

Universidade de Aveiro

Comunicações Móveis

Projeto 1 5G

Open5GS



João Torrinhas(98435), Diogo Torrinhas(98440), Diogo Almeida (89186)

Departamento de Electrónica, Telecomunicações e Informática

6 de Janeiro de 2023



Índice

Índice	1
Glossário	2
Resumo	3
Introdução	3
Objetivos.....	3
Metodologia	3
Implementação.....	3
Open5GS.....	3
UERANSIM	4
Configuração.....	4
Problemas	4
Problemas na instalação do mongoDB.....	4
WebUI não estava a realizar o login, não conseguimos ter acesso através do localhost.....	5
Conseguimos fazer <i>pings</i> para a internet, mas não entre os UE's.....	6
Conclusão	9
Bibliografia	10

Glossário

5G Five Generation of Wireless Communications

5GC Core 5G

AMF Access and Mobility Management Function

AUFS Authentication Server Function

eNB Envolved NodeB

EPC Evolved Packet Core (4GC)

gNB Next Generation NodeB

LTE Long Term Evolution

NR New Radio

NSSF Network Slice Selection Function

PCF Policy Control Function

RAN Radio Access Network

SMF Session Management Function

UDM Unified Data Management

UDR Unified Data Repository

UE User Equipament

UPF User Plane Function

USIM Universal Subscriber Identity Module

VM Virtual Machine

Resumo

Ao longo deste projeto simulamos uma rede 5G com ajuda do Open5GS e o UERANSIM, inicialmente criámos duas VM's (para Core 5G e para UE+gNB), mas devido a um grande número de erros optámos por seguir outros métodos mais complexos, nomeadamente usar mais VM's.

Introdução

Objetivos

- Instanciar e usar uma rede 5G completa usando o Open5GS.

Metodologia

- Instalar e iniciar o Open5GC como uma solução para o Core 5G;
- Instalar e usar o UERANSIM para o acesso 5G;
- Alterar os arquivos necessários para a configuração do Core 5G, gNB e UE;
- Iniciar os gNB's e os UE's;
- Observar o funcionamento da rede 5G com o *Wireshark*.

Implementação

Open5GS

O open5GS é a implementação de um 5GC ou EPC *open source*, ou seja é a implementação de uma rede NR ou LTE realizada em linguagem C. Para criar uma rede própria precisamos de um gNB ou eNB e de um USIM. A WebUI é uma interface de utilizador fornecida para a realização de testes e é implementado em Node.JS[3]. A figura seguinte apresenta o esquema simples da arquitetura 5G tendo como *core* o Open5GS.

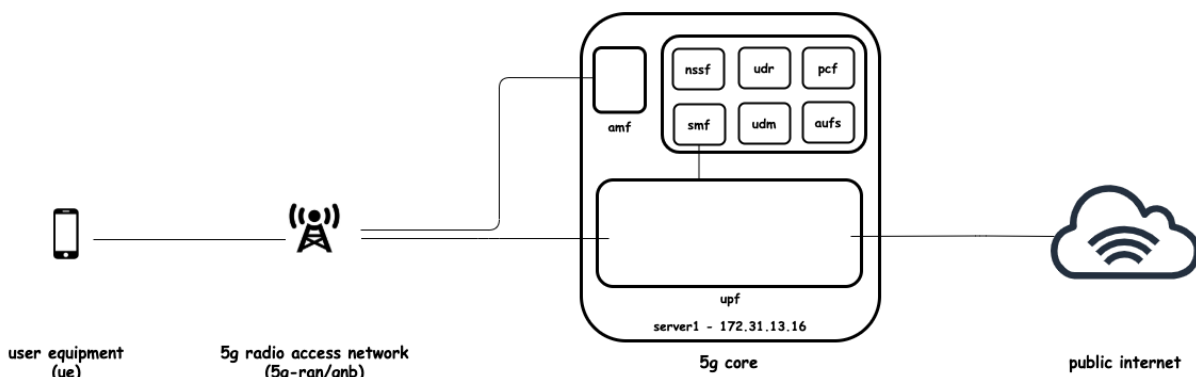


Figura 1

UERANSIM

UERANSIM é um simulador de última geração *open source* que implementa UE's 5G e RAN's [4]. No nosso caso podemos considerar que o UE é um Smartphone 5G e o RAN é um gNodeB. O UERANSIM em junção com o Open5GS, pode ser usados para testar e estudar uma rede 5G.

