RNP Educação, Pesquisa e Inovação em Rede

Manual de Instalação de *Core* 5G

Instalando e Configurando Instância do Projeto Free5GC

Novembro de 2023

Coordenador do CT-GId
Emerson Ribeiro de Mello, Dr. (IFSC)

Assistente Técnica do CT-GId
Shirlei Aparecida de Chaves, Ma. (IFSC)

Coordenador da RNP para o CT-GId
Clayton Reis da Silva, Me. (RNP)

Diretor Adjunto de e-Ciência e Ciberinfraestrutura Avançada
Leandro Neumann Ciuffo, Me. (RNP)

Diretora de Pesquisa e Desenvolvimento
Iara Machado, Ma. (RNP)

Autores
Leonardo Azalim de Oliveira (UFJF)
Edelberto Franco Silva, Dr. (UFJF)

SUMÁRIO

Lista de abreviaturas e siglas

5G quinta geração de redes móveis.

5GC *5G Core*.

AMF *Access and Mobility Management Function*.

EAP *Extensible Authentication Protocol*.

GNB *gNodeB*.

GTP *GPRS Tunnelling Protocol*.

LAN *Local Area Network*.

LVM *Logical Volume Manager*.

N3IWF *Non-3GPP Inter Working Function*.

NF Função de Rede (*Network Function*).

PGId Programa de Gestão de Identidade.

RNP Rede Nacional de Pesquisa.

SO Sistema Operacional.

UE *User Equipment*.

UPF *User Plane Function*.

Lista de Figuras

1 Introdução

O presente documento foi produzido no âmbito do projeto intitulado “Estudo e Avaliação de Métodos de Autenticação EAP na Infraestrutura de Redes de Telecomunicação 5G” que faz parte do grupo de propostas selecionadas no Programa de Gestão de Identidade (PGId) 2023.

O documento detalha os aspectos técnicos do ambiente de testes que foi construído utilizando um estilo de escrita semelhante ao de um guia e com o objetivo de melhorar a reprodutibilidade dos resultados obtidos e do ambiente utilizado no estudo. O conteúdo deste foi colocado de forma separada pois não condizia com o presente nos demais documentos produzidos por este projeto e que podem ser acessados na Wiki[1] da Rede Nacional de Pesquisa (RNP).

Este manual assume que o leitor, além de alguma familiaridade com os componentes do 5GC, possui experiência com redes de computadores e com a administração de sistemas Linux, e que os sistemas operacionais utilizados no ambiente são iguais aos mencionados na Subseção 2.2, ou que ao menos sejam do grupo de sistemas “Debian-based”[2].

2 Configuração Inicial do Ambiente

Esta seção contém as instruções para construção do ambiente de testes juntamente com outras informações gerais.

2.1 Requisitos para Criação das Máquinas Virtuais

O TrueNAS[3] foi escolhido como software de virtualização para instalação das máquinas virtuais por conta de sua maior flexibilidade e controle quando comparado com soluções como o Oracle VM VirtualBox[4] e o VMWare Workstation[5].

Os softwares citados, bem como os demais disponíveis no mercado podem ser utilizados para a construção do ambiente, pois a forma como as máquinas são executadas é independente do ponto de vista do escopo coberto por este manual.

O principal requisito que deve ser observado durante a escolha da solução de virtualização é a possibilidade de acesso direto à rede do sistema hospedeiro (que geralmente é realizado via *bridge*[6]).

[1] https://wiki.rnp.br/display/comitetgi/PGId+2023+-+Acompanhamento+e+Resultados
[2] https://en.wikipedia.org/wiki/List_of_Linux_distributions#Debian-based
[3] https://www.truenas.com/
[4] https://www.virtualbox.org/
[5] https://www.vmware.com/br/products/workstation-pro.html
[6] https://www.virtualbox.org/manual/ch06.html#network_bridged

2.2 Descrição do Ambiente Virtual Testado

Esta subseção detalha os requisitos de *software* para correta implantação do ambiente de teste.

