR	debug	`ph=<n>dB`, optional `p_cmax=<n>dBm`
SRS	debug	optional `tpmi_info=[...]`
Slice Reconfig	debug	`cell=<idx>`

Common field glossary

Field	Meaning
`pci`	Physical cell id
`ue`	DU UE index (internal); `INVALID` / `t`-prefix denotes a temporary (pre-attach) context
`c-rnti` / `rnti`	UE C-RNTI
`tc-rnti`	Temporary C-RNTI (RA / Msg3 / Msg4)
`ra-rnti` / `msgb-rnti`	RA-RNTI (4-step) / MsgB-RNTI (2-step) derived from the PRACH occasion
`h_id`	HARQ process id
`ndi` / `rv`	New-data indicator / redundancy version
`mcs`	Modulation and coding scheme index
`tdra_idx`	PDSCH/PUSCH time-domain-resource-allocation row index signalled in the DCI
`tbs`	Transport block size, bytes
`rb`	Allocated PRBs, half-open interval
`symb`	Allocated OFDM symbols, half-open interval
`k1`	PDSCH-to-HARQ-ACK slot offset
`k2`	PDCCH/DCI-to-PUSCH slot offset
`nrtx` / `nof_retxs`	HARQ retransmission count
`reps`	Nominal number of occasions in a Rel-16 PDSCH repetition bundle (1 = no repetition)
`reps_remaining`	Occasions of the bundle still to be transmitted after this one, counting only committed occasions
`dl_bo`	DL buffer occupancy (pending bytes)
`olla`	Outer-loop link-adaptation MCS offset
`ss_id` / `cs_id`	SearchSpace id / CORESET id

Outputs

Console Reference

OCUDU can output metrics to command line while running, to enable this, type `t` in the console.

A sample output from the gNB/DU showing bi-directional traffic can be seen here:

```
|--------------------DL---------------------|-------------------------UL-------------------
-------------
pci rnti | cqi  ri  mcs  brate   ok  nok  (%)  dl_bs | pusch  rsrp  ri  mcs  brate   ok  nok
(%)    bsr    ta  phr
1 4601 |  15   1   28    17M  908    0   0%    44k |  26.7 -21.2   1   28    17M  545    0
0%  39.8k   0us   18
1 4601 |  15   1   28    18M  966    0   0%      0 |  26.8 -21.3   1   28    17M  547    0
0%      0   0us   18
1 4601 |  15   1   28    17M  912    0   0%      0 |  26.8 -21.2   1   28    17M  558    0
0%  77.3k   n/a   18
1 4601 |  15   1   28    17M  913    0   0%   116k |  26.8 -21.2   1   28    17M  542    0
0%   108k   0us   18
1 4601 |  15   1   28    18M  949    0   0%  28.5k |  26.8 -21.2   1   28    17M  559    0
0%      0   n/a   18
1 4601 |  15   1   28    18M  936    0   0%      0 |  26.7 -21.2   1   28    17M  549    0
0%   108k   0us   18
1 4601 |  15   1   28    17M  918    0   0%      0 |  26.6 -21.2   1   28    17M  552    0
0%   108k   n/a   18
1 4601 |  15   1   28    17M  913    0   0%   110k |  26.6 -21.2   1   28    17M  552    0
0%  28.6k   n/a   18
1 4601 |  15   1   28    18M  904    0   0%  17.9k |  26.7 -21.2   1   28    17M  547    0
0%   150k   0us   18
1 4601 |  15   1   28    18M  909    0   0%      0 |  26.5 -21.2   1   28    17M  548    0
0%   150k   n/a   18
1 4601 |  15   1   28    17M  903    0   0%      0 |  26.5 -21.2   1   28    17M  552    0
0%  55.5k   n/a   18
```

Metrics are provided on a per-UE basis. The following metrics are provided:

- **pci:** Physical Cell Identifier
- **rnti:** Radio Network Temporary Identifier (UE identifier)
- **cqi:** Channel Quality Indicator reported by the UE (1-15)
- **ri:** Rank Indicator as reported by the UE
- **mcs:** Modulation and coding scheme (0-28)
- **brate:** Bitrate (bits/sec)
- **ok:** Number of packets successfully sent
- **nok:** Number of packets dropped
- **(%):** % of packets dropped

- **dl_bs:** Downlink Buffer Status, data waiting to be transmitted as reported by the gNB (bytes)
- **pusch:** PUSCH SINR (Signal-to-Interference-plus-Noise Ratio)
- **ri:** Rank Indicator as reported by the gNB
- **rsrp:** Reference Signal Received Power
- **bsr:** Buffer Status Report, data waiting to be transmitted as reported by the UE (bytes)
- **ta:** Timing Advance in microseconds
- **phr:** Power Headroom as reported by the UE

Logs

OCUDU provides users with a highly configurable logging mechanism, with per-layer and per-component log levels. Set the log file path and log levels in the gNB config file. See the Configuration Reference for more details.

The format used for all log messages is as follows:

```
*Timestamp [Layer] [Level] [TTI] message*
```

Where the fields are:

- Timestamp in *YYYY-MM-DDTHH:MM:SS.UUUUUU* format at which log message was generated
- Layer can be one of *MAC/RLC/PDCP/RRC/SDAP/NGAP/GTPU/RADIO/FAPI/F1U/DU/CU/LIB*. PHY layers are specified as downlink or uplink and with executor number e.g. *DL-PHY1*.
- Level can be one of *E/W/I/D* for error, warning, info and debug respectively.
- TTI is only shown for PHY or MAC messages and is in the format *SFN.sn* where SFN is System Frame Number and sn is slot number.

An example log file excerpt can be seen below:

```
2023-03-15T18:29:25.142200 [MAC     ] [I] [  276.14] UL PDU rnti=0x4601 ue=0 subPDUs:
[lcid=1: len=96, SBSR: lcg=0 bs=0, SE_PHR: total_len=3, PAD: len=424]
2023-03-15T18:29:25.142204 [RLC     ] [I] ue=0 SRB1 UL: RX PDU. pdu_len=96 dc=data p=1
si=full sn=0 so=0
2023-03-15T18:29:25.142226 [PDCP    ] [I] ue=0 SRB1 UL: RX PDU. type=data pdu_len=94 sn=0
count=0
2023-03-15T18:29:25.142228 [PDCP    ] [I] ue=0 SRB1 UL: RX SDU. count=0
2023-03-15T18:29:25.142245 [RRC     ] [D] ue=0 SRB1 - Rx DCCH UL rrcSetupComplete (88 B)
2023-03-15T18:29:25.142249 [RRC     ] [D] Content: [
{
"UL-DCCH-Message": {
"message": {
"c1": {
"rrcSetupComplete": {
"rrc-TransactionIdentifier": 0,
"criticalExtensions": {
"rrcSetupComplete": {
"selectedPLMN-Identity": 1,
```

```
"registeredAMF": {
"amf-Identifier": "000000100000000001000000"
},
"guami-Type": "native",
"dedicatedNAS-Message":
"7e01820be950137e004139000bf200f110020040e7000f272e04f070f0707100307e004139000bf200f11002004
0e7000f27100200402e04f070f0702f0201015200f1100000071801007400090530101"
}
}
}
}
}
}
}
]
2023-03-15T18:29:25.142253 [RRC     ] [D] ue=0 "RRC Setup Procedure" finished successfully
2023-03-15T18:29:25.142263 [NGAP    ] [I] ue=0 Sending InitialUeMessage (ran_ue_id=0)
```

PCAPs

OCUDU can output PCAPs at the following layers:

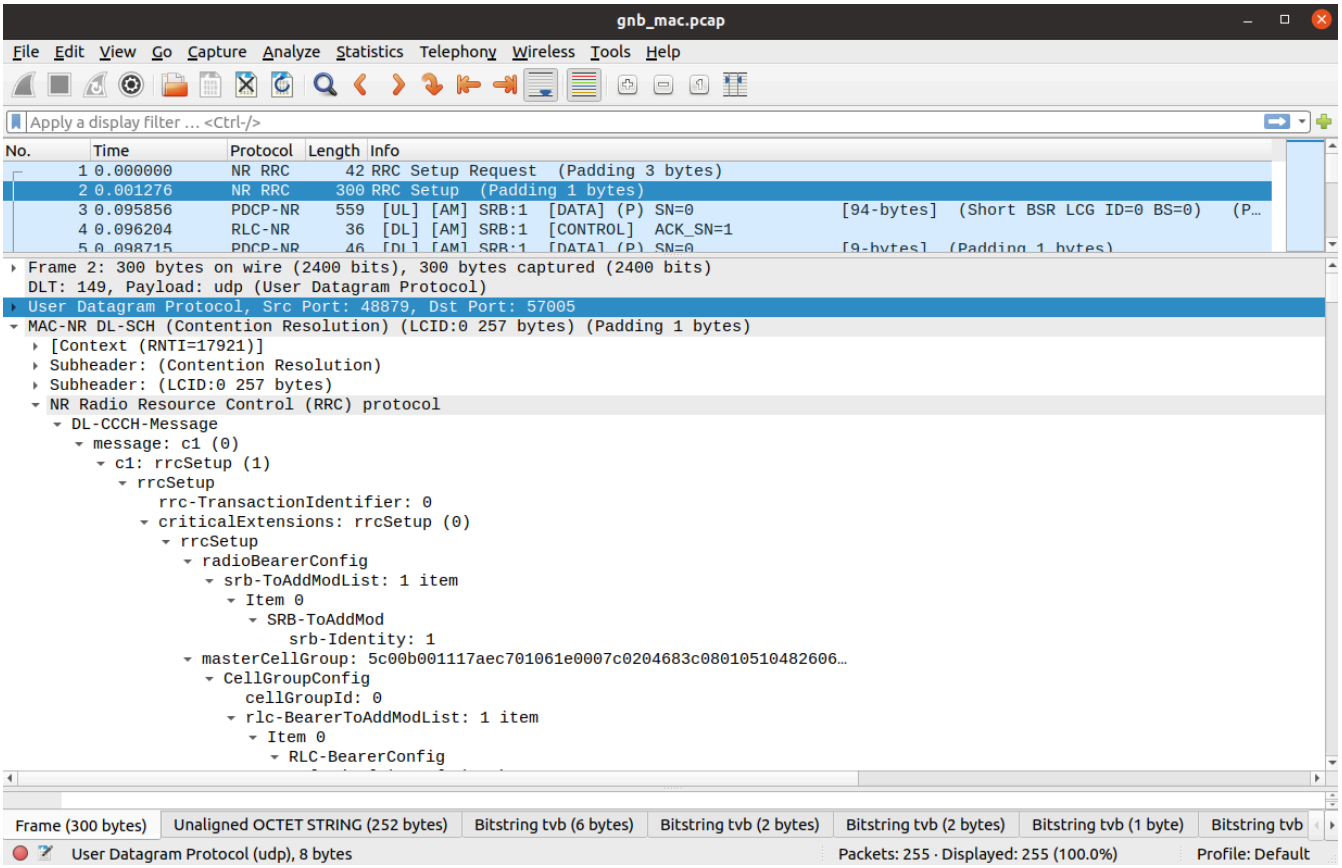
- MAC
- RLC
- NGAP
- N3
- E1AP
- F1AP
- E2AP

