Opis środowiska testowego wybranego rdzenia sieci 5G

[**1. Początkowe założenia**](#_awh9xk4xk0ar) **2**

[1.1 Używane skróty](#_cfrxmon66cs3) 2

[1.2 Założenia opisowe](#_mwxo086ztpcd) 2

[1.3 Wybrana implementacja rdzenia sieci 5G oraz środowisko](#_p9nh8b3vc53g) 2

[1.4 Schemat połączeń maszyn wirtualnych](#_8ez51nh5zudx) 3

[1.5 Konfiguracja systemów operacyjnych maszyn wirtualnych](#_emslb2375ef9) 3

[**2. Konfiguracje oprogramowania**](#_tjuy8aphk9lg) **4**

[2.2 Konfiguracja UE](#_pvxiou62rsff) 4

[UERANSIM/config/open5gs-ue.yaml](#_plx5nesa2ehp) 4

[2.3 Konfiguracja gNB](#_pivlrwecsogn) 4

[UERANSIM/config/open5gs-gnb.yaml](#_6vcu1mfi5cx9) 4

[2.3 Konfiguracja Core](#_owlls3ocw85s) 5

[/etc/open5gs/amf.yaml](#_l9ilea885bjj) 5

[/etc/open5gs/upf.yaml](#_z0siy8q456vq) 6

[Dodanie subskrybenta](#_cgnhi4yxn93b) 6

[**3. Skrypty automatyzacyjne**](#_vbuxqt2momkd) **7**

[3.1 Założenia](#_opgft95b0ofk) 7

[3.2 UE](#_id2fknojherg) 7

[auto.sh](#_ywcorq25j1tm) 7

[start.sh](#_ibdymytdl6h6) 7

[setup\_network.sh](#_lb3597cl7odt) 7

[3.3 gNB](#_5bbmi6lki2x0) 7

[auto.sh](#_scerez319o0j) 7

[start.sh](#_wejnw6lzig4t) 8

[3.4 Core](#_flj3ulapu265) 8

[auto.sh](#_35bua1qbjc5y) 8

[**4. Możliwości testowanego rdzenia**](#_3cwt3e3rlw86) **8**

# 

# Początkowe założenia

## 1.1 Używane skróty

Data Network - sieć zewnętrzna

Core - maszyna wirtualna z rdzeniem Open5GS

gNB - maszyna wirtualna z symulowanym stacją bazową (gNodeB)

UE - maszyna wirtualna z symulowanym terminalem końcowym (UE)

## 1.2 Założenia opisowe

Pogrubione linie oznaczają nasze komentarze, które nie są zawarte w plikach konfiguracyjnych lub skryptach, natomiast podkreślone znaki oznaczają zmienione wartości z domyślnych. Brak reszty pliku konfiguracyjnego oznacza brak jego zmiany w porównaniu do konfiguracji domyślnej, chyba że zaznaczono inaczej.

## 1.3 Wybrana implementacja rdzenia sieci 5G oraz środowisko

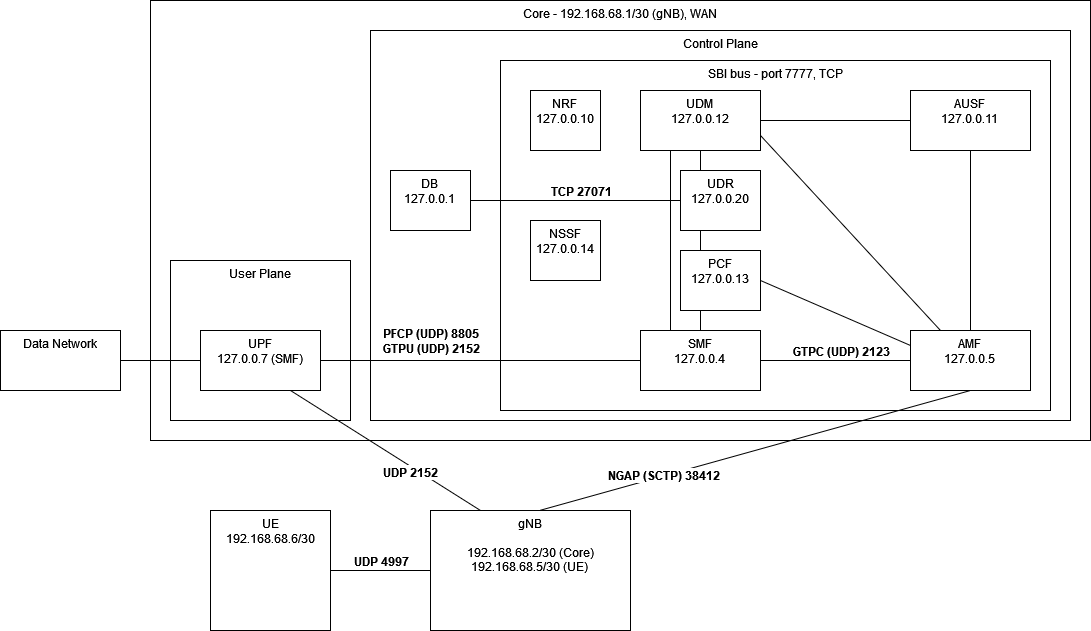
Na potrzeby zaznajomienia się z wybraną przez nas implementacją rdzenia sieci 5G, czyli z Open5GS (<https://open5gs.org/>), utworzyliśmy środowisko 3 maszyn wirtualnych, oparte na VirtualBoxie. Stacja bazowa oraz terminal końcowy symulowane są przy pomocy symulatora stacji bazowej i terminala końcowego UERANSIM (<https://github.com/aligungr/UERANSIM>).