2.2.1 *Hardware* e Sistema Operacional (SO)

- Máquina do 5GC (requisitos mínimos)
  - 2GB RAM
  - 1 *core* de CPU
  - 20GB de HDD
  - Sistema Operacional (SO) Linux Ubuntu Server 20.04.6
- Máquina do 5GC (requisitos recomendados)[7]
  - 3GB RAM ou mais
  - 2 *cores* de CPU ou mais
  - 160GB de HDD
  - SO Linux Ubuntu Server 20.04.6

A versão do Ubuntu Server foi escolhida em virtude da presença do kernel Linux versão 5.4.x, que é uma dependência do módulo de *GPRS Tunnelling Protocol* (GTP)[8] que deve estar instalado no mesmo ambiente que o 5GC.

- Máquina do UERANSIM (requisitos mínimos)
  - 1GB RAM
  - 1 *core* de CPU
  - 10GB de HDD
  - Sistema Operacional (SO) Linux Ubuntu Server 20.04.6
- Máquina do UERANSIM (requisitos recomendados)
  - 2GB RAM
  - 2 *cores* de CPU
  - 20GB de HDD
  - SO Linux Ubuntu Server 22.04.3

O Ubuntu Linux que foi selecionado como requisito para esta outra máquina virtual é igual a da VM do 5GC somente por que torna mais fácil a criação da mesma a partir de uma clonagem de máquinas, mas é possível, neste caso, utilizar também a versão 22.04.

2.2.2 *Software* / Aplicações

Por conta da complexidade do ambiente, e com o objetivo de documentar os demais *softwares* e componentes utilizados em sua construção, as Tabelas 1 e 2 apresentam uma listagem dos *softwares* e das respectivas versões testadas.

[7] Caso o UPF seja instanciado na mesma máquina que o 5GC e as demais Funções de Rede (NFs, *Network Function*), utilizar esta configuração

[8] https://en.wikipedia.org/wiki/GPRS_Tunnelling_Protocol

Tabela 1: Mapeamento das NFs do 5GC e respectivas versões utilizadas.

NF	Commit	Versão
AMF	7907d3c0	1.0.9
AUSF	64f47ebe	1.0.3
N3IWF	22988ee2	1.0.5
NRF	db4c0f90	1.0.2
NSSF	28cd7936	1.0.2
PCF	17f2a8fc	1.0.2
SMF	8eb6843b	1.0.7
UDM	f9aad0ef	1.0.3
UDR	a8ef9d9f	1.0.2
UPF	4474dc86	1.0.3

Tabela 2: Listagem das versões dos demais *softwares* utilizados.

Software	Versão
MongoDB	3.6.8
Go	1.17.8
GTP-U	v0.8.3
UERANSIM	v3.2.6
CMake	3.28.0

2.3 Ambiente Virtual

Para correta execução do ambiente de testes, serão necessárias ao menos 2 máquinas virtuais. Como sugerido na documentação do Free5GC, é possível realizar a configuração das máquinas virtuais a partir de uma máquina "base".

De acordo com a documentação[9], durante o processo de instalação do SO dessa máquina, é recomendado habilitar o servidor OpenSSH para simplificar o acesso remoto ao sistema e *não* habilitar *Logical Volume Manager* (LVM)[10] no disco da máquina para facilitar futuras modificações no tamanho e disposição de suas partições.

Após terminar de instalar uma máquina virtual[11] com Ubuntu Server, criar uma cópia desta máquina e seguir as instruções contidas na Subseção 2.4.

2.4 Configuração dos Pré-requisitos

Esta subseção contém as instruções para construção do ambiente de testes e, idealmente, devem ser executadas somente uma vez.

2.4.1 Etapas iniciais

- Atualizar os pacotes do sistema

```
sudo apt update && sudo apt upgrade -y
```

- Verificar a versão do kernel

```
uname -r
```

NOTA: Por conta do *patch* aplicado pelo GTP necessário para o 5GC funcionar é necessário instalar um kernel da versão 5.4.x.