To output these PCAPs, they must first be enabled on a per-layer basis in the gNB configuration file. See the Configuration Reference for more details.

MAC

To analyze a MAC-layer PCAP using Wireshark, you will need to configure User DLT 149 for UDP and enable the mac_nr_udp protocol:

1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=149 and Payload protocol=udp.
2. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=157 and Payload protocol=mac-nr-framed.
3. Go to Analyze->Enabled Protocols->MAC-NR and enable mac_nr_udp
4. Go to Edit->Preferences->Protocols->MAC-NR: Enable both checkboxes “Attempt to…”; Set LCID->DRB mapping to “From configuration protocol”.



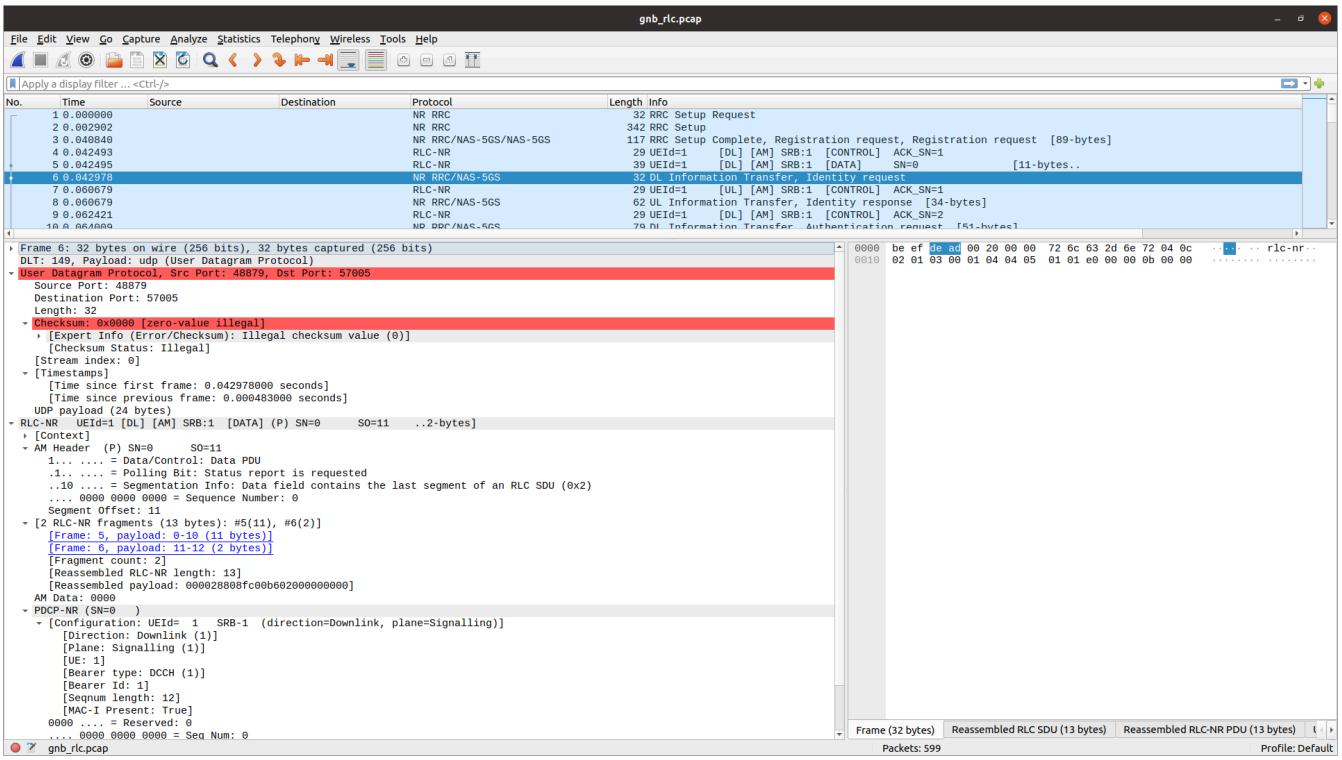
RLC

> **INFO**
>
> To correctly view the RLC PCAPs you will need Wireshark v4.3.x or later.

To analyze a RLC-layer PCAP using Wireshark, you will need to configure User DLT 149 for UDP and enable the rlc_nr_udp protocol:

1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=149 and Payload protocol=udp.
2. Go to Analyze->Enabled Protocols->RLC-NR and enable rlc_nr_udp

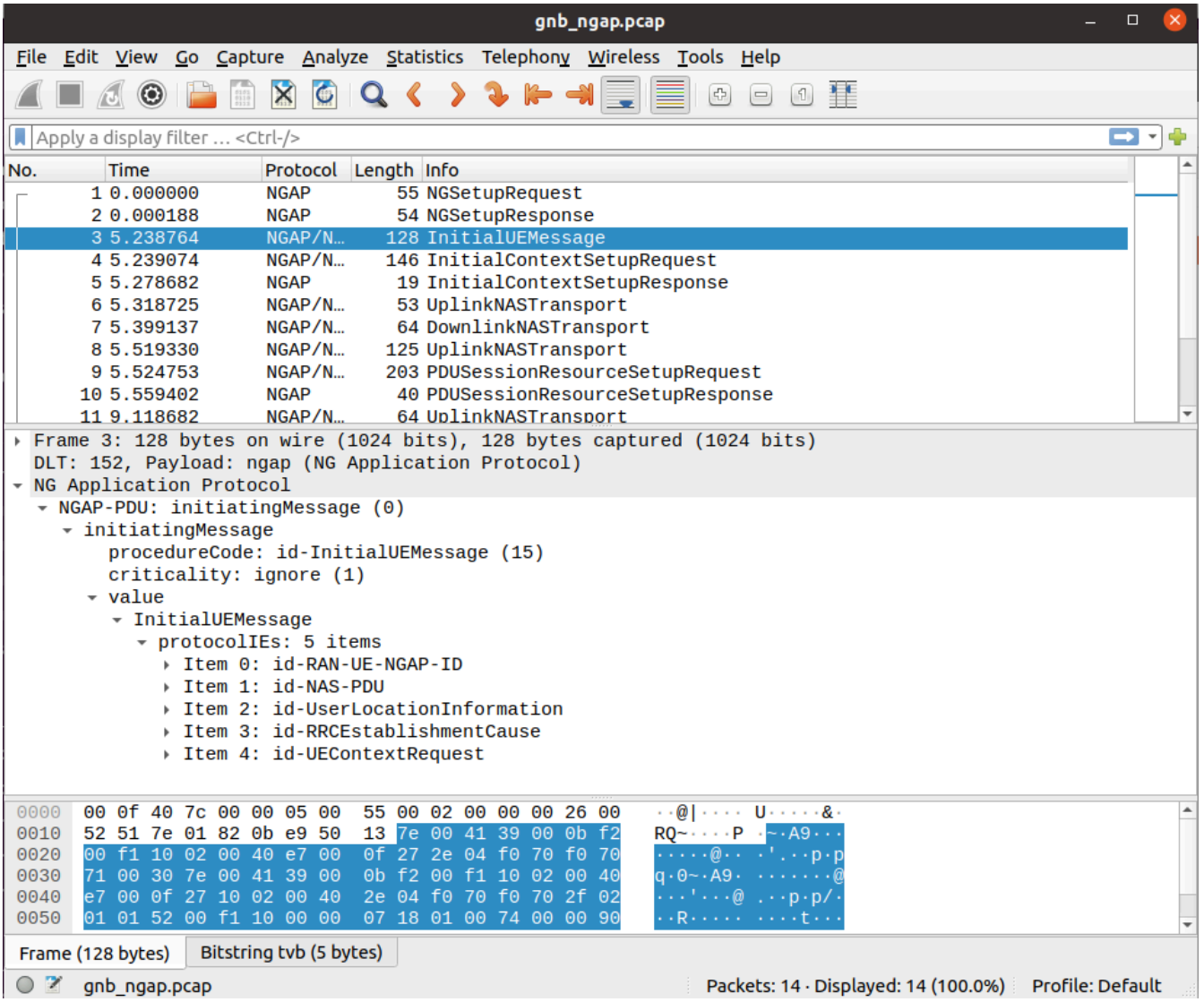
3. Go to Edit->Preferences->Protocols->RLC-NR and configure according to your needs.