Configuração

Para simular o funcionamento da rede NR criamos 2 VM's uma a correr o URANSIM (server 2) e outra a correr o Core 5G (server 1), como mostra na figura seguinte.

Instalamos o Open5GS, o mongoDB e o Node.JS (usado no WebUI) na primeira VM e o UERANSIM na segunda VM seguindo a proposta de projeto "Deploying 5G Core Network with Open5GS and UERANSIM"[5].

A configuração das duas VM's está disponível no ficheiro *commands.txt*.

Depois de realizada a instalação alteramos os ficheiros de configuração do Core 5G, do gNB e do UE com os devidos ip's das máquinas criadas, que também está disponível para consulta.

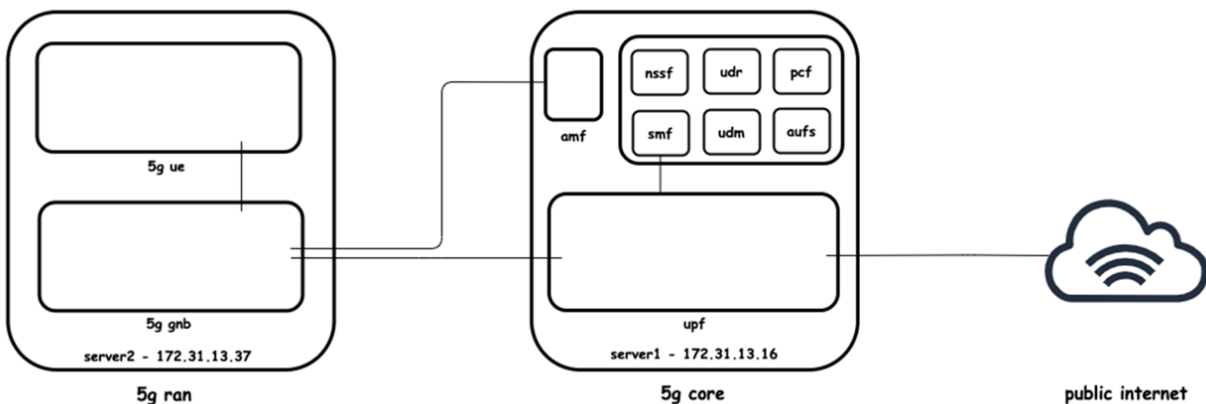


Figura 2

Problemas

Ao longo do trabalho fomos obtendo vários problemas.

Problemas na instalação do mongoDB.

