



**DEMONSTRAÇÃO DO SOFTWARE DE  
GERENCIAMENTO DA O-RU  
DEZEMBRO/2023**

A2.4 – Desenvolvimento de Software

OpenRAN@Brasil – Fase 2

## Sumário

Sumário.....	2
Lista de Figuras.....	3
Lista de Acrônimos.....	4
1 Introdução.....	5
1.1 Visão Geral do Equipamento.....	6
1.2 Visão Geral do Software de Gerenciamento da O-RU.....	7
2 Demonstração do Software de Gerenciamento Desenvolvido.....	8
2.1 Funcionalidades Demonstradas.....	11
3 Resultados Demonstrados.....	11
3.1 Função Call Home.....	12
3.2 Monitoração e Controle de Parâmetros do PA.....	13
3.3 Exemplos de Mensagens XML do Protocolo NETCONF.....	14
3.4 Processos Típicos Realizados em Uma Transação NETCONF.....	15
4 Conclusões.....	16
5 Histórico de Alterações do Documento Consolidado.....	17
6 Execução e Aprovação.....	18

## Lista de Figuras

<a href="#">Figura 1 – Visão geral da unidade de rádio de interface aberta em desenvolvimento.....</a>	<a href="#">6</a>
<a href="#">Figura 2 – Ambiente da demonstração do software de gerenciamento da O-RU.....</a>	<a href="#">8</a>
<a href="#">Figura 3 – Elementos utilizados para implementar a demonstração do software de gerenciamento.....</a>	<a href="#">8</a>
<a href="#">Figura 4 – Principais elementos da demonstração do software de gerenciamento no laboratório do CPQD.....</a>	<a href="#">9</a>
<a href="#">Figura 5 – Componentes de software integrados na O-RU para demonstração.....</a>	<a href="#">10</a>
<a href="#">Figura 6 – Resultados preliminares de integração, compilação cruzada e testes da solução Netoppeer2 + Sysrepo realizadas pelo CPQD na fase inicial do projeto.....</a>	<a href="#">10</a>
<a href="#">Figura 7 – Funcionalidades do software de gerenciamento demonstradas.....</a>	<a href="#">11</a>
<a href="#">Figura 8 – Demonstração da função Call Home com o M-Plane Controller aguardando uma O-RU se anunciar na rede (a) e as mensagens de texto geradas logo após o Call Home (b).....</a>	<a href="#">12</a>
<a href="#">Figura 9 – Demonstração da função Call Home com as mensagens de texto geradas pelo M-Plane Controller (a) e o diagrama de sequência correspondente (b).....</a>	<a href="#">13</a>
<a href="#">Figura 10 – Demonstração das funções de monitoração e controle do PA através de comandos inseridos via CLI (a) e o diagrama de sequência correspondente (b).....</a>	<a href="#">14</a>
<a href="#">Figura 11 – Trechos de mensagens XML do protocolo NETCONF geradas pelo NMS (a) e pela O-RU (b) quando o comando get pa power é inserido na CLI.....</a>	<a href="#">14</a>
<a href="#">Figura 12 – Representação da interação entre o NMS e o servidor NETCONF (Netopeer2 + Sysrepo).....</a>	<a href="#">15</a>

### Lista de Acrônimos

CLI	Command Line Interface
CU	Centralized Unit
DFE	Digital Front-End
DU	Distributed Unit
EVK	Evaluation Kit
IETF	Internet Engineering Task Force
IP	Intellectual Property
M-Plane	Management Plane
NETCONF	Network Configuration
NMS	Network Management System
O-RAN	Open Radio Access Network
O-RU	Open Radio Unit
PA	Power Amplifier
QEMU	Quick Emulation
RAN	Radio Access Network
RFSoc	Radiofrequency System-on-Chip
RIC	Radio Intelligent Controllers
RU	Radio Unit
XML	Extensible Markup Language
YANG	Yet Another Next Generation
5G NR	Fifth Generation New Radio

## 1 Introdução

Este documento apresenta uma visão geral da unidade de rádio de interface aberta (*open radio unit* – O-RU) em desenvolvimento no escopo do projeto OpenRAN@Brasil Fase 2 através de esforços conjuntos de diferentes instituições de ciência e tecnologia (CPQD, Inatel, Eldorado e RNP), bem como do seu software de gerenciamento, constituindo o chamado Plano de Gerenciamento (*management plane* – M-Plane) responsável pela coordenação eficiente e abrangente dos elementos de rede visando promover a interoperabilidade, eficiência operacional e flexibilidade na implementação e gestão de redes 5G.

A demonstração da integração preliminar de componentes desse software de gerenciamento sendo desenvolvido pelo CPQD, realizada em dezembro de 2023, é então apresentada através da descrição do ambiente laboratorial, detalhes de implementação, funcionalidades e resultados obtidos.

Alguns detalhes dos protocolos de comunicação, transações e modelos de dados padronizados utilizados nesse software também são apresentados, fornecendo assim ao leitor interessado um esclarecimento técnico mais aprofundado.

## 1.1 Visão Geral do Equipamento

O software de gerenciamento cuja demonstração é descrita nesse relatório é executado em uma O-RU — um equipamento sendo desenvolvido pelo CPQD, Inatel e Eldorado com acompanhamento e gestão da RNP. Sua macroarquitetura está representada na Figura 1.