Na każdej maszynie utworzony został użytkownik “user” z hasłem “zaq1@WSX”.

## 

## 1.4 Schemat połączeń maszyn wirtualnych

Połączenia maszyn wirtualnych zostały ustalone zgodnie z poniższym schematem:

gdzie karta sieciowa połączona z Data Network jest kartą NAT, natomiast pozostałe są kartami sieci wewnętrznych z sieciami nazwanymi “ue-gnb” oraz “gnb-5gcore”, które są połączone z kartami maszyn zgodnych z ich nazwą.

## 1.5 Konfiguracja systemów operacyjnych maszyn wirtualnych

Każda z maszyn bazuje na [Ubuntu Server 20.04.5 LTS (Focal Fossa)](https://releases.ubuntu.com/20.04/). Instalacja została pomniejszona o pakiety snap, pliki cache .deb oraz wyłączona została pamięć SWAP.

Na Core doinstalowano Open5GS oraz MongoDB (przy pomocy dokumentacji [Quickstart](https://open5gs.org/open5gs/docs/guide/01-quickstart/)), natomiast na gNB oraz UE zainstalowano symulator stacji bazowej i terminala końcowego UERANSIM.

# Konfiguracje oprogramowania

## 2.2 Konfiguracja UE

### UERANSIM/config/open5gs-ue.yaml

# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)

**# numer subskrybenta (pierwsze 5 cyfr to kod mcc oraz mnc)**

supi: 'imsi-001010000000001'

# Mobile Country Code value of HPLMN

**# kod krajowy**

mcc: '001'

# Mobile Network Code value of HPLMN (2 or 3 digits)

**# kod sieci**

mnc: '01'

**# oba te kody razem oznaczają sieć testową**

**# dane subskrybenta:**

# Permanent subscription key

key: '465B5CE8B199B49FAA5F0A2EE238A6BC'

# Operator code (OP or OPC) of the UE

op: 'E8ED289DEBA952E4283B54E88E6183CA'

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

opType: 'OPC'

# Authentication Management Field (AMF) value

amf: '8000'

# IMEI number of the device. It is used if no SUPI is provided

imei: '356938035643803'

# IMEISV number of the device. It is used if no SUPI and IMEI is provided

imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation

**# adres gNB**

gnbSearchList:

- 192.168.68.5

## 2.3 Konfiguracja gNB

### UERANSIM/config/open5gs-gnb.yaml

**# kod krajowy**

mcc: '001' # Mobile Country Code value

**# kod sieci**

mnc: '01' # Mobile Network Code value (2 or 3 digits)

ci: '0x000000010' # NR Cell Identity (36-bit)

idLength: 32 # NR gNB ID length in bits [22...32]

tac: 1 # Tracking Area Code

**# adres IP interfejsu widocznego przez UE**

linkIp: 192.168.68.5 # gNB's local IP address for Radio Link Simulation (Usually same with local IP)

**# adres IP interfejsu widocznego przez AMF**

ngapIp: 192.168.68.2 # gNB's local IP address for N2 Interface (Usually same with local IP)

**# adres IP interfejsu widocznego przez UPF**

gtpIp: 192.168.68.2 # gNB's local IP address for N3 Interface (Usually same with local IP)

**# adres IP i port AMF**

# List of AMF address information

amfConfigs:

- address: 192.168.68.1

port: 38412

## 2.3 Konfiguracja Core

### /etc/open5gs/amf.yaml

amf:

**# adres dla protokołu NGAP**

ngap:

- addr: 192.168.68.1

metrics:

addr: 127.0.0.5

port: 9090

**# identyfikator GUAMI**

guami:

- plmn\_id:

mcc: 001

mnc: 01

amf\_id:

region: 2

set: 1

tai:

- plmn\_id:

mcc: 001

mnc: 01

tac: 1

plmn\_support:

- plmn\_id:

mcc: 001

mnc: 01

s\_nssai:

- sst: 1

### /etc/open5gs/upf.yaml

upf:

**# adres dla protokołu GTPU (komunikacja z gNB)**

gtpu:

- addr: 192.168.68.1

**# adres wyjściowy do sieci globalnej**

**# jest to standardowa sieć tworzona na interfejsie ogstun przez Open5GS**

subnet:

- addr: 10.45.0.1/16

- addr: 2001:db8:cafe::1/48

### Dodanie subskrybenta

Dodaliśmy subskrybenta wykorzystując WebUI Open5GS oraz [dane subskrybenta ustawione przy konfiguracji UE](#_plx5nesa2ehp)

# 

# Skrypty automatyzacyjne

## 