O primeiro foi na instalação do mongoDB pois, depois da instalação o serviço do mongoDB não ficava ativo mesmo parando-o e ativando-o novamente. Posto isto tentámos desinstalá-lo e instalá-lo novamente mas sem sucesso porque estava a dar os erros presentes na figura abaixo. Só depois da ajuda do professor Daniel Corujo é que conseguimos com sucesso resolver o problema usando os comandos fornecidos pelo mesmo.

```
- sudo dpkg -i --force-overwrite /var/cache/apt/archives/mongodb-org-mongos_6.0.3_amd64.deb  
-sudo apt -f install
```

```
diogo@DiogoT-PC: ~  
Get:2 https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/6.0/multiverse amd64  
mongodb-org-mongos amd64 6.0.3 [20,3 MB]  
Fetched 49,2 MB in 2s (24,8 MB/s)  
(Reading database ... 303089 files and directories currently installed.)  
Preparing to unpack .../mongodb-org-server_6.0.3_amd64.deb ...  
Unpacking mongodb-org-server (6.0.3) ...  
dpkg: error processing archive /var/cache/apt/archives/mongodb-org-server_6.0.3_  
amd64.deb (--unpack):  
trying to overwrite '/usr/bin/mongod', which is also in package mongodb-server-  
core 1:3.6.9+really3.6.8+90-g8e540c0b6d-0ubuntu5.3  
dpkg-deb: error: paste subprocess was killed by signal (Broken pipe)  
Preparing to unpack .../mongodb-org-mongos_6.0.3_amd64.deb ...  
Unpacking mongodb-org-mongos (6.0.3) ...  
dpkg: error processing archive /var/cache/apt/archives/mongodb-org-mongos_6.0.3_  
amd64.deb (--unpack):  
trying to overwrite '/usr/bin/mongos', which is also in package mongodb-server-  
core 1:3.6.9+really3.6.8+90-g8e540c0b6d-0ubuntu5.3  
dpkg-deb: error: paste subprocess was killed by signal (Broken pipe)  
Errors were encountered while processing:  
/var/cache/apt/archives/mongodb-org-server_6.0.3_amd64.deb  
/var/cache/apt/archives/mongodb-org-mongos_6.0.3_amd64.deb  
E: Sub-process /usr/bin/dpkg returned an error code (1)  
diogo@DiogoT-PC:~$  
  
diogo@DiogoT-PC:~$ sudo apt-get install -y mongodb-org  
[sudo] password for diogo:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
mongodb-org is already the newest version (6.0.3).  
You might want to run 'apt --fix-broken install' to correct these.  
The following packages have unmet dependencies:  
mongodb-org-database : Depends: mongodb-org-server but it is not going to be in  
stalled  
Depends: mongodb-org-mongos but it is not going to be in  
stalled  
E: Unmet dependencies. Try 'apt --fix-broken install' with no packages (or speci  
fy a solution).  
diogo@DiogoT-PC:~$
```

Figura 3

WebUI não estava a realizar o login, não conseguimos ter acesso através do localhost.

Outro problema que tivemos foi no WebUI. Depois de feita a instalação do open5gs e do nodejs, não estava a ser possível ligar aceder à página de login através do <http://localhost:3000> como se pode ver na seguinte figura. Este problema persistiu e só criando outra VM é que foi possível resolvê-lo.

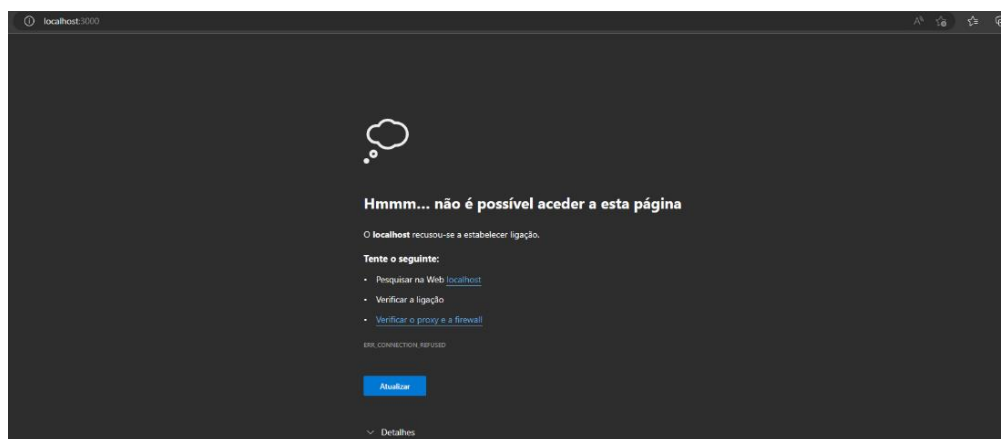


Figura 4

Conseguimos fazer pings para a internet, mas não entre os UE's.