NGAP

To analyze an NGAP-layer PCAP using Wireshark, you will need to configure User DLT 152 for NGAP and enable detection and decoding 5G-EA0 ciphered messages:

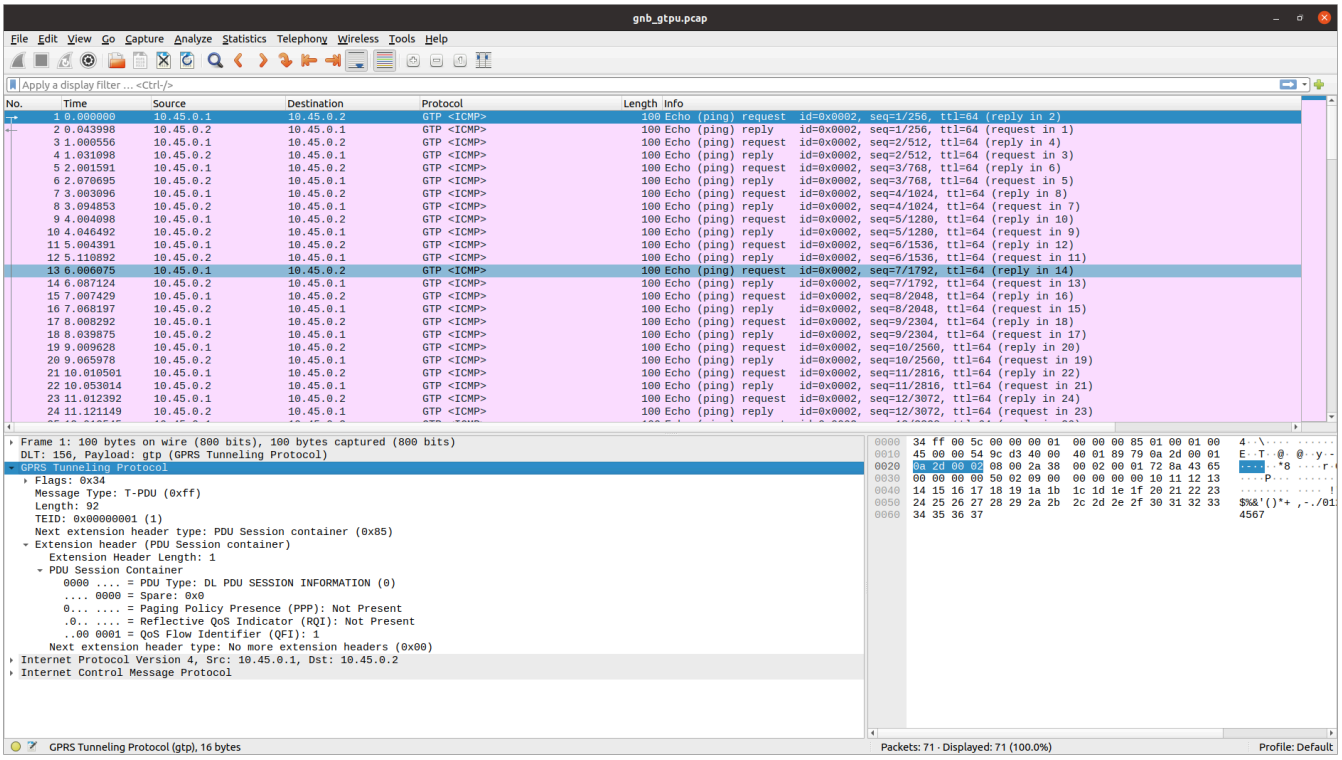
1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=152 and Payload protocol=ngap.
2. Go to Edit->Preferences->Protocols->NAS-5GS and enable “Try to detect and decode 5G-EA0 ciphered messages”.



N3

To analyze a N3 PCAP using Wireshark, you will need to configure User DLT 156 for GTP:

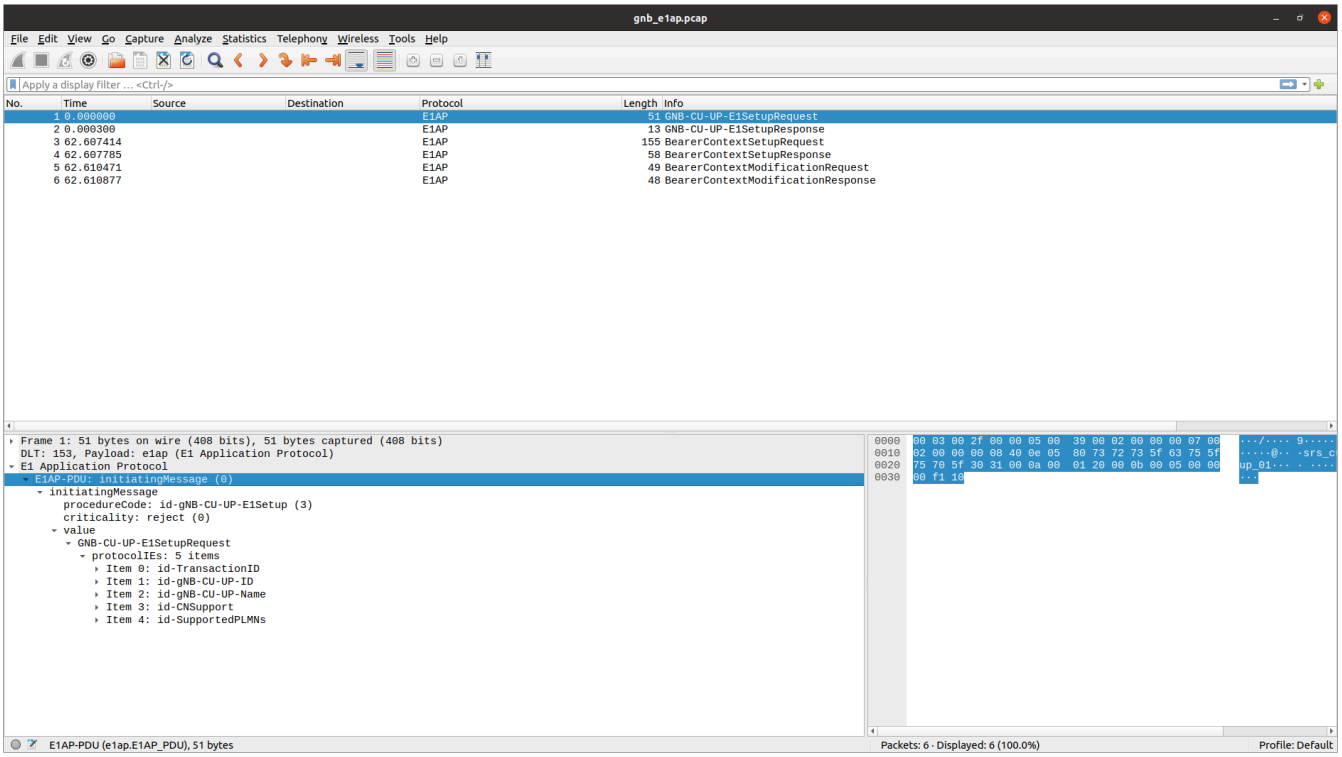
1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=156 and Payload Protocol=gtp.



E1AP

To analyze an E1AP PCAP using Wireshark, you will need to configure User DLT 153 for E1AP:

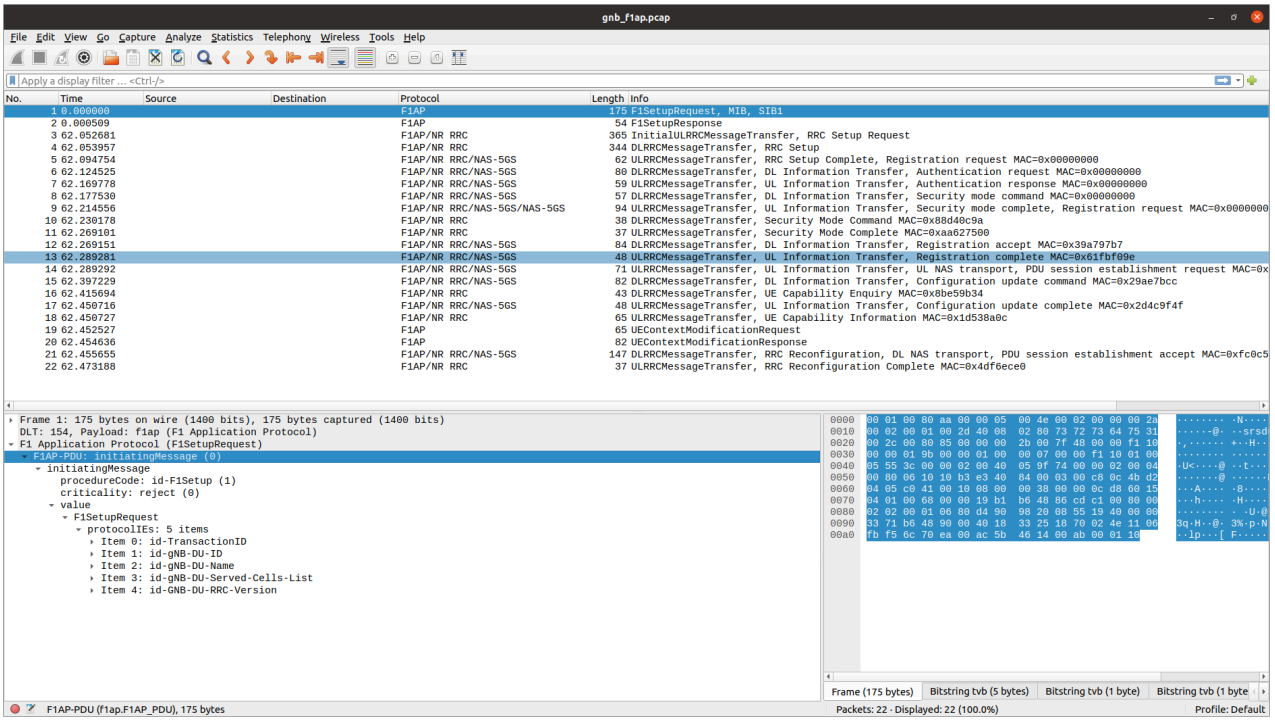
1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=153 and Payload Protocol=e1ap.