2.4.2 Suporte à Go Lang

- Verificar a versão instalada

```
go version
```

Saída exemplo:

```
go version go1.18.10 linux/amd64
```

- Instalar o Go 1.18.10

Caso uma outra versão de Go já tenha sido instalada no passado, executar:

```
# this assumes your current version of Go is in the default location:
sudo rm -rf /usr/local/go
wget https://dl.google.com/go/go1.18.10.linux-amd64.tar.gz
sudo tar -C /usr/local -zxvf go1.18.10.linux-amd64.tar.gz
```

[9] https://free5gc.org/guide/1-vm-en/#3-create-a-ubuntu-server-vm
[10] https://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)
[11] https://free5gc.org/guide/1-vm-en/

Se não, basta executar esses outros comandos abaixo:

```
wget https://dl.google.com/go/go1.18.10.linux-amd64.tar.gz
sudo tar -C /usr/local -zxvf go1.18.10.linux-amd64.tar.gz
mkdir -p ~/go/{bin,pkg,src}
# The following assume that your shell is bash:
echo 'export GOPATH=$HOME/go' >> ~/.bashrc
echo 'export GOROOT=/usr/local/go' >> ~/.bashrc
echo 'export PATH=$PATH:$GOPATH/bin:$GOROOT/bin' >> ~/.bashrc
echo 'export GO111MODULE=auto' >> ~/.bashrc
source ~/.bashrc
```

2.4.3 Banco de Dados

- Se for usar o mongoDB 5.x ou superior, verificar se as instruções AVX são suportadas pela CPU onde a máquina está rodando, pois estas são requisito para correta execução

```
grep --color avx /proc/cpuinfo
```

Um exemplo de saída para o comando acima pode ser visto na Figura 1.

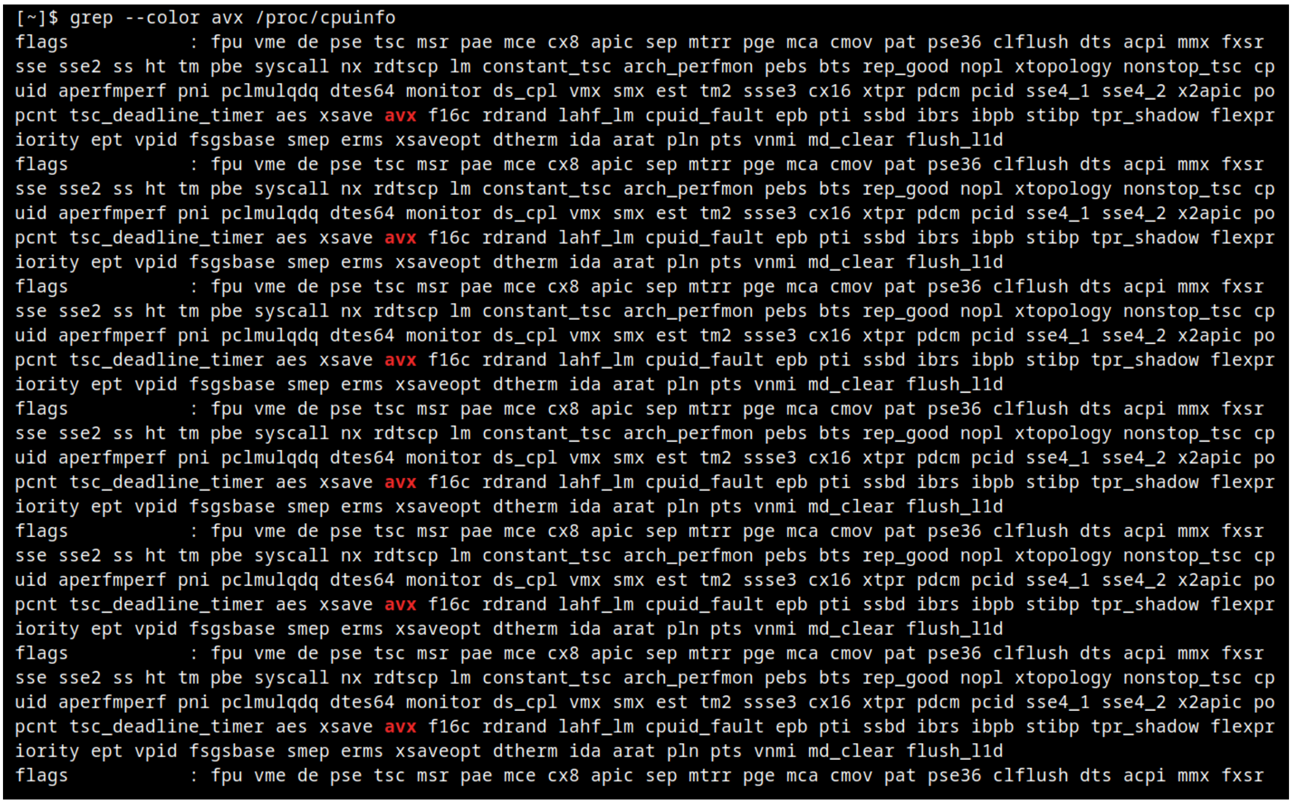


Figura 1: Exemplo de saída para verificar a presença da instrução AVX na CPU.