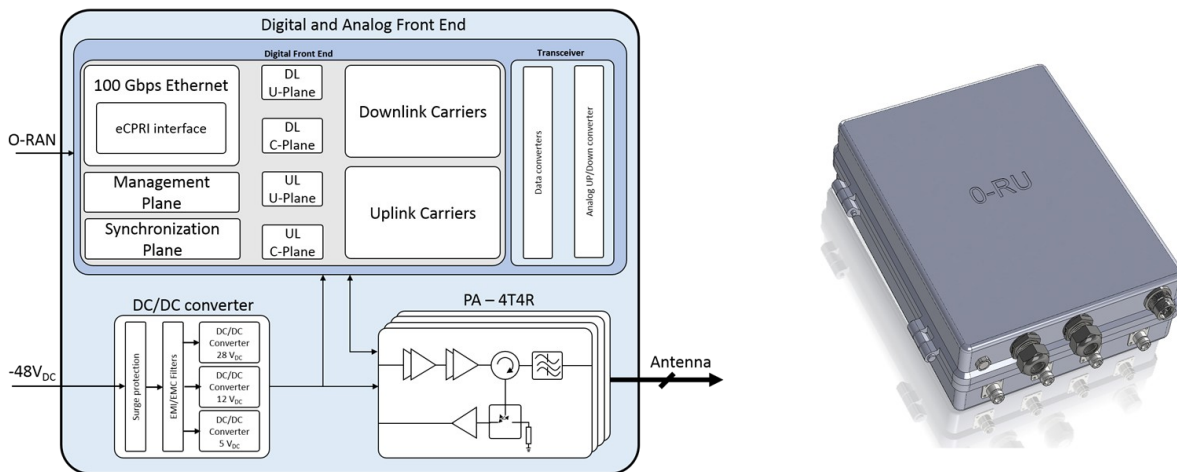


Figura 1 – Visão geral da unidade de rádio de interface aberta em desenvolvimento.

Trata-se de uma unidade de rádio 5G NR composta por um módulo conversor DC/DC que provê a alimentação elétrica para todos os componentes internos, um módulo de processamento digital (*digital front-end* – DFE), o qual implementa em hardware as principais funções da cadeia de processamento dos sinais de rádio definido por software e um módulo amplificador de potência (*power amplifier* – PA), responsável pela interface de radiofrequência conectando o DFE às antenas da O-RU.

Os requisitos dessa unidade em desenvolvimento foram estabelecidos para viabilizar a prestação de serviços de comunicação sem fio a partir de macrocélulas na faixa de sub-6 GHz, alinhados aos critérios estabelecidos pela Open Radio Access Network (O-RAN) Alliance.

As principais características que norteiam seu desenvolvimento são:

- Custo acessível;
- Elevada programabilidade;
- Atendimento a diversos nichos de mercado;
- Promoção do desenvolvimento de alta tecnologia no país.

Dentro desse contexto, a visão geral do equipamento destaca não apenas a inovação técnica, mas também a sinergia entre as instituições envolvidas, refletindo o compromisso com o avanço da tecnologia 5G Open RAN de maneira acessível, adaptável e alinhada às demandas do mercado.

## 1.2 Visão Geral do Software de Gerenciamento da O-RU

No âmbito do Open RAN 5G, o software de gerenciamento (constituindo o chamado “plano de gerenciamento”) desempenha um papel crucial na orquestração eficiente e na administração abrangente dos diversos elementos de rede, incluindo as unidades de rádio (*radio unit* – RU), as unidades distribuídas (*distributed unit* – DU), unidades centralizadas (*centralized unit* – CU) e controladores inteligentes de rádio (*radio intelligent controller* – RIC). Assim, o plano de gerenciamento pode ser entendido como um conjunto estruturado de diretrizes e protocolos que coordenam a configuração e operação de cada componente de rede, promovendo a interoperabilidade, eficiência operacional e flexibilidade na implementação e gestão de soluções 5G.

Em uma rede 5G Open RAN, cada elemento é dotado de um plano de gerenciamento específico — uma abordagem granular que permite a customização e otimização direcionada para as particularidades inerentes a cada componente, maximizando a eficiência operacional do sistema como um todo. Além disso, cada plano de gerenciamento lida com aspectos específicos da configuração e operação de seus respectivos elementos. Por exemplo, o plano de gerenciamento da RU pode abranger parâmetros relacionados à interface de rádio, enquanto o da CU pode se concentrar em aspectos mais centralizados, como a alocação de recursos e a coordenação geral da rede.

O software de gerenciamento abordado nesse relatório é focado no plano de gerenciamento da RU, o qual possui funcionalidades essenciais para a configuração, monitoração e otimização de desempenho da unidade. Uma de suas principais funções reside na administração dos parâmetros da interface de rádio, permitindo ajustes dinâmicos na potência de transmissão, modulação e configurações de formação de feixe (*beamforming*) para otimizar a qualidade e a eficiência da comunicação sem fio.

A funcionalidade de detecção e mitigação de interferências é outra característica vital do plano de gerenciamento da RU, monitorando ativamente o ambiente de rádio e aplicando estratégias para minimizar interferências e otimizar a qualidade do sinal.

Adicionalmente, a capacidade de realizar diagnósticos remotos e atualizações de software contribui para a manutenção proativa da RU, aumentando sua confiabilidade operacional.