F1AP

To analyze an F1AP PCAP using Wireshark, you will need to configure User DLT 154 for F1AP:

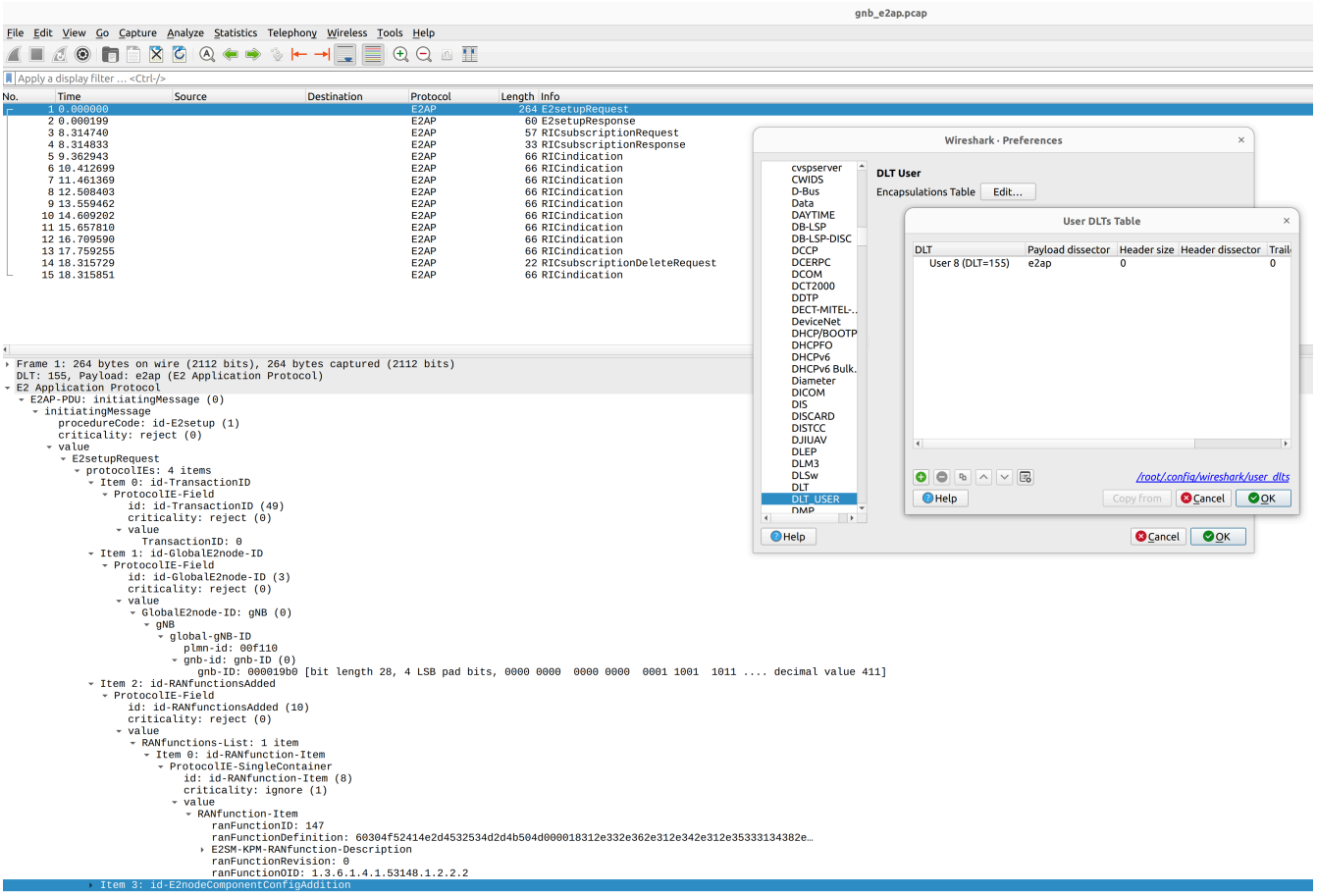
1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=154 and Payload Protocol=f1ap.



E2AP

To analyze an E2AP PCAP using Wireshark, you will need to configure User DLT 155 for E2AP:

1. Go to Edit->Preferences->Protocols->DLT_USER->Edit and add an entry with DLT=155 and Payload Protocol=e2ap.



JSON Metrics

OCUDU supports the reporting of the console metrics to JSON data format over websocket. This is used to generate the output seen in the Grafana GUI.

The metrics can be received and printed to `stdout` using a Python script.

Firstly, the metrics output and remote control can be enabled by adding the following to your configuration file:

```
metrics:
enable_json: true
remote_control:
bind_addr: 0.0.0.0
port: 8001
enabled: true
```

With this enabled, the websocket will listen on `127.0.0.1:8001`. The metrics can be subscribed and written to `stdout` in real-time while the gNB is running using the following Python code:

```
#!/usr/bin/env python3

from contextlib import suppress
import os
import json
from time import sleep
import websocket

def _on_open(ws: websocket.WebSocketApp):
ws.send(json.dumps({"cmd": "metrics_subscribe"}))

def _on_message(_ws: websocket.WebSocketApp, message: str):
with suppress(json.JSONDecodeError):
metric = json.loads(message)
if "cmd" not in metric:
print(json.dumps(metric))

if __name__ == "__main__":
ws_app = websocket.WebSocketApp(
"ws://" + os.environ["WS_URL"],
on_open=_on_open,
on_message=_on_message,
)
while ws_app.run_forever():  # Returns False when the connection is closed
sleep(1)
```

You must specify the address and port of the websocket via environment variable `WS_URL`, e.g. start the script as follows:

```
WS_URL="127.0.0.1:8001" ./metrics_ws_receiver.py
```

This script will need to be run in parallel with the gNB to successfully print the metrics. The metrics will be written to `stdout`.

You can download the source-code for the above Python script here.

The metrics are generated on a layer basis, and each layer can be enabled through the configuration file.

```
metrics:
layers:                    # Enable metrics per layer
enable_executor: true     # false to disable
enable_app_usage: true    # false to disable
enable_ngap: true         # false to disable
enable_rrc: true          # false to disable
enable_e1ap: true         # false to disable
enable_pdcp: true         # false to disable
enable_sched: true        # false to disable
enable_rlc: true          # false to disable
enable_mac: true          # false to disable
enable_du_low: true       # false to disable
enable_ru: true           # false to disable
```

MAC metrics

The MAC metrics are encapsulated in the DU metric object, which contains the following keys:

- `du`: Includes the DU high metrics under the `du_high` key, which contains the `mac` metrics, which in turn contains `dl` metrics, which is a list of cells with specific MAC dowlink metrics. Each entry in the `dl` list corresponds to a cell configured in the DU.
- `timestamp`: Date and time at which the metrics were generated.

and is structured as shown below.

```
{
"du": {
"du_high": {
"mac": {
"dl": [   # List of cells, one item with MAC metrics per cell.
{
"average_latency_us": 45.473, "cpu_usage_percent": 0.0045473, "max_latency_us": 255.174,
"min_latency_us": 6.918, "pci": 1
},
{
"average_latency_us": 44.533, "cpu_usage_percent": 0.0044533, "max_latency_us": 232.088,
"min_latency_us": 5.822, "pci": 2
}
]
}
}
},
"timestamp": "2025-11-04T15:51:26.845"
}
```

The table below describes the MAC metrics reported in the DU metrics JSON output, under the `dl` key.

MAC metrics	Description
`pci`	Physical cell ID.
`average_latency_us`	Average latency of the MAC at handling slot indications, in microseconds.
`cpu_usage_percent`	Average CPU usage percentage of the MAC handling slot indications.
`min_latency_us`, `max_latency_us`	Minimum/Maximum latency of the MAC at handling slot indications, in microseconds.

Scheduler metrics

The scheduler metrics object contains 2 keys:

- `cells`: List of cell metrics objects, one per cell configured in the DU; this list is accessed through the key `cells`, which is structured similarly as shown below:

- `timestamp`: Date and time at which the metrics were generated.

```
{
"cells": [   # List of cells, one metric object per cell.
{
"cell_metrics": {
"average_latency": 3,
...
},
"event_list": [   # List of events for this cell.
{
"event_type": "ue_create",
"rnti": 17921,
"slot": "50.5"
},
...
],
"ue_list": [   # List of UEs for this cell with their metrics.
{
"avg_ce_delay": 3.75,
...
}
]
},
...
],
"timestamp": "2025-11-04T02:12:48.105"
}
```

Each element of the `cells` list contains the metrics of the cell, which can be accessed through the following keys:

- `cell_metrics`: reports the cell metrics for a given cell; refer to Scheduler cell metrics for details.
- `event_list`: list of events reported in this cell metric report. Refer to Cell events for details.
- `ue_list`: reports the list of UEs connected to this cell; each UE entry contains the UE metrics described in Scheduler UE metrics.