Usando as 2 VM's, depois de todas as configurações feitas, verificámos que inicialmente não era possível fazer pings nem para a internet nem entre os UE's. Mais tarde descobrimos que o problema poderia estar na ligação entre as duas VM's, pois antes tínhamos Host only Adapter e mudámos para Bridge Adapter. Esta solução de facto funcionou, parcialmente uma vez que só era possível fazer pings para a internet, não foi possível ter conectividade entre UE's e por isso parte do problema persistiu. Como é possível ver pelas figuras 7,8 e 9, o pacote era enviado de uma UE para a UPF e a UPF enviava o pacote para o outro UE só que quando o pacote chegava à UE, ela voltava a enviar um request em vez de um reply, com um novo ID, ou seja, um novo ping que não correspondia ao ping anterior. Posto isto nós fomos analisar esses mesmos pacotes em questão e reparámos que também tinham o mesmo TID o que fez com que nós não conseguíssemos entender o porquê de não funcionar o ping entre eles porque as configurações estavam corretas.

Em seguida, decidimos adotar outra solução e então testámos usar 3VM's, 1 para o 5G Core, uma para o gNB e outra para ambos os UE's. Depois de feitas configurações bastante semelhantes às anteriores conferimos que esta solução tinha exatamente o mesmo problema anterior, ou seja, os UE's continuavam a não comunicar entre si e o ping para a internet era bem sucedido. Com as mesmas 3 VM's tentamos alterar a ligação entre a UPF e os UE's ou seja, em vez de ser OGSTUN para os 2 UE's, um tinha a ligação OGSTUN e outro a ligação OGSTUN2, ou seja, duas interfaces diferentes.

Face a estes resultados, decidimos aplicar a solução dada pelo professor Francisco Fontes e usar 4VM's, 1 para o 5G Core, 1 para o gNB, 1 para um dos UE's, e a última para o outro UE. Verificamos que fazia o mesmo que nas implementações anterior, ou seja, as UE's comunicavam para o core(UPF) e ambas comunicavam para a internet, mas não entre elas.

Por fim, numa última tentativa tentamos usar 5 VM's, com a distribuição das VM's semelhante à Figura 5, mas mais uma vez não conseguimos ter sucesso[6], porque após inicializar as UE's, as TUN Interfaces não eram criadas.