Por fim, o software de gerenciamento da RU desempenha um papel central na coleta de dados operacionais, fornecendo informações cruciais para análises de desempenho, manutenção preditiva e adaptação dinâmica às demandas do tráfego de rede, sendo assim uma peça fundamental na orquestração eficiente e na maximização do potencial das redes 5G Open RAN.

## 2 Demonstração do Software de Gerenciamento Desenvolvido

Uma demonstração do software de gerenciamento da O-RU desenvolvido pelo CPQD foi realizada em dezembro/2023, integrando de forma preliminar alguns componentes baseados em soluções de código aberto.

Conforme ilustrado na Figura 2, um sistema de gerenciamento de rede (*network management system* – NMS) tem acesso a uma O-RU para realizar operações de leitura e configuração de parâmetros.

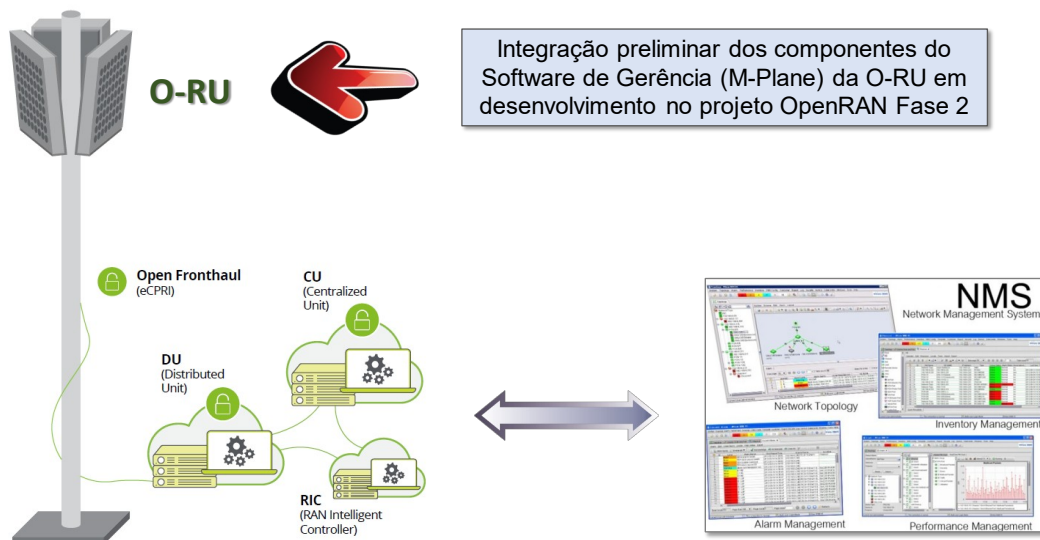


Figura 2 – Ambiente da demonstração do software de gerenciamento da O-RU.

A Figura 3 mostra como essa demonstração foi implementada em laboratório.

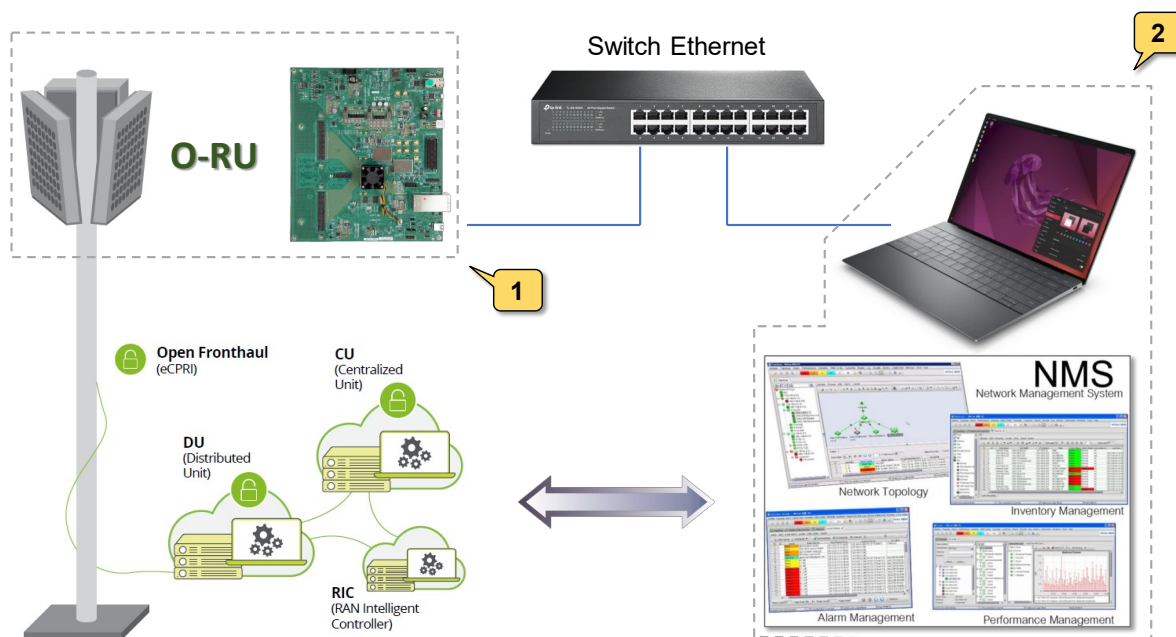


Figura 3 – Elementos utilizados para implementar a demonstração do software de gerenciamento.