Scheduler cell metrics

Scheduler cell metric	Description
`error_indication_count`	Number of error indications received by the scheduler from lower layers.
`average_latency`, `max_latency`	Average/Maximum scheduler decision latency in microseconds.
`nof_failed_pdcch_allocs`	Number of failed PDCCH allocation attempts.
`nof_failed_uci_allocs`	Number of failed UCI allocation attempts.
`latency_histogram`	Scheduler decision latency histogram: each bucket counts the number of scheduling decisions that took a latency within a specific range. Each bucket range is 50 microseconds, starting from 0 microseconds up to 500 microseconds.
`msg3_nof_ok`, `msg3_nof_nok`	Number of MSG3s correctly received/decoded (CRC=OK) / not correctly received/decoded (CRC=KO).
`avg_prach_delay`	Average PRACH delay in slots: it measures the time between the slot when the PRACH is received by PHY layer and the slot when the PRACH indication reach the scheduler. Value is null if no PRACH was received during the measurement period.
`late_dl_harqs`	Number of failed PDSCH allocations due to late HARQs.
`late_ul_harqs`	Number of failed PUSCH allocations due to late HARQs.
`pucch_tot_rb_usage_avg`	Average number of RBs per UL slot used for PUCCH grants.
`pusch_prbs_used_per_tdd_slot_idx`	Sum of the number of RBs per slot used for PUSCH grants. Each entry in the array corresponds to a TDD slot index.
`pdsch_prbs_used_per_tdd_slot_idx`	Sum of the number of RBs per slot used for PDSCH grants. Each entry in the array corresponds to a TDD slot index.
`ue_list`	List of UE metrics reported in this cell metric report. Refer to Scheduler UE metrics for details.

Cell events

Metric	Description
`event_type`	Possible events are: ue_create for UE creation, ue_reconf for UE reconfiguration, ue_rem for UE removal.
`rnti`	RNTI of the UE for which the event occured.
`slot`	Slot at which the creation/reconfiguration/removal procedure was completed in the scheduler.

Scheduler UE metrics

Metric	Description
`ue`	UE index in the DU for this UE.
`pci`	Physical cell id.
`rnti`	Currently used C-RNTI for this UE.
`cqi`	Mean Channel Quality Indicator.
`dl_ri`	Mean Downlink Rank Indicator.
`ul_ri`	Mean Uplink Rank Indicator.
`dl_mcs`	Average MCS index used for DL grants.
`dl_brate`	Experienced MAC DL bit rate in kbps, considering the size of the allocated MAC DL PDUs for which a positive HARQ-ACK was received.
`dl_nof_ok`	Number of positive DL HARQ-ACKs received.
`dl_nof_nok`	Number of detected DL HARQ NACKs or DL HARQ-ACK misdetections.
`dl_bs`	Sum of the last DL buffer occupancy reports of all logical channels.
`pusch_snr_db`	SNR in dB estimated for the PUSCH.
`pusch_rsrp_db`	RSRP in dB estimated for the PUSCH.
`pucch_snr_db`	SNR in dB estimated for the PUCCH.
`ta_ns`	Average Timing Advance in nanoseconds: it includes the TA offset for all UL grants.
`pusch_ta_ns`	Average Timing advance in nanoseconds: it includes the TA offset for PUSCH grants only.
`pucch_ta_ns`	Average Timing advance in nanoseconds: it includes the TA offset for PUCCH grants only.
`srs_ta_ns`	Average Timing advance in nanoseconds: it includes the TA offset for SRS grants only.
`ul_mcs`	Average MCS index used for UL grants.
`ul_brate`	Experienced MAC UL bit rate in kbps, considering the size of the allocated MAC UL PDUs for which the respective CRC was decoded.

Metric	Description
`ul_nof_ok`	Number of PUSCH grants positively decoded (CRC=OK).
`ul_nof_nok`	Number of PUSCH grants that could not be decoded (CRC=KO).
`last_phr`	Last PHR value reported.
`max_pusch_distance`	Maximum interval in slots between 2 consecutive PUSCH allocated to this UE.
`max_pdsch_distance`	Maximum interval in slots between 2 consecutive PDSCH allocated to this UE.
`bsr`	Sum of the last UL buffer status reports (BSRs) of all logical channel groups.
`nof_pucch_f0f1_invalid_harqs`	Number of invalid UCI carrying HARQ-ACK bits detected on PUCCH format 0 and 1.
`nof_pucch_f2f3f4_invalid_harqs`	Number of invalid UCI carrying HARQ-ACK bits detected on PUCCH format 2, 3 and 4.
`nof_pucch_f2f3f4_invalid_csis`	Number of invalid UCI carrying bits CSIs detected on PUCCH format 2, 3 and 4.
`nof_pusch_invalid_harqs`	Number of invalid UCI carrying HARQ-ACK bits detected on PUSCH.
`nof_pusch_invalid_csis`	Number of invalid UCI carrying CSI bits detected on PUSCH.
`avg_ce_delay`, `max_ce_delay`	Average/Maximum MAC CE decoding delay in ms: it measures the time between the slot when the MAC PDU carrying the MAC CE is received and the slot when the MAC CE processing is completed.
`avg_crc_delay`, `max_crc_delay`	Average/Maximum CRC decoding delay in ms: it measures the time between the slot when the MAC PDU carrying the PUSCH is received and the slot when the PUSCH processing is completed.
`avg_pusch_harq_delay`, `max_pusch_harq_delay`	Average/Maximum PUSCH HARQ delay in ms: it measures the time between the slot when the PUSCH is received and the slot when the UCI carrying HARQ-ACK bits for that PUSCH decoding is completed.
`avg_pucch_harq_delay`, `max_pucch_harq_delay`	Average/Maximum PUCCH HARQ delay in ms: it measures the time between the slot when the PUCCH is received and the slot when the UCI carrying HARQ-ACK bits for that PUCCH decoding is completed.

Metric	Description
`avg_sr_to_pusch_delay`, `max_sr_to_pusch_delay`	Average/Maximum SR to PUSCH delay in ms: it measures the time between the slot when the SR is received and the slot when the first PUSCH (following the SR) is allocated for this UE.

OFH metrics

The structured OFH metrics are shown below:


```
{
"timestamp": "2025-11-04T15:51:26.845",
"ru": {
"ofh": {
"timing_stats": {
"nof_skipped_symbols": 0,
"skipped_symbols_max_burst": 0
},
"cells": [
{
"pci": 0,
"ul": {
"received_packets": {
"total": 0,
"early": 0,
"on_time": 0,
"late": 0,
"earliest_msg_us": 0.0,
"latest_msg_us": 0.0
},
"ethernet_receiver": {
"average_throughput_mbps": 0.0,
"average_latency_us": 0.0,
"max_latency_us": 0.0,
"cpu_usage_percent": 0.0
},
"message_decoder": {
"prach": {
"nof_dropped_messages": 0,
"average_latency_us": 0.0,
"max_latency_us": 0.0,
"cpu_usage_percent": 0.0
},
"data": {
"nof_dropped_messages": 0,
"average_latency_us": 0.0,
"max_latency_us": 0.0,
"cpu_usage_percent": 0.0
},
"ecpri": {
"nof_future_seqid_messages": 0,
"nof_past_seqid_messages": 0
}
},
```

```
"rx_window_stats": {
"nof_missed_uplink_symbols": 0,
"nof_missed_prach_occasions": 0
}
},
"dl": {
"ethernet_transmitter": {
"average_throughput_mbps": 0.0,
"average_latency_us": 0.0,
"max_latency_us": 0.0,
"cpu_usage_percent": 0.0
},
"message_encoder": {
"dl_cp": {
"average_latency_us": 0,
"max_latency_us": 0,
"cpu_usage_percent": 0
},
"ul_cp": {
"average_latency_us": 0,
"max_latency_us": 0,
"cpu_usage_percent": 0
},
"dl_up": {
"average_latency_us": 0,
"max_latency_us": 0,
"cpu_usage_percent": 0
},
},
"message_transmitter": {
"average_latency_us": 0,
"max_latency_us": 0,
"cpu_usage_percent": 0
},
"transmitter_stats": {
"late_dl_slots": 0,
"late_ul_slot_requests": 0,
"late_cp_dl_messages": 0,
"late_up_dl_messages": 0,
"late_cp_ul_messages": 0
}
}
}
]
}
}
}
```


The OFH metrics are structured into the following sections:

- `timing_stats` representing the timing statistics.
- `cells` representing a list of cells metrics.

The `timing_stats` metrics are shown in the following table:

OFH timing_stats metrics	Description
`nof_skipped_symbols`	Number of symbols skipped when the timing worker wakes up late.
`skipped_symbols_max_burst`	Maximum number of skipped symbols in a burst.

A cell is composed of three elements as it is shown in the following table:

OFH cells metrics	Description
`pci`	Physical Cell Identifier (described in 3GPP TS 38.331). Supported [0 - 1007].
`ul`	Object holding the Uplink metrics for the given cell.
`dl`	Object holding the Downlink metrics for the given cell.

The Uplink metrics are structured in the following sections:

- `received_packets`
- `ethernet_receiver`
- `message_decoder`
- `rx_window_stats`

The Received packets metrics for Uplink are described in the following table:

OFH Uplink cells received packets metrics	Description
`total`	Total number of received packets.
`early`	Number of OFH packets received early.
`on_time`	Number of OFH packets received on time.
`late`	Number of OFH packets received late.
`earliest_msg_us`	Earliest received packet in microseconds.
`latest_msg_us`	Latest received packet in microseconds.

The Ethernet receiver metrics for Uplink is described in the following table:

OFH cell Ethernet receiver metrics	Description
`average_throughput_mbps`	Average received throughput during metrics period represented in megabits per second.
`average_latency_us`	Average latency of packet reception in microseconds.
`max_latency_us`	Maximum latency of packet reception in microseconds.
`cpu_usage_percent`	CPU usage in percentage of the Ethernet receiver.

The Uplink message decoder metrics are structured in the following sections:

- `prach`
- `data`
- `ecpri`

The PRACH metrics for Uplink message decoder are described in the following table:

OFH cell PRACH messages decoder metrics	Description
`nof_dropped_messages`	Number of dropped PRACH messages.
`average_latency_us`	Average latency of PRACH message decoding.
`max_latency_us`	Maximum latency of PRACH message decoding.
`cpu_usage_percent`	CPU usage in percentage of PRACH messages decoding.

The Data metrics for Uplink message decoder are described in the following table:

OFH cell Data messages decoder metrics	Description
`nof_dropped_messages`	Number of dropped Data messages.
`average_latency_us`	Average decoding latency of the received Data messages.
`max_latency_us`	Maximum decoding latency of the received Data messages.
`cpu_usage_percent`	CPU usage in percentage of the Data flow.