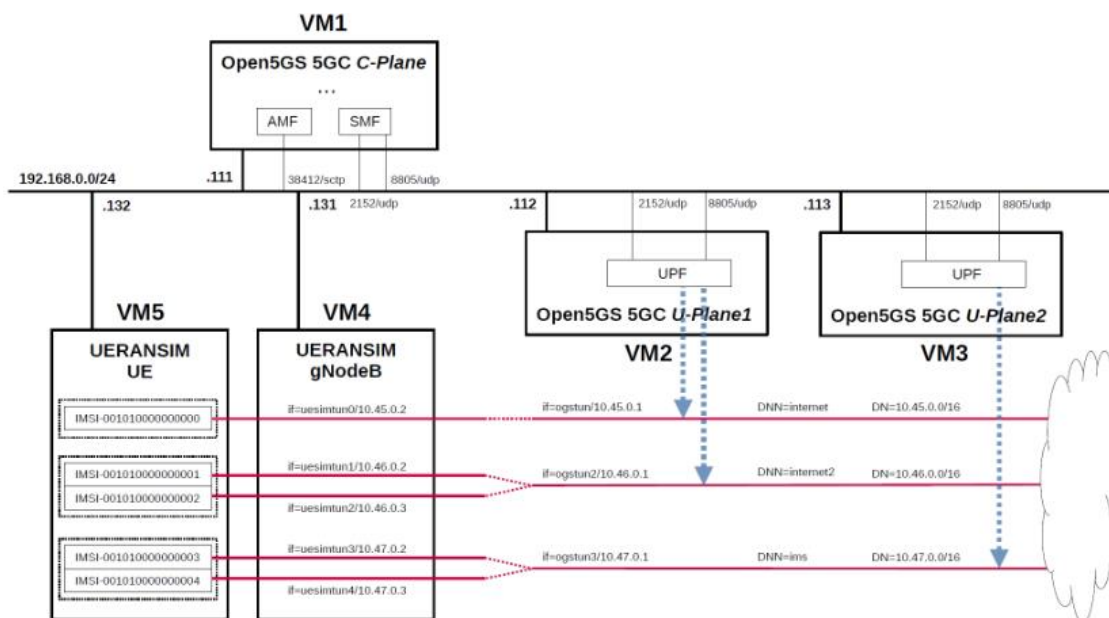


Figura 5

```

ED/PS]
[2023-01-06 22:00:37.865] [nas] [info] Performing local release of NAS con
nection
[2023-01-06 22:00:37.865] [nas] [info] UE switches to state [MM-DEREGISTER
ED/ATTEMPTING-REGISTRATION]
[2023-01-06 22:00:37.865] [rrc] [info] UE switches to state [RRC-IDLE]
[2023-01-06 22:00:37.865] [nas] [info] UE switches to state [CM-IDLE]
[2023-01-06 22:00:47.880] [nas] [debug] NAS timer[3511] expired [i]
[2023-01-06 22:00:47.880] [nas] [debug] Initial registration required due
to [T3511-EXPIRY-IN-ATT-REG]
[2023-01-06 22:00:47.880] [nas] [debug] UAC access attempt is allowed for
identity[0], category[MO_sig]
[2023-01-06 22:00:47.880] [nas] [debug] Sending Initial Registration
[2023-01-06 22:00:47.880] [rrc] [debug] Sending RRC Setup Request
[2023-01-06 22:00:47.880] [nas] [info] UE switches to state [MM-REGISTER-I
NITIATED]
[2023-01-06 22:00:47.880] [rrc] [info] RRC connection established
[2023-01-06 22:00:47.880] [rrc] [info] UE switches to state [RRC-CONNECTED
]
[2023-01-06 22:00:47.880] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-01-06 22:01:02.903] [nas] [debug] NAS timer[3510] expired [i]
[2023-01-06 22:01:02.903] [nas] [info] UE switches to state [MM-DEREGISTER
ED/PS]
[2023-01-06 22:01:02.903] [nas] [info] Performing local release of NAS con
nection
[2023-01-06 22:01:02.903] [rrc] [info] UE switches to state [RRC-IDLE]
[2023-01-06 22:01:02.903] [nas] [info] UE switches to state [MM-DEREGISTER
ED/ATTEMPTING-REGISTRATION]
[2023-01-06 22:01:02.903] [nas] [info] UE switches to state [CM-IDLE]
[2023-01-06 22:01:12.916] [nas] [debug] NAS timer[3511] expired [i]
[2023-01-06 22:01:12.916] [nas] [debug] Initial registration required due
to [T3511-EXPIRY-IN-ATT-REG]
[2023-01-06 22:01:12.916] [nas] [debug] UAC access attempt is allowed for
identity[0], category[MO_sig]
[2023-01-06 22:01:12.916] [nas] [debug] Sending Initial Registration
[2023-01-06 22:01:12.916] [nas] [info] UE switches to state [MM-REGISTER-I
NITIATED]
[2023-01-06 22:01:12.916] [rrc] [debug] Sending RRC Setup Request
[2023-01-06 22:01:12.916] [rrc] [info] RRC connection established
[2023-01-06 22:01:12.916] [rrc] [info] UE switches to state [RRC-CONNECTED
]
[2023-01-06 22:01:12.917] [nas] [info] UE switches to state [CM-CONNECTED]

```

Figura 6 – Terminal para um dos UE'S após inicialização (Implementação com 5 VM'S)

Nas imagens abaixo podemos ver que nos primeiros 8 pacotes o ping entre as UE's falha e os restantes o ping bem sucedido entre a UE e a UPF :

No.	Time	Source	Destination	Protocol	Length	Info
63	53.518253592	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
64	53.518635739	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
68	54.541117099	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
69	54.541421956	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect (f
70	54.541422056	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
74	55.563352153	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
75	55.563726169	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect (f
76	55.563726289	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request 10
111	70.118112846	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request 10
112	70.118416012	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply 10
114	71.179289945	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request 10
115	71.179597131	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply 10
117	72.207310604	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request 10
118	72.207894969	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply 10

```

<
>
> Frame 63: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface enp0s8, id 0
> Ethernet II, Src: PcsCompu_2b:2d:93 (08:00:27:2b:2d:93), Dst: PcsCompu_c5:2c:f1 (08:00:27:c5:2c:f1)
> Internet Protocol Version 4, Src: 192.168.56.101, Dst: 192.168.56.102
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
> Internet Protocol Version 4, Src: 10.45.0.2, Dst: 10.45.0.3
> Internet Control Message Protocol