O principal elemento (1), simulando uma O-RU, é um kit de avaliação (*evaluation kit – EVK*) de um sistema em chip de radiofrequência (*radiofrequency system-on-chip – RFSoC*) da AMD (Xilinx), modelo ZCU 670. Esse dispositivo possui blocos lógicos de propriedade intelectual (*IP cores*) dedicados para o processamento dos sinais digitais de rádio definido por software, além de uma unidade de processamento com núcleos ARM rodando uma versão dedicada do sistema operacional Linux (Petalinux) e softwares de aplicação, entre os quais a versão preliminar do software de gerenciamento desenvolvido.

O sistema de gerenciamento, por sua vez, é simulado por um notebook (2) com sistema operacional Linux Ubuntu rodando *scripts* na linguagem Python.

Ambos são interconectados através de uma interface Ethernet.

A Figura 4 mostra uma foto desses dois principais elementos na bancada de demonstração no laboratório de conectividade do CPQD.



Figura 4 – Principais elementos da demonstração do software de gerenciamento no laboratório do CPQD.

Os módulos de software integrados na O-RU para compor a demonstração do software de gerenciamento estão representados na Figura 5.

O servidor do plano de gerenciamento (M-Plane Server) integra um componente de código aberto denominado Netoppper2, responsável pelo estabelecimento de um servidor NETCONF (*NETwork CONFiguration*) — um protocolo de gerenciamento largamente utilizado pela indústria de equipamentos de redes de comunicação. Ele se conecta internamente a outro módulo também de código aberto denominado Sysrepo, o qual constitui um gerenciador de banco de dados dedicado ao armazenamento e gerenciamento de configurações baseadas em modelos YANG (*Yet Another Next Generation*) — uma linguagem de modelagem de dados que representa informações estruturadas em uma árvore XML (*Extensible Markup Language*).

Um modelo YANG específico de padrão aberto, padronizado pela IETF (*Internet Engineering Task Force*) e adotado pela O-RAN Alliance, foi utilizado nessa implementação: **ietf-hardware.yang**.

Componentes denominados YANG Handlers interagem com o sistema operacional e *drivers* do hardware para obter ou alterar os parâmetros solicitados pelo NMS. Nessa demonstração um deles foi desenvolvido especificamente para tratar parâmetros relacionados ao amplificador de potência (PA), o qual foi simulado em software visto que ainda se encontra em desenvolvimento.

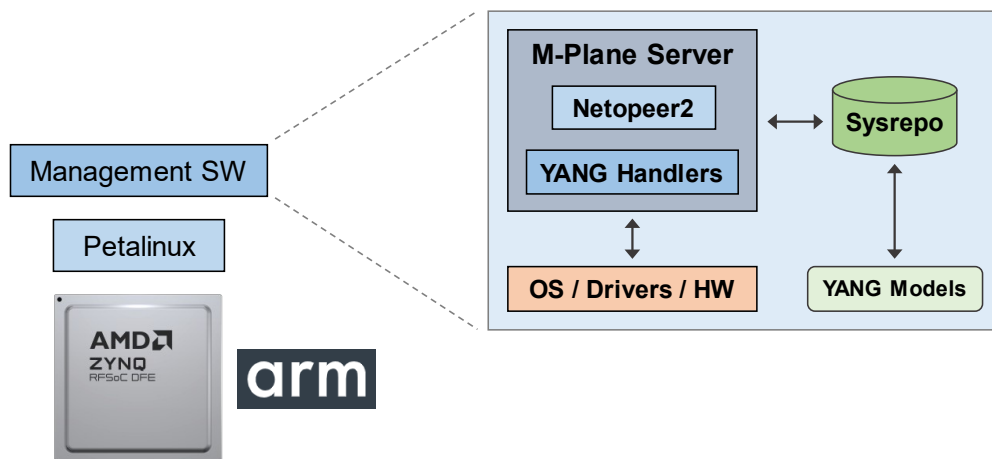


Figura 5 – Componentes de software integrados na O-RU para demonstração.

A estrutura do M-Plane Server foi aproveitada a partir de uma contribuição da Meta, desenvolvida em seu programa Evenstar, denominada Open M-Plane. Seu núcleo funcional (baseado no uso do Netopeer2 + Sysrepo) está alinhado às avaliações e experimentações realizadas pelo CPQD na fase inicial do projeto, conforme ilustrado na Figura 6, quando então esses componentes foram integrados, compilados para a plataforma ARM e testados em um ambiente virtualizado (ARM em X86) através da ferramenta QEMU (*Quick Emulation*).