The eCPRI metrics for Uplink message decoder are described in the following table:

OFH Uplink cells message decoder eCPRI metrics	Description
`nof_future_seqid_messages`	Number of skipped messages by the eCPRI.
`nof_past_seqid_messages`	Number of dropped messages by the eCPRI.

The Receiver window statistics metrics for Uplink are described in the following table:

OFH Uplink cells Receiver window statistics metrics	Description
`nof_missed_uplink_symbols`	Number of requested uplink symbols that were not fully received within the expected reception window.
`nof_missed_prach_occasions`	Number of PRACHs that were not fully received within the expected reception window.

The Downlink metrics are structured in the following sections:

- `ethernet_transmitter`
- `message_encoder`
- `message_transmitter`
- `transmitter_stats`

The Ethernet transmitter metrics for Downlink are described in the following table:

OFH cell Ethernet transmitter metrics	Description
`average_throughput_mbps`	Average transmitted throughput during metrics period represented in megabits per second.
`average_latency_us`	Average latency of packet transmission in microseconds.
`max_latency_us`	Maximum latency of packet transmission in microseconds.
`cpu_usage_percent`	CPU usage in percentage of the Ethernet transmitter.

The Downlink message encoder metrics are structured in the following sections:

- `dl_cp`
- `ul_cp`
- `dl_up`

The message encoder Downlink Control-Plane metrics are described in the following table:

OFH cell Downlink Control-Plane messages encoder metrics	Description
`average_latency_us`	Average latency of the Downlink Control-Plane processing in microseconds.
`max_latency_us`	Maximum latency of the Downlink Control-Plane processing in microseconds.
`cpu_usage_percent`	CPU usage in percentage of the Downlink Control-Plane processing.

The message encoder Uplink Control-Plane metrics are described in the following table:

OFH cell Uplink Control-Plane messages encoder metrics	Description
`average_latency_us`	Average latency of the Uplink Control-Plane processing in microseconds.
`max_latency_us`	Maximum latency of the Uplink Control-Plane processing in microseconds.
`cpu_usage_percent`	CPU usage in percentage of the Uplink Control-Plane processing.

The Message encoder Downlink User-Plane metrics are described in the following table:

OFH cell Downlink User-Plane messages encoder metrics	Description
`average_latency_us`	Average latency of the Downlink User-Plane processing in microseconds.
`max_latency_us`	Maximum latency of the Downlink User-Plane processing in microseconds.
`cpu_usage_percent`	CPU usage in percentage of the Downlink User-Plane processing.

The Message transmitter metrics for Downlink are described in the following table:

OFH cell messages transmitter metrics	Description
`average_latency_us`	Average latency of messages processing (including Ethernet transmission) performed at each symbol boundary.
`max_latency_us`	Maximum latency of messages processing (including Ethernet transmission) performed at each symbol boundary.
`cpu_usage_percent`	CPU usage in percentage of the messages transmitter.

The Transmitter statistics for Downlink is described in the following table:

OFH Downlink cells transmitter statistics metrics	Description
`late_dl_slots`	Number of late downlink resource grids received from the PHY.
`late_ul_slot_requests`	Number of late uplink request.
`late_cp_dl_messages`	Number of late Control-Plane downlink messages, i.e., messages that were not transmitted.
`late_up_dl_messages`	Number of late User-Plane downlink messages, i.e., messages that were not transmitted.
`late_cp_ul_messages`	Number of late Control-Plane uplink messages, i.e., messages that were not transmitted.

NGAP metrics

The NGAP metrics are encapsulated in the CU-CP metric object, which contains the following keys:

- `cu-cp`: Includes the CU-CP ID under the `id` key, NGAP metrics under the `ngaps` key, and RRC metrics under the `rrcs` key.
- `timestamp`: Date and time at which the metrics were generated.

and is structured as shown below.

```
{
"cu-cp": {
"id": "srs-cu-cp",
"ngaps": {
"ngap": [  # List of NGAP instances for connected AMFs.
...
],
"nof_handover_preparations_requested": 0,
"nof_successful_handover_preparations": 0
```

```
},
"rrcs": {
"du": [  # List of RRC instances for connected DUs.
...
],
"nof_handover_executions_requested": 0,
"nof_successful_handover_executions": 0
}
},
"timestamp": "2025-11-04T15:51:26.845"
}
```


The NGAP metrics are structured as shown below.


```
{
"ngaps": {
"ngap": [
{
"amf_name": "open5gs-amf0",
"connected": true,
"paging_measurement": {
"nof_cn_initiated_paging_requests": 0
},
"pdu_session_management": {
"nof_pdu_sessions_failed_to_setup": {
"misc-ctrl_processing_overload": 0,
"misc-hardware_fail": 0,
"misc-not_enough_user_plane_processing_res": 0,
"misc-om_intervention": 0,
"misc-unknown_plmn_or_sn_pn": 0,
"nas-authentication_fail": 0,
"nas-deregister": 0,
"nas-normal_release": 0,
"nas-unspecified": 0,
"protocol-abstract_syntax_error_falsely_constructed_msg": 0,
"protocol-abstract_syntax_error_ignore_and_notify": 0,
"protocol-abstract_syntax_error_reject": 0,
"protocol-msg_not_compatible_with_receiver_state": 0,
"protocol-semantic_error": 0,
"protocol-transfer_syntax_error": 0,
"protocol-unspecified": 0,
"radio_network-cag_only_access_denied": 0,
"radio_network-cell_not_available": 0,
"radio_network-encryption_and_or_integrity_protection_algorithms_not_supported": 0,
"radio_network-fail_in_radio_interface_proc": 0,
"radio_network-ho_cancelled": 0,
"radio_network-ho_desirable_for_radio_reason": 0,
"radio_network-ho_fail_in_target_5_gc_ngran_node_or_target_sys": 0,
"radio_network-ho_target_not_allowed": 0,
"radio_network-ims_voice_eps_fallback_or_rat_fallback_triggered": 0,
"radio_network-inconsistent_remote_ue_ngap_id": 0,
"radio_network-inconsistent_slice_info_for_the_session": 0,
"radio_network-indicated_mbs_session_area_info_not_served_by_the_gnb": 0,
"radio_network-insufficient_ue_cap": 0,
"radio_network-interaction_with_other_proc": 0,
"radio_network-invalid_qos_combination": 0,
"radio_network-misaligned_assoc_for_multicast_unicast": 0,
"radio_network-multiple_location_report_ref_id_instances": 0,
```

```
"radio_network-multiple_pdu_session_id_instances": 0,
"radio_network-multiple_qos_flow_id_instances": 0,
"radio_network-n26_interface_not_available": 0,
"radio_network-ng_inter_sys_ho_triggered": 0,
"radio_network-ng_intra_sys_ho_triggered": 0,
"radio_network-no_radio_res_available_in_target_cell": 0,
"radio_network-not_supported_5qi_value": 0,
"radio_network-npn_access_denied": 0,
"radio_network-partial_ho": 0,
"radio_network-radio_conn_with_ue_lost": 0,
"radio_network-radio_res_not_available": 0,
"radio_network-redcap_ue_not_supported": 0,
"radio_network-redirection": 0,
"radio_network-reduce_load_in_serving_cell": 0,
"radio_network-release_due_to_5gc_generated_reason": 0,
"radio_network-release_due_to_cn_detected_mob": 0,
"radio_network-release_due_to_ngran_generated_reason": 0,
"radio_network-release_due_to_pre_emption": 0,
"radio_network-res_not_available_for_the_slice": 0,
"radio_network-res_optim_ho": 0,
"radio_network-rsn_not_available_for_the_up": 0,
"radio_network-slice_not_supported": 0,
"radio_network-successful_ho": 0,
"radio_network-time_crit_ho": 0,
"radio_network-tngrelocoverall_expiry": 0,
"radio_network-tngrelocprep_expiry": 0,
"radio_network-txnrelocoverall_expiry": 0,
"radio_network-ue_context_transfer": 0,
"radio_network-ue_in_rrc_inactive_state_not_reachable": 0,
"radio_network-ue_max_integrity_protected_data_rate_reason": 0,
"radio_network-unknown_local_ue_ngap_id": 0,
"radio_network-unknown_mbs_session_id": 0,
"radio_network-unknown_pdu_session_id": 0,
"radio_network-unknown_target_id": 0,
"radio_network-unkown_qos_flow_id": 0,
"radio_network-unspecified": 0,
"radio_network-up_confidentiality_protection_not_possible": 0,
"radio_network-up_integrity_protection_not_possible": 0,
"radio_network-user_inactivity": 0,
"radio_network-xn_ho_triggered": 0,
"transport-transport_res_unavailable": 0,
"transport-unspecified": 0
},
"nof_pdu_sessions_requested_to_setup": 1,
"nof_pdu_sessions_successfully_setup": 1,
"s-nssai": "(sst=1 sd=na)"
},
"supported_plmns": "[ 00101, 99902, ]"
}
],
"nof_handover_preparations_requested": 0,
"nof_successful_handover_preparations": 0
}
}
```


The table below describes the NGAP metrics reported in the CU-CP metrics JSON output, under the `ngaps` key.

NGAP metrics	Description
`ngap`	A list of NGAP instances for connected AMFs.
`nof_handover_preparations_requested`	Number of handover preparations requested, see TS 128.552 section 5.1.3.7.1.1.
`nof_successful_handover_preparations`	Number of successful handover preparations, see TS 128.552 section 5.1.3.7.1.2.

The `ngap` list contains one object per connected AMF, structured as shown below.