Se houver suporte, é possível instalar o mongoDB 5.x, senão usar a versão 4.4.x. O comando abaixo instala por padrão no Ubuntu 20.04 a versão 3.x

```
sudo apt install mongodb wget git
```

- Iniciar o serviço do banco de dados

```
sudo systemctl start mongodb
```

- Verificar se o serviço do banco de dados está ativo

```
systemctl status mongodb
```

NOTA: O comando acima deve retornar "active (running)"

- [Opcional] Para iniciar o serviço do banco de dados quando o SO iniciar

```
sudo systemctl enable mongodb
```

2.4.4 Configurar os pré-requisitos do UPF

- Instalar os pré-requisitos para executar o UPF

```
sudo apt -y install git gcc g++ cmake autoconf libtool pkg-config libmnl-dev libyaml-dev
```

- Habilitar o encaminhamento de pacotes e configura o *firewall* do servidor

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o <dn_interface> -j MASQUERADE
sudo iptables -A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1400
sudo systemctl stop ufw
sudo systemctl disable ufw
```

No caso do Ubuntu Server 20.04, os comandos ficariam da seguinte forma:

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o enp0s4 -j MASQUERADE
sudo iptables -A FORWARD -p tcp -m tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1400
sudo systemctl stop ufw
sudo systemctl disable ufw
```

2.4.5 Configurar o Control Plane

- Clonar o repositório

```
git clone --recursive -b v3.3.0 -j `nproc` https://github.com/free5gc/free5gc.git
cd free5gc
```

NOTA: A versão mais recente disponível no momento de escrita era a 3.3.0, verificar se há uma nova versão no repositório oficial[12]

- Compilar as NFs

Para compilar todas as NFs:

```
cd ~/free5gc
make
```

Para compilar uma NF específica:

```
cd ~/free5gc
make <NF>
```

OBS: Substituir <NF> pelo nome em caixa baixa do módulo a ser compilado (ex: upf)

[12] https://github.com/free5gc/free5gc

2.5 Instalar o UPF

Começar instalando o módulo personalizado do kernel

```
cd ~/free5gc
git clone -b v0.8.3 https://github.com/free5gc/gtp5g.git
cd gtp5g
make
sudo make install
```

NOTA: A versão mais recente disponível no momento de escrita era a 0.8.3, verificar se há uma nova versão no repositório oficial[13]

OBS: Neste tutorial será mantida a configuração padrão, mas se necessário, personalize o UPF como desejado. O arquivo de configuração invocado pelo script `run.sh` está localizado em `free5gc/config/upfcfg.yaml`

- Instalar o WebConsole

Começar instalando as dependências:

```
sudo apt remove cmdtest yarn
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
sudo apt update
sudo apt install -y ca-certificates curl gnupg # dependencias do nodejs
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o \
/etc/apt/keyrings/nodesource.gpg
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_20.x
    nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list
sudo apt update
sudo apt install -y nodejs yarn
```

Compilar o webconsole[14] usando:

```
cd ~/free5gc
make webconsole
```

Para executar o Webconsole, olhar a Subseção 2.6.2.

2.6 Executando o Free5GC

2.6.1 Execução do 5GC

Para instanciar o 5GC Free5GC[15]

```
cd ~/free5gc/
./run.sh
```

2.6.2 Execução do Webconsole

Para que o módulo funcione corretamente, ele deve ter tido seus pré-requisitos instalados de acordo com as instruções da Subseção 2.5 e a instância do 5GC deve estar sendo executada de acordo com as instruções contidas na Subseção 2.6.1.

[13] https://github.com/free5gc/gtp5g

[14] Mesmo ao executar o comando *make* para compilar todas as NF, o Webconsole não é compilado

[15] Caso seja necessário, dar permissão de execução para o *script* usando o comando `chmod +x run.sh`

- Executar o servidor do módulo Webconsole

Na máquina onde está o *core*:

```
cd ~/free5gc/webconsole
./bin/webconsole
```

Após o servidor estar em execução, a seguinte mensagem deve ser exibida no terminal:

```
[GIN-debug] Listening and serving HTTP on :5000
```

2.7 Instalação do UERANSIM

Aproveitando a máquina virtual que foi criada a partir das instruções da Subseção 2.3, criar um clone dela[16] e então executar

```
sudo apt update && sudo apt upgrade -y
```

- Instalar as dependências

```
sudo apt install make g++ libsctp-dev lksctp-tools iproute2
sudo snap install cmake --classic
```

- Clonar o repositório

```
cd ~
git clone https://github.com/aligungr/UERANSIM
cd UERANSIM
```

E retornar a um *commit* específico

```
git checkout 3a96298
```

Caso ainda não tenha sido feito, executar na máquina do 5GC as instruções para instalação do Webconsole que se encontram na Subseção 2.5. Esse console será útil para a configuração do *User Equipment* (UE) virtual.

- Compilar o UERANSIM

```
cd ~/UERANSIM
make
```

NOTA: O processo de compilação demorou em torno de 5 minutos na máquina com a configuração descrita como recomendada que foi utilizada nos testes

O processo deve finalizar com a seguinte mensagem: "`UERANSIM successfully built.`"

2.8 Criação de um UE virtual no Free5GC

Caso ainda não esteja, executar o Webconsole de acordo com as instruções da Subseção 2.6.2.

2.8.1 Adição de um novo UE

- Para adicionar um novo UE no core

[16] Lembrar de alterar o *hostname* e o IP desta nova máquina para não conflitar com a máquina do 5GC

Abrir um *browser* e navegar até o endereço `http://<IP>:5000`, onde IP é o endereço de IP da VM do *core*.

Exemplo: `http://172.16.0.114:5000`

As credenciais de acesso ao Webconsole são `admin:free5gc`

Na página inicial, navegar pela barra lateral esquerda até a opção “Subscribers”. Clicar nela e depois, na nova página que abrir, no botão “New Subscriber”. No *pop-up* que abrir, haverão diversos parâmetros de configuração. Rolar a tela até atingir o parâmetro “Operator Code Type” e alterá-lo para “OP”.

Neste tutorial os demais parâmetros ficarão com seus valores padrão. Clicar no botão “Submit” para salvar os dados do novo UE.

NOTA: Neste ponto já é possível sair do *browser* e fechar o processo do servidor do Webconsole usando `Ctrl+C`

2.9 Conexão do UERANSIM com o Free5GC

Na máquina do *core*, alterar os IPs que se econtram dentro de 3 arquivos de configuração mudando dos IPs de *loopback* para o da *Local Area Network* (LAN). Inicialmente, editar o arquivo `amfcfg.yaml`.

```
cd ~/free5gc
nano config/amfcfg.yaml
```

Na linha

```
ngapIpList:  # the IP list of N2 interfaces on this AMF
  - 127.0.0.18
```

Alterar para

```
ngapIpList:  # the IP list of N2 interfaces on this AMF
  - <IP>
```

Caso o IP da máquina do 5GC fosse `172.16.0.114`, a linha ficaria da seguinte forma:

```
ngapIpList:  # the IP list of N2 interfaces on this AMF
  - 172.16.0.114
```

O próximo arquivo a ser alterado é `smfcfg.yaml`

```
nano config/smfcfg.yaml
```

Navegar pelas seções `userplaneInformation / upNodes / UPF / interfaces / endpoints` de dentro do arquivo até encontrar:

```
interfaces: # Interface list for this UPF
   - interfaceType: N3 # the type of the interface (N3 or N9)
     endpoints: # the IP address of this N3/N9 interface on this UPF
       - 127.0.0.8
```

E alterar o IP. Considerando o exemplo, o arquivo ficaria da seguinte forma:

```
interfaces: # Interface list for this UPF
   - interfaceType: N3 # the type of the interface (N3 or N9)
     endpoints: # the IP address of this N3/N9 interface on this UPF
       - 172.16.0.114
```

Por fim, no arquivo `upfcfg.yaml`

```
nano config/upfcfg.yaml
```

Alterar o IP em:

```
gtpu:
    forwarder: gtp5g
    # The IP list of the N3/N9 interfaces on this UPF
    # If there are multiple connection, set addr to 0.0.0.0 or list all the addresses
    ifList:
        - addr: 127.0.0.8
          type: N3
```

Para, de acordo com o ambiente de exemplo:

```
gtpu:
    forwarder: gtp5g
    # The IP list of the N3/N9 interfaces on this UPF
    # If there are multiple connection, set addr to 0.0.0.0 or list all the addresses
    ifList:
        - addr: 172.16.0.114
          type: N3
```

Agora, na máquina do UERANSIM, devemos apontar onde o *core* se encontra na rede.

Existem dois arquivos de configuração no UERANSIM que são relacionados com o Free5GC:

- `~/UERANSIM/config/free5gc-gnb.yaml`
- `~/UERANSIM/config/free5gc-ue.yaml`

Configurar tanto o IP da LAN no UERANSIM, quanto apontar qual o IP da *Access and Mobility Management Function* (AMF) do 5GC

```
nano ~/UERANSIM/config/free5gc-gnb.yaml
```

Nas linhas que possuem:

```
ngapIp: 127.0.0.1   # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 127.0.0.1    # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
    - address: 127.0.0.1
```

Colocar nas duas primeiras linhas o IP do UERANSIM, já na outra o IP da máquina do 5GC. Abaixo segue um exemplo com os IPs fictícios:

```
ngapIp: 172.16.0.118 # gNB's local IP address for N2 Interface (Usually same with local IP)
gtpIp: 172.16.0.118 # gNB's local IP address for N3 Interface (Usually same with local IP)

# List of AMF address information
amfConfigs:
  - address: 172.16.0.114
```

Abrir o arquivo do UE:

```
nano ~/UERANSIM/config/free5gc-ue.yaml
```

Alterar o parâmetro:

```
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OPC'
```

Para:

```
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OP'
```

E checar se os demais parâmetros do arquivo conferem com aqueles que aparecem no Webconsole[17]. Então verificar se é necessário executar os comandos da subseção 2.9.1.

2.9.1 Reaplicar Configurações ná Máquina do Free5GC após *Reboot*

Toda vez que a máquina do 5GC reiniciar, executar os comandos de configuração do *firewall* e habilitar o encaminhamento de pacotes no kernel

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -t nat -A POSTROUTING -o enp0s4 -j MASQUERADE
sudo iptables -I FORWARD 1 -j ACCEPT
```

Lembrar manter o serviço de banco de dados em execução por meio do comando abaixo

```
sudo systemctl start mongodb
```

2.10 Teste de Conexão do UE virtual com o Free5GC

Na máquina do 5GC, instanciar o *core*:

```
cd ~/free5gc
./run.sh
```

Agora serão necessárias 3 conexões SSH (ou 3 abas de terminal) na máquina do UERANSIM:

Na primeira, instanciar um *gNodeB* (gNB):

```
cd ~/UERANSIM
build/nr-gnb -c config/free5gc-gnb.yaml
```

Na segunda, instanciar um UE:

```
cd ~/UERANSIM
sudo build/nr-ue -c config/free5gc-ue.yaml # for multiple-UEs, use -n and -t for number and delay
```

Na terceira, verificar se a interface de conexão foi criada com sucesso:

```
ip addr show
```

O comando acima deve retornar algo como o que é mostrado no exemplo da Figura 2. Note que a interface `uesimtun0` é aquela que foi criada pelo UE. O próximo passo é testar a conexão do UE com a internet usando, por exemplo, o comando abaixo:

```
ping -I uesimtun0 google.com
```

Uma saída de exemplo pode ser vista na Figura 3.

[17] Caso o UE tenha sido criado com a configuração padrão, provavelmente os parâmetros já se encontram corretos

```
netlab@ueransim:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:a0:98:34:4f:4a brd ff:ff:ff:ff:ff:ff
    inet 172.16.0.118/16 metric 100 brd 172.16.255.255 scope global dynamic enp0
s4
       valid_lft 5426sec preferred_lft 5426sec
    inet6 fe80::2a0:98ff:fe34:4f4a/64 scope link
       valid_lft forever preferred_lft forever
3: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 qdisc fq_cod
el state UNKNOWN group default qlen 500
    link/none
    inet 10.60.0.1/32 scope global uesimtun0
       valid_lft forever preferred_lft forever
    inet6 fe80::2c5f:7ba1:123b:7ac/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Figura 2: Lista de interfaces disponíveis após conexão do UE com o 5GC.

```
3: uesimtun0: <POINTOPOINT,PROMISC,NOTRAILERS,UP,LOWER_UP> mtu 1400 qdisc fq_cod
el state UNKNOWN group default qlen 500
    link/none
    inet 10.60.0.1/32 scope global uesimtun0
       valid_lft forever preferred_lft forever
    inet6 fe80::2c5f:7ba1:123b:7ac/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
netlab@ueransim:~$ ping -I uesimtun0 google.com
PING google.com (142.251.132.238) from 10.60.0.1 uesimtun0: 56(84) bytes of data
.
64 bytes from gru14s46-in-f14.1e100.net (142.251.132.238): icmp_seq=1 ttl=50 tim
e=25.6 ms
64 bytes from gru14s46-in-f14.1e100.net (142.251.132.238): icmp_seq=2 ttl=50 tim
e=25.5 ms
64 bytes from gru14s46-in-f14.1e100.net (142.251.132.238): icmp_seq=3 ttl=50 tim
e=26.8 ms
64 bytes from gru14s46-in-f14.1e100.net (142.251.132.238): icmp_seq=4 ttl=50 tim
e=25.7 ms
64 bytes from gru14s46-in-f14.1e100.net (142.251.132.238): icmp_seq=5 ttl=50 tim
e=25.7 ms
^C
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 6634ms
rtt min/avg/max/mdev = 25.535/25.891/26.842/0.480 ms
```

Figura 3: Exemplo de saída para o teste de conexão do UE com a internet.

3 *Non-3GPP Inter Working Function* (N3IWF)

3.1 Testando o Módulo

O próprio *core* free5GC possui um *script* chamado `test.sh` que executa alguns testes básicos sendo que, nesse caso, será testado o *Non-3GPP Inter Working Function* (N3IWF). Para isso, na máquina do core executar:

```
cd ~/free5gc/
./test.sh TestNon3GPP
```

NOTA: Caso o teste falhe ou seja interrompido antes de finalizar, executar o comando abaixo de dentro da pasta `free5gc` para reestabelecer o funcionamento do 5GC

```
cp config/amfcfg.conf.bak config/amfcfg.conf
```

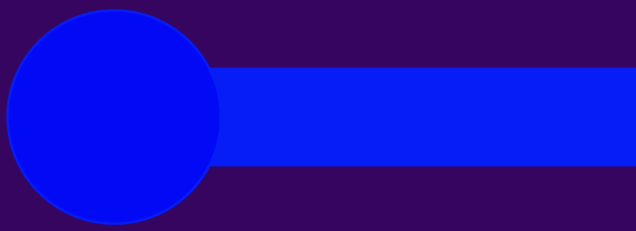
Exemplo de saída[18]:

```
[...]
--- PASS: TestNon3GPPUE (9.66s)
PASS
ok    test  9.675s
2023-10-03T13:24:00.029537567Z [INFO][UPF][Main] Shutdown UPF ...
2023-10-03T13:24:00.029600330Z [INFO][UPF][PFCP][LAddr:10.200.200.101:8805]
Stopping pfcp server
[...]
2023-10-03T13:24:00.308471602Z [INFO][N3IWF][Init] N3IWF terminated
[...]
--- PASS: TestCN (26.39s)
PASS
ok    test  28.033s
```

Para executar o 5GC carregando o N3IWF, usar o comando:

```
cd ~/free5gc
./run.sh -n3iwf
```

[18] Por questões de visualização, algumas partes do texto foram omitidas



RNP

MINISTÉRIO DA **CULTURA**

MINISTÉRIO DA **DEFESA**

MINISTÉRIO DA **SAÚDE**

MINISTÉRIO DAS **COMUNICAÇÕES**

MINISTÉRIO DA **EDUCAÇÃO**

MINISTÉRIO DA **CIÊNCIA, TECNOLOGIA E INOVAÇÃO**

GOVERNO FEDERAL

BRASIL

UNIÃO E RECONSTRUÇÃO