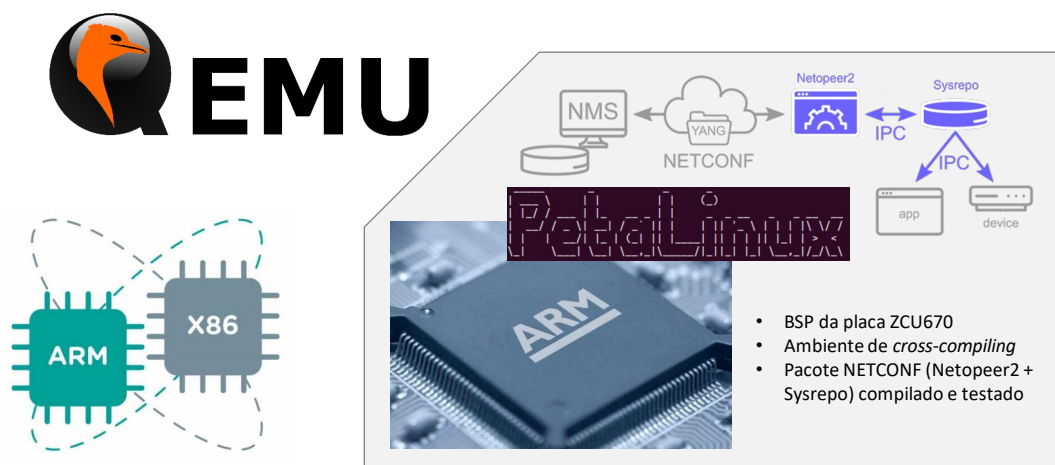


Figura 6 – Resultados preliminares de integração, compilação cruzada e testes da solução Netopeer2 + Sysrepo realizadas pelo CPQD na fase inicial do projeto.

Nesse ambiente de demonstração, ilustrado na Figura 3, os componentes de software em execução no módulo de processamento ARM do RFSoc presente na placa ZCU 670 constituem um servidor do plano de gerenciamento da O-RU, enquanto o notebook com seus *scripts* em Python age como um cliente (controlador) conectado à O-RU.

## 2.1 Funcionalidades Demonstradas

A Figura 7 mostra as funcionalidades do software de gerenciamento que foram demonstradas em ambiente laboratorial.

A primeira delas, denominada função “Call Home”, corresponde a um processo em que a O-RU se anuncia na rede depois do seu ciclo de inicialização (carga do sistema operacional e configurações iniciais) que ocorre logo após ser energizada, estabelecendo comunicação com um NMS (através da conexão com uma DU) e respondendo a requisições NETCONF que solicitam informações acerca de suas capacidades e componentes disponíveis para gerenciamento e monitoração.

As demais correspondem a funções de monitoração e controle de parâmetros de hardware, nesse caso habilitação, potência e temperatura do PA simulados em software.

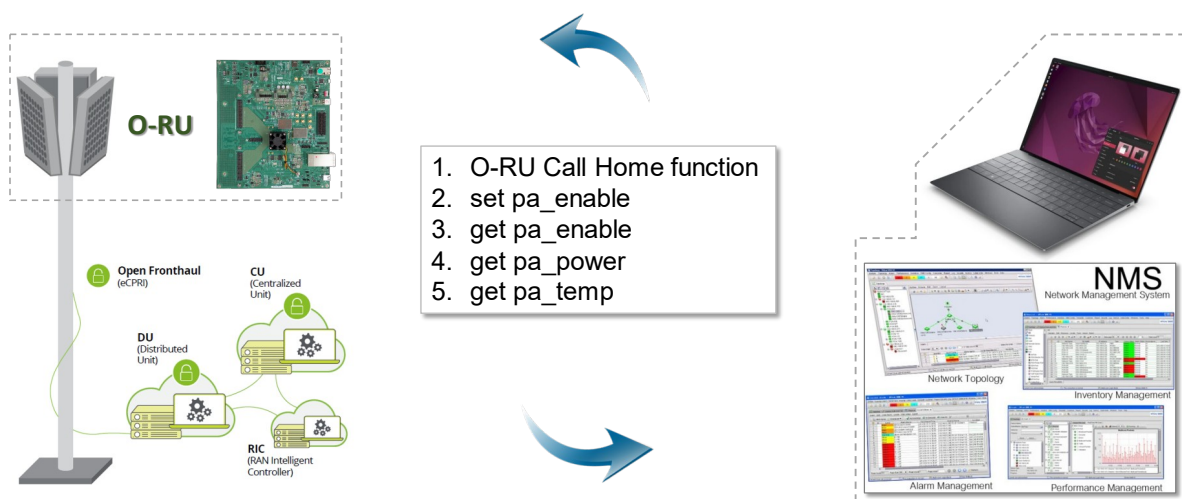


Figura 7 – Funcionalidades do software de gerenciamento demonstradas.

## 3 Resultados Demonstrados

Na demonstração da função Call Home podem ser observadas mensagens de texto na tela do notebook indicando informações como a inicialização da camada de transporte, o estabelecimento de uma conexão segura, entre outras.

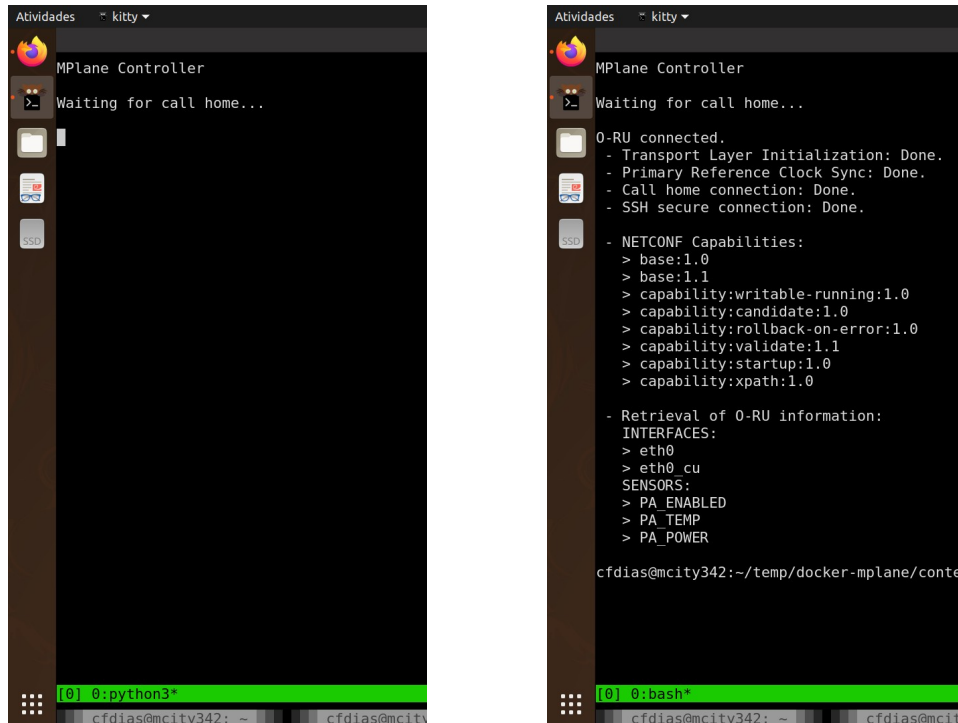
Para a demonstração das funções de monitoração e controle de parâmetros do PA, uma interface de linha de comando (*command line interface* – CLI) também foi desenvolvida em Python e executada no notebook.