NGAP metrics	Description
`amf_name`	Name of the related AMF.
`connected`	AMF connection status.
`paging_measurement`	Contains number of CN initiated paging requests, see TS 128.552 section 5.1.1.27.1.
`pdu_session_management`	Contains: : * number of PDU sessions requested to setup, see TS 128.552 section 5.1.1.5.1 * number of PDU sessions successfully setup, see TS 128.552 section 5.1.1.5.2 * number of PDU Sessions failed to setup, see TS 128.552 section 5.1.1.5.3 * the S-NSSAI associated with the PDU sessions.
`supported_plmns`	Contains a list of supported PLMNs for the AMF.

RRC metrics

The RRC metrics are encapsulated in the CU-CP metric object, which contains the following keys:

- `cu-cp`: Includes the CU-CP ID under the `id` key, NGAP metrics under the `ngaps` key, and RRC metrics under the `rrcs` key.
- `timestamp`: Date and time at which the metrics were generated.

and is structured as shown below.

```
{
"cu-cp": {
"id": "srs-cu-cp",
"ngaps": {
"ngap": [  # List of NGAP instances for connected AMFs.
...
],
"nof_handover_preparations_requested": 0,
```

```
"nof_successful_handover_preparations": 0
},
"rrcs": {
"du": [  # List of RRC instances for connected DUs.
...
],
"nof_handover_executions_requested": 0,
"nof_successful_handover_executions": 0
}
},
"timestamp": "2025-11-04T15:51:26.845"
}
```


The RRC metrics are structured as shown below.


```
{
"rrcs": {
"du": [
{
"gnb_du_id": 0,
"rrc_connection_establishment": {
"attempted_rrc_connection_establishments": {
"emergency": 0,
"high_prio_access": 0,
"mcs_prio_access": 0,
"mo_data": 0,
"mo_sig": 1,
"mo_sms": 0,
"mo_video_call": 0,
"mo_voice_call": 0,
"mps_prio_access": 0,
"mt_access": 0
},
"successful_rrc_connection_establishments": {
"emergency": 0,
"high_prio_access": 0,
"mcs_prio_access": 0,
"mo_data": 0,
"mo_sig": 1,
"mo_sms": 0,
"mo_video_call": 0,
"mo_voice_call": 0,
"mps_prio_access": 0,
"mt_access": 0
}
},
"rrc_connection_number": {
"max_nof_rrc_connections": 1,
"mean_nof_rrc_connections": 1
},
"rrc_connection_reestablishment": {
"attempted_rrc_connection_reestablishments": 0,
"successful_rrc_connection_reestablishments_with_ue_context": 0,
"successful_rrc_connection_reestablishments_without_ue_context": 0
}
}
],
"nof_handover_executions_requested": 0,
```

```
"nof_successful_handover_executions": 0
}
}
```

The table below describes the RRC metrics reported in the CU-CP metrics JSON output, under the `rrcs` key.

RRC metrics	Description
`du`	A list of RRC instances for connected DUs.
`nof_handover_executions_requested`	Number of handover executions requested, see TS 128.552 section 5.1.1.6.2.1.
`nof_successful_handover_executions`	Number of successful handover executions, see TS 128.552 section 5.1.1.6.2.2.

The `du` list contains one object per connected DU, structured as shown below.

RRC metrics	Description
`gnb_du_id`	ID of the related DU.
`rrc_connection_establishment`	RRC connection establisment metrics, containing: : * number of attempted RRC connection establishments, see TS 128.552 section 5.1.1.15.1. * number of successful RRC connection establishments, see TS 128.552 section 5.1.1.15.2.
`rrc_connection_number`	Contains RRC connections statistics, containing: : * mean number of RRC Connections, see TS 128.552 section 5.1.1.4.1. * max number of RRC Connections, see TS 128.552 section 5.1.1.4.2.
`rrc_connection_reestablishment`	RRC connection reestablisment metrics, containing: : * number of RRC connection re-establishment attempts, see TS 128.552 section 5.1.1.17.1. * number of successful RRC connection reestablishment with UE context, see TS 128.552 section 5.1.1.17.2. * number of successful RRC connection reestablishment without UE context, see TS 128.552 section 5.1.1.17.3.

Executors metrics

The structured Executors metrics are shown below:

```
{
"timestamp": "2025-11-04T15:51:26.845",
```

```
"executor_metrics": {
"name": "",
"nof_executes": 0,
"nof_defers": 0,
"enqueue_avg": 0,
"enqueue_max": 0,
"task_avg": 0,
"task_max": 0,
"cpu_load": 0,
"nof_vol_ctxt_switch": 0,
"nof_invol_ctxt_switch": 0.0
}
}
```


The table below describes the Executors metrics.

Executors metrics	Description
`name`	Executor name.
`nof_executes`	Number of times executor ran a task.
`nof_defers`	Number of times a task execution was deferred.
`enqueue_avg`	Average task enqueueing latency in microseconds.
`enqueue_max`	Maximum task enqueueing latency in microseconds.
`task_avg`	Average task duration in microseconds.
`task_max`	Maximum task duration in microseconds.
`cpu_load`	Average CPU usage percentage of the executor.
`nof_vol_ctxt_switch`	Number of voluntary context switches during the report period.
`nof_invol_ctxt_switch`	Number of involuntary context switches during the report period.

Application resource usage metrics

The structured Application resource usage metrics are shown below:


```
{
"timestamp": "2025-11-04T15:51:26.845",
"app_resource_usage": {
"cpu_usage_percent": 0,
"memory_usage_mb": 0,
"power_consumption_watts": 0
}
}
```

The table below describes the Application resource usage metrics.

Application resource usage metrics	Description
`cpu_usage_percent`	CPU usage of the application as a percentage. It can exceed 100% indicating how many CPU cores are utilized.
`memory_usage_mb`	Memory consumption of the application in Megabytes.
`power_consumption_watts`	CPU package power consumption in Watts measured while running the application.

Grafana Metrics GUI

> **INFO**
>
> The GrafanaGUI works with both the monolithic gNB and the DU application. The configuration changes required are the same for both applications.

OCUDU allows the reporting and visualization of the CU/DU metrics to a Grafana WebUI. This is done through the use of a Docker container that comes as standard with the OCUDU code base, located in the `docker/` folder, as shown here. This container allows users to bring up the Grafana dashboard in a single command.

Further Reading:

- Grafana Docs

Configuration

To use the Grafana webUI, you will first need to have Docker installed on your system, you will also need to modify the gNB or DU configuration file to allow the reporting of the metrics to the necessary JSON format for use in the webUI.

Docker

Using the Docker Containers included with OCUDU requires `docker compose` to be installed on your system. You can read about `docker compose` here.
There are multiple ways to install this, but the most basic way to do so is to install Docker Desktop. For installing Docker Desktop on linux, take a look at the Docker documentation.

NOTE

We recommend using Docker Compose V2 or later.

OCUDU

To enable the correct reporting of metrics to the Grafana UI, the gNB or DU configuration file needs to be updated to allow the metrics to be output in the correct JSON format and then sent through a udp-socket to the metrics-server, where it can be parsed and displayed correctly by the GUI.

The `addr` and `port` values defined above mirror those set in the `docker-compose.yml` file found in the `docker/` folder, as shown here. Any changes in these values must be kept consistent across both files.

Launching GUI

To launch the docker image for the Grafana UI, run the following command from the main folder containing OCUDU:

```
sudo docker compose -f docker/docker-compose.yml up grafana
```

The following output should be observed:

```
Creating network "docker_ran" with the default driver
Starting metrics_server ...
Starting metrics_server ... done
Creating grafana ... done
Attaching to grafana
```

Navigating to http://localhost:3300/ in your preferred web browser will allow you to view the UI.

You can then run OCUDU as normal. As the UE(s) connect to the network you will begin to see an output for each. These figures and graphics will update automatically during runtime, showing plots for each UE on the network.

GUI Output

A sample of the UI output can be seen here:

The above figure shows a single COTS UE connected to the network, with different traffic bursts of varying bandwidth being generated using iPerf. The cell bandwidth is 20 MHz.

Troubleshooting

For support and help using OCUDU, check out the issues section. The OCUDU issue tracker is also a rich source of general information.

What you need

To debug issues in an end-to-end deployment you will need access to most or all of the following:

- RAN components: Core, CU/DU, RU
- A COTS UE (physical or soft)
- A USRP (can substitute for a spectrum analyzer)
- A spectrum analyzer (optional but recommended for RF diagnostics)

From the OCUDU software itself, the most useful diagnostic outputs are:

- **Logs:** set to `info` or `debug` level. See Outputs for the log format.
- **PCAPs:** per-layer packet captures for MAC, RLC, NGAP, N3, E1AP, F1AP, E2AP.
- **Metrics:** per-layer runtime metrics, viewable via the Grafana GUI.
- **Console trace:** the real-time per-UE trace, activated by pressing `t` while the gNB is running.

To quickly scan a log file for warnings, use:

```
grep "\[W\]" <logfile>
```

Performance Tuning

The following sections outline key steps to improve gNB performance.

CPU Performance Mode

The CPU governor of the PC should be set to performance mode to allow for maximum compute power and throughput. This can be configured for e.g. Ubuntu using:

```
echo "performance" | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
```

It is also recommended that users running on a laptop keep the PC connected to a power-source at all times while running the gNB, as this will avoid performance loss due to CPU frequency scaling on the machine.

Performance Configuration Script

Before running the gNB application, we recommend tuning your system for best performance. We provide a script to configure known performance parameters:

- OCUDU performance script

The script does the following:

> 1. Sets the scaling governor to performance
> 2. Disables DRM KMS polling
> 3. Tunes network buffers (Ethernet based USRPs only)

Run the script as follows from the main project folder:

```
sudo ./scripts/ocudu_performance
```

USRP Configuration

Users should always ensure that the USRP they are using is running over USB 3.0 or ethernet and correctly configured. If UHD is built from source, users will have multiple example applications available in `/usr/lib/uhd/examples/`. User can verify their USRP is correctly configured by running the `uhd_benchmark` application as follows:

```
sudo ./benchmark_rate --rx_rate [rate in Hz] --tx_rate [rate in Hz]
```

More details can be found in this guide on the Ettus Knowledge Base.