```

Figura 7

No.	Time	Source	Destination	Protocol	Length	Info
63	53.518253592	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
64	53.518635739	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
68	54.541117099	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
69	54.541421956	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect
70	54.541422056	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
74	55.563352153	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
75	55.563726169	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect
76	55.563726289	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
111	70.118112846	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
112	70.118416012	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply
114	71.179289945	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
115	71.179597131	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply
117	72.207310604	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
118	72.207894969	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply

Frame 64: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface enp0s8, id 0
Ethernet II, Src: PcsCompu_c5:2c:f1 (08:00:27:c5:2c:f1), Dst: PcsCompu_2b:2d:93 (08:00:27:2b:2d:93)
Internet Protocol Version 4, Src: 192.168.56.102, Dst: 192.168.56.101
User Datagram Protocol, Src Port: 2152, Dst Port: 2152
GPRES Tunneling Protocol
Internet Protocol Version 4, Src: 10.45.0.2, Dst: 10.45.0.3
Internet Control Message Protocol

Figura 8

No.	Time	Source	Destination	Protocol	Length	Info
63	53.518253592	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
64	53.518635739	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
68	54.541117099	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
69	54.541421956	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect
70	54.541422056	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
74	55.563352153	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
75	55.563726169	10.45.0.1	10.45.0.2	GTP <I...	170	Redirect
76	55.563726289	10.45.0.2	10.45.0.3	GTP <I...	142	Echo (ping) request
111	70.118112846	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
112	70.118416012	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply
114	71.179289945	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
115	71.179597131	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply
117	72.207310604	10.45.0.2	10.45.0.1	GTP <I...	142	Echo (ping) request
118	72.207894969	10.45.0.1	10.45.0.2	GTP <I...	142	Echo (ping) reply

Figura 9

```
joao@UbuntuCM:~/UERANSIN$ ping www.google.com -I uesimtun1
PING (142.250.184.4) from 10.45.0.3 uesimtun1: 56(84) bytes of data.
64 bytes from mad41s10-in-f4.1e100.net (142.250.184.4): icmp_seq=1 ttl=113 time=17.3 ms
64 bytes from mad41s10-in-f4.1e100.net (142.250.184.4): icmp_seq=2 ttl=113 time=17.8 ms
64 bytes from mad41s10-in-f4.1e100.net (142.250.184.4): icmp_seq=3 ttl=113 time=18.9 ms
64 bytes from mad41s10-in-f4.1e100.net (142.250.184.4): icmp_seq=4 ttl=113 time=18.7 ms
```

Figura 10 – Ping para a internet a funcionar

Conclusão

Para concluir, mesmo com grande esforço da nossa parte, infelizmente não conseguimos resolver a comunicação entre os UE's, e por isso os objetivos cumpridos para o projeto ficaram um pouco aquém do esperado. Mas, estes obstáculos que encontramos ao longo do projeto contribuíram bastante para a nossa aprendizagem pois levou-nos a pesquisar e a estudar mais a fundo as redes 5G, o seu funcionamento, a sua arquitetura e os seus diversos elementos, nomeadamente UPF, AMF, SMF, gNB, UE's para tentarmos resolver os problemas encontrados pelo caminho, ou seja, podemos dizer que de certa forma estes obstáculos foram um "mal que vieram por bem".

Bibliografia

- [1] <https://open5gs.org/open5gs/docs/guide/01-quickstart/>
- [2] <https://github.com/aligungr/UERANSIM/wiki>
- [3] <https://open5gs.org/>
- [4] <https://github.com/aligungr/UERANSIM>
- [5] <https://medium.com/rahasak/5g-core-network-setup-with-open5gs-and-ueransim-cd0e77025fd7>
- [6] https://github.com/s5uishida/open5gs_5gc_ueransim_sample_config