Um usuário insere comandos *get* e *set* que invocam scripts adicionais responsáveis pela geração e interpretação de mensagens XML trocadas com a O-RU, correspondentes ao protocolo NETCONF.

A O-RU, por sua vez, também interpreta as mensagens XML recebidas, altera suas configurações e responde com os parâmetros requisitados.

### 3.1 Função Call Home

Na Figura 8 (a) observa-se o M-Plane Controller (atuando como um NMS) aguardando uma O-RU se anunciar na rede (Call Home). Em (b), logo após a energização da O-RU e seu ciclo de inicialização, ocorre o processo de Call Home e o M-Plane Controller exibe as mensagens correspondentes.



```
Atividades ~ kitty
MPlane Controller
> Waiting for call home...

[0] 0:python3*
cfdias@mcity342: ~
```

```
Atividades ~ kitty
MPlane Controller
> Waiting for call home...

O-RU connected.
- Transport Layer Initialization: Done.
- Primary Reference Clock Sync: Done.
- Call home connection: Done.
- SSH secure connection: Done.

- NETCONF Capabilities:
> base:1.0
> base:1.1
> capability:writable-running:1.0
> capability:candidate:1.0
> capability:rollback-on-error:1.0
> capability:validate:1.1
> capability:startup:1.0
> capability:xpath:1.0

- Retrieval of O-RU information:
INTERFACES:
> eth0
> eth0 cu
SENSORS:
> PA_ENABLED
> PA_TEMP
> PA_POWER

cfdias@mcity342:~/temp/docker-mplane/conter

[0] 0:bash*
cfdias@mcity342: ~
```

(a) (b)

Figura 8 – Demonstração da função Call Home com o M-Plane Controller aguardando uma O-RU se anunciar na rede (a) e as mensagens de texto geradas logo após o Call Home (b).

Esse processo também está representado por um diagrama de sequência na Figura 9 (b).

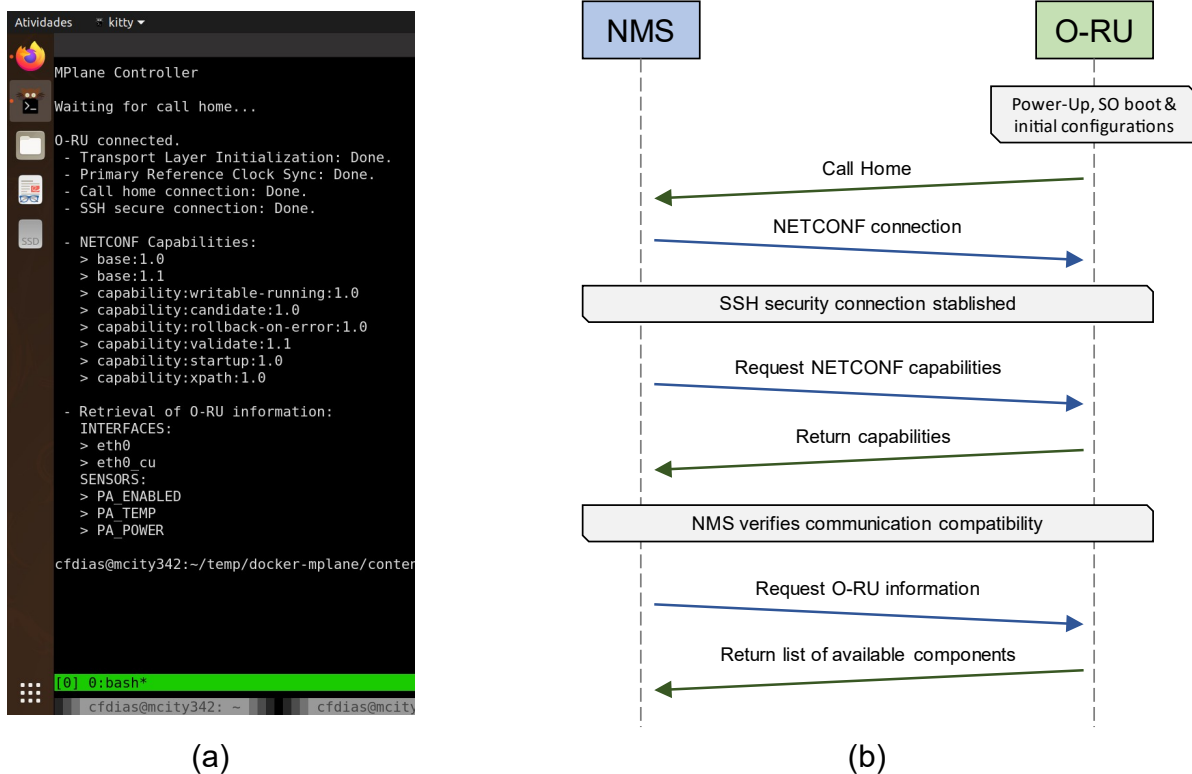


Figura 9 – Demonstração da função Call Home com as mensagens de texto geradas pelo M-Plane Controller (a) e o diagrama de sequência correspondente (b).

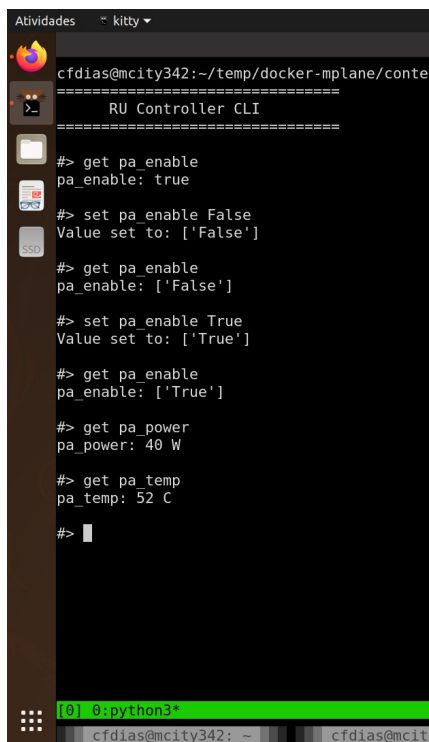
### 3.2 Monitoração e Controle de Parâmetros do PA

A Figura 10 (a) mostra os comandos inseridos via CLI com as respectivas respostas recebidas da O-RU, bem como o diagrama de sequência correspondente (b).

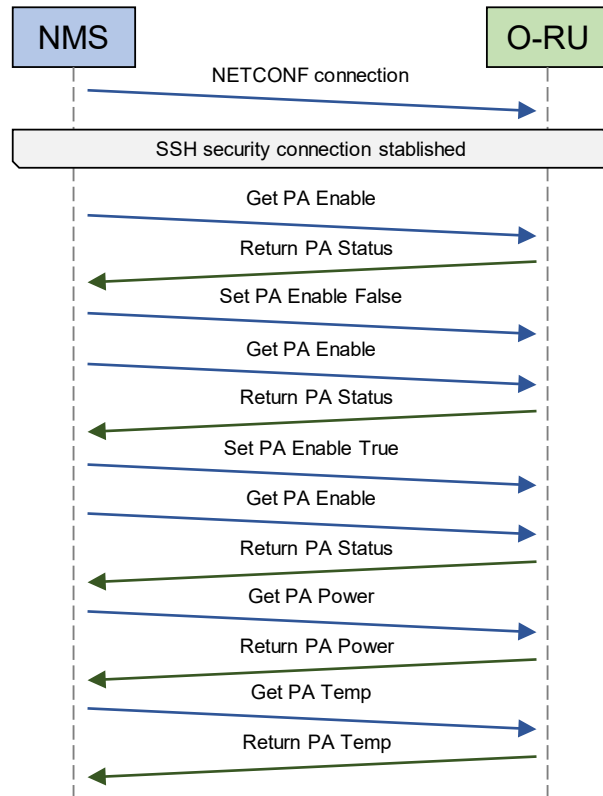
Nessa demonstração pode ser observado o seguinte conjunto de comandos:

- O NMS envia um comando *set* para ativar o PA;
- O NMS envia um comando *get* para ler o estado do PA;  
A O-RU responde com *true* ou *false*, indicando se o PA está ligado ou desligado;
- O NMS envia um comando *get* para ler a potência de RF na saída do PA;  
A O-RU responde com o valor correspondente em watt (W);
- O NMS envia um comando *get* para ler a temperatura do PA;  
A O-RU responde com o valor correspondente em graus Celsius.





(a)



(b)

Figura 10 – Demonstração das funções de monitoração e controle do PA através de comandos inseridos via CLI (a) e o diagrama de sequência correspondente (b).

### 3.3 Exemplos de Mensagens XML do Protocolo NETCONF

A Figura 11 mostra trechos de mensagens XML do protocolo NETCONF geradas pelo NMS (a) e pela O-RU (b) quando o comando `get pa_power` é inserido na CLI. Os destaques em negrito indicam o *namespace* do modelo YANG utilizado, o parâmetro PA\_POWER monitorado e o valor (40) correspondente sendo retornado.

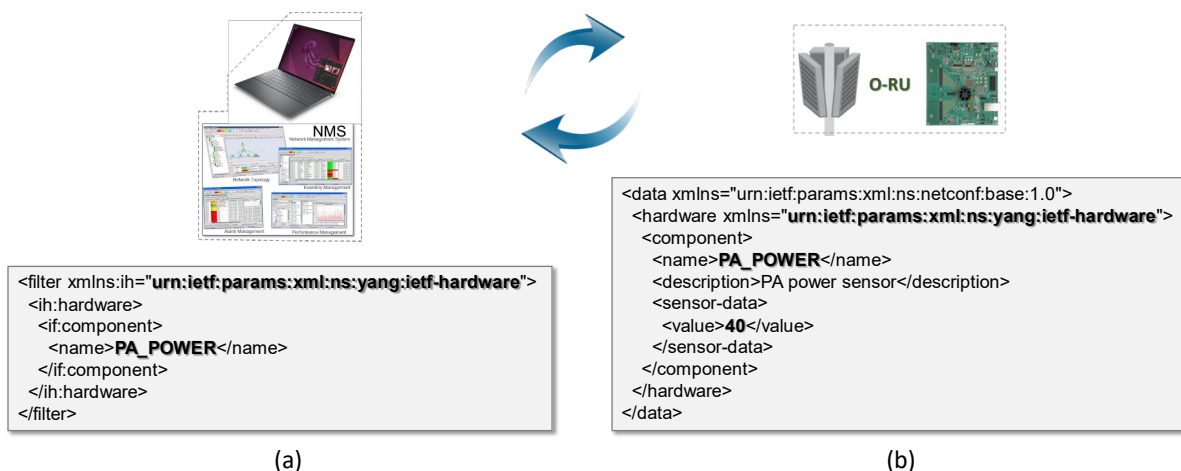


Figura 11 – Trechos de mensagens XML do protocolo NETCONF geradas pelo NMS (a) e pela O-RU (b) quando o comando `get pa_power` é inserido na CLI.

### 3.4 Processos Típicos Realizados em Uma Transação NETCONF

Os seguintes processos são tipicamente realizados durante uma transação NETCONF na qual o NMS envia o comando *get pa\_power*, por exemplo:

1. Uma mensagem XML é transmitida pelo NMS (cliente NETCONF) para a O-RU, contendo campos que referenciam um determinado modelo YANG e o parâmetro que está sendo solicitado (*PA Power*);
2. Esta mensagem é então interpretada pelo servidor NETCONF da O-RU (Netopeer2), que consulta uma base de dados (gerenciada pelo Sysrepo) e valida o seu conteúdo e sintaxe com base nos parâmetros estruturados no respetivo modelo YANG (armazenado em uma Biblioteca YANG);
3. Finalmente, um “YANG Handler” correspondente é invocado pelo Sysrepo (após solicitação do Netopeer2) para interagir com outros módulos de software que obtêm os dados solicitados do sistema operacional ou diretamente do HW associado.

Essa interação é ilustrada na Figura 12.

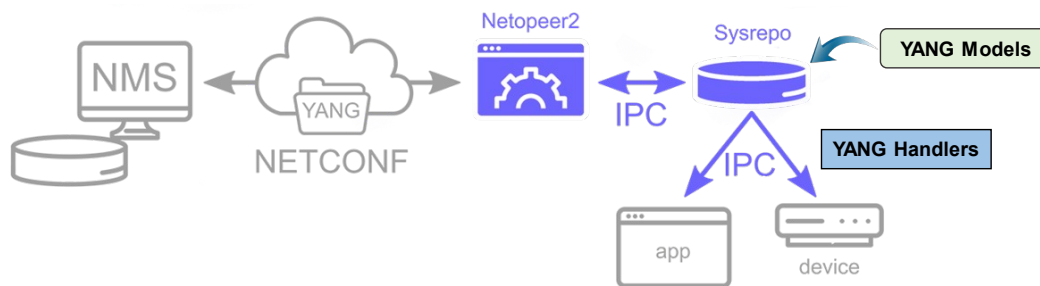


Figura 12 – Representação da interação entre o NMS e o servidor NETCONF (Netopeer2 + Sysrepo).

#### 4 Conclusões

O desenvolvimento e integração dos componentes de código que compõem a aplicação do Software de Gerenciamento da O-RU foram realizados com sucesso.

Testes realizados diretamente na plataforma de hardware contendo o RFSoc demonstraram o funcionamento do núcleo do Software de Gerenciamento, contendo:

- Servidor NETCONF (Netopeer2);
- Gerenciador de banco de dados de configuração (Sysrepo);
- Modelos YANG abertos (padronizados);
- YANG Handlers que acessam o sistema operacional e/ou o hardware para obter os parâmetros solicitados (neste momento de forma simulada).

A demonstração realizada pelo CPQD em dezembro/2023 evidencia o ganho de competência técnica da equipe e a preparação do ambiente (infraestrutura) de desenvolvimento, necessários para a continuidade do trabalho visando a implementação da versão final do software de gerenciamento da O-RU no escopo do projeto OpenRAN@Brasil Fase 2.



---

## 5 Histórico de Alterações do Documento Consolidado

<b>Data de emissão</b>	<b>Versão</b>	<b>Descrições das alterações realizadas</b>
04/Mar/24	AA	Versão inicial.

## 6 Execução e Aprovação

**Elaborado por:**

Eduardo Mobilon

Claudio Ferreira Dias

Fundação CPQD – Centro de Pesquisa e Desenvolvimento em Telecomunicações

**Revisado por:**

Eduardo Mobilon

Fundação CPQD – Centro de Pesquisa e Desenvolvimento em Telecomunicações

**Aprovado por:**

RNP – Rede Nacional de Ensino e Pesquisa