USRP Time Calibration

Incorrect time calibration of a USRP can lead to preventing the gNB from receiving PRACH transmissions. The TX/RX time calibration adjusts the offset between TX and RX processing chains delay in the USRP. This value varies as a function of sampling rate, USRP model and UHD version. Users experiencing issues with incorrect time calibration will see a message similar to the following in the logs:

```
2023-03-08T08:38:34.130365 [UL-PHY0 ] [I] [ 1001.18] PRACH:  rssi=+0.5dB detected_preambles=
[{idx=55 ta=-7.29us power=+85.8dB snr=0.0dB}] t=351.3us
2023-03-08T08:38:34.130377 [FAPI    ] [E] [    0.0] Detected 1 errors in RACH.indication
```

```
message at slot=1001.18:
- Property=Timing advance offset in nanoseconds, value=-7291, expected value=[0-2005000]
```

In the above log the PRACH arrived 7.29 us early to the gNB, which at a sampling rate of 23.04 MHz is approximately 168 samples. The TX/RX offset needs to be adjusted to a higher value than this so that the PRACH arrives within the detection window. This can be done by adding the following to the config file under the `ru_sdr` options:

```
ru_sdr:
time_alignment_calibration: -170    # This will set an offset of -170 samples
```

In general, a larger negative value is better as it will make sure the PRACH falls within the detection window. The consequence of increasing this too much is that the effective cell size is reduced (this is not important for a lab set up).

By default the `time_alignment_calibration` parameter is set to 0. This means that in most SDR frontends the PRACH will arrive a few samples late within the window. With preamble format 0, there is enough space in the detection window and this does not cause any problem. However, if you are trying to set up a very large cell, or using different preamble formats, you might want to set a positive `time_alignment_calibration` value such that there is space in the window for UEs far in the cell.

UE Attach Failure

Try a wired connection first

When debugging a UE attach failure, start with a wired connection if one is available. This removes RF as a variable and isolates the problem to the network stack. COTS UEs such as Quectel modules and UE simulators such as amariUE can be wired directly to the RU.

Wireless UE placement

If a wired connection is not available, place the UE at an appropriate distance from the antenna:

- **Indoor RUs:** 3 to 5 m from the antenna
- **Outdoor RUs:** at least 5 m from the antenna

Placing the UE too close will saturate the receivers. Ensure there are no reflective objects between the UE and the RU, and that the line of sight is clear.

Network visible but UE cannot attach

If the UE can see the network but the attach procedure fails, there is likely an authentication issue between the UE and the core, or a PRACH configuration mismatch. Work through the following checks:

1. **Verify UE credentials.** Confirm that the IMSI, Ki, and OPC values are consistent between the SIM and the core network subscriber database.

2. **Verify APN credentials.** Check that the APN configured in the UE matches the APN configured in the core. Any discrepancy, including trailing spaces, will cause attach failure.

3. **Eject and reinsert the SIM.** This resets the UE's connection timers and can resolve attach failures once credentials have been verified.

4. **Check the PRACH configuration.** If the UE still cannot attach, check the gNB log for missing uplink-plane packets. If u-plane packets are absent, the PRACH configuration in the gNB is likely not matching the RU configuration. Verify that the PRACH settings in the gNB reflect those in the RU's configuration.

Connectivity Optimization

Once a UE is attached, use the following steps to verify and optimize throughput if the observed rate is below what is expected for the configured bandwidth.

Ping test

Ping the UE from the gNB and monitor latency and packet loss. High latency or packet loss generally indicates an RF link problem. Proceed to the UDP tests below to narrow down whether the issue is on the downlink or uplink.

UDP downlink test

Run a UDP downlink test and monitor the `nok`, `(%)`, and `mcs` columns in the gNB console trace.

- **Low error rate and high MCS:** downlink RF conditions are acceptable.
- **High error rate and fluctuating MCS:** there is an RF link problem. Check the gNB logs for warnings and review the RU KPIs.
- **Low error rate and high MCS, but maximum rate not reached:** check the `dl_bs` field. If it is consistently below 1 Mb, no data is being sent over the N3 interface. Check the core configuration and N3 backhaul traffic.

UDP uplink test

Run a UDP uplink test and monitor the `nok`, `(%)`, and `mcs` columns in the gNB console trace.

- **Low error rate and high MCS:** uplink RF conditions are acceptable.
- **High error rate and fluctuating MCS:** there is an RF link problem. Check the gNB logs for warnings and review the RU KPIs.

- **Low BLER and high MCS, but maximum rate not reached:** check the `bsr` field. If it is consistently below 300 Kb, the UE is not transmitting data. Use a spectrum analyzer to verify that the UE is producing a physical signal. If no signal is visible, check for physical issues with the UE.

UDP bidirectional test

Run a bidirectional UDP test to repeat the above checks for both uplink and downlink simultaneously.

Open Fronthaul Issues

OFH packet drops

OFH packet drops will appear in the gNB logs as warnings similar to the following:

```
[OFH ] [W] Detected '1' late downlink C-Plane messages in the transmitter queue for slot '841.6
[OFH ] [W] Missed incoming User-Plane PRACH messages for slot '0' and sector#0
```

or:

```
[OFH ] [W] Sector#2305618064: dropped received Open Fronthaul message as no uplink slot context
[OFH ] [W] Sector#2305618064: dropped received Open Fronthaul message as no uplink slot context
```

Work through the following steps to diagnose and fix the drops:

1. **Enable DPDK.** This is recommended for all O-RAN fronthaul deployments and will significantly improve packet processing stability. See the DPDK tutorial for setup instructions.

2. **Check thread usage with `htop`.** Monitor the OCUDU thread pool while the gNB is running.

   - If usage is consistently above 70% and warnings are present, increase the number of threads in the configuration file.
   - If usage is consistently below 30% and warnings are present, decrease the number of threads.

3. **Verify the `ru_ofh` configuration.** Confirm that the MAC addresses and VLAN tags in the `ru_ofh` section of the gNB configuration match those in the RU configuration file exactly.

4. **Verify PTP synchronization.** Check that `ptp4l` and `phc2sys` are running and within acceptable ranges:

   - `ptp4l` rms value should be around 10.
   - `phc2sys` offset should be in the range of -100 to +100. Ensure that the RU is locked to the PTP signal; this can be confirmed via the RU logs.

5. **Check compression settings.** Verify that the compression method and compression header settings are consistent across both the DU and RU configurations. A mismatch (for example, static compression header enabled in the RU but not in the DU) will produce similar warnings.

6. **Check the fronthaul switch.** Confirm that the switch is configured correctly, that all physical connections are secure and in the correct ports, and that VLAN tags are configured correctly. Refer to the relevant switch integration guide for your hardware.

7. **Check RU KPIs.** Verify that there are no sustained late or early packets in the RU KPI output. See RU late or early packets below.

RU late or early packets

The RU KPI output will show packet timing statistics. An example from a Benetel RU:

```
SAMPLE_TIME | RX_TOTAL | RX_ON_TIME | RX_EARLY | RX_LATE | RX_ON_TIME_C | RX_EARLY_C | RX_LATE_
23:27:02.15 | 89447 | 82367 | -0 | 119 | 6320 | -0 | 16 | -0
23:27:03.12 | 89483 | 82956 | -0 | 183 | 6388 | -0 | 4 | -0
23:27:04.27 | 98370 | 85830 | -0 | 26 | 7016 | -0 | -0 | -0
```

One or two intermittent early or late packets are normal. A sustained stream of late or early packets across multiple reporting periods indicates a configuration problem.

If late or early packets are caused by PTP issues, check that `ptp4l` and `phc2sys` are running correctly and within the acceptable ranges above. Reboot the RU if issues persist after PTP has been verified.

If PTP is healthy, adjust the `t1a_max` and `ta4_max` parameters in the `ru_ofh` section of the gNB configuration:

Symptom	Action
Early packets at the RU	`t1a_max` is too high; reduce it
Late packets at the RU	`t1a_max` is too low; increase it
Late packets at the DU	Increase `ta4_max`
Both early and late at the RU simultaneously	PTP clock is not properly synchronized

RU power and DPD saturation

If the RU transmit power is set too high, the Power Amplifier (PA) may reach Digital Pre-distortion (DPD) saturation. This causes the output signal to be distorted, which typically results in reduced UE throughput or an inability to connect to the network.

To diagnose this, refer to your RU's documentation to verify whether DPD saturation is occurring. If it is confirmed via the RU's own monitoring outputs, lower the `iq_scaling` parameter in the `ru_ofh` section of the gNB configuration file. Only adjust this value if you have confirmed that DPD saturation is the issue. If the problem persists after adjusting the parameter, consult your RU manufacturer's documentation.

FAQ

The following are frequently asked questions about OCUDU deployment and capacity planning. A key design principle of OCUDU is to be as agnostic and generic as possible: there are no strict hard-coded limits. The system is designed to scale up and down as a function of the available compute resources.

Are there separate OCUDU software packages for the CU and the DU, or are they installed on a single server?

Both CU and DU can be installed on a single server. Either as separate applications or as a co-located (all-in-one) gNB.

How many DUs are supported by one CU?

There is no hard-coded limit. However, the number of DUs accepted by a single CU can be limited with the `--max_nof_dus` parameter.

How many RUs / radio cells are supported by one DU?

There is no hard-coded limit. If the physical layer runs in software, the number of component-carriers/sectors/cells primarily depends on the cell bandwidth and MIMO layers supported by the underlying compute.

How many 5G UEs can be managed by the CU and DU?

There is also no hard limit. The maximum number of UEs depends on a number of cell parameters, especially in TDD networks the number of uplink resources is a critical dimensioning factor. However, OCUDU is constantly tested in scenarios that have 1500 UEs per carrier in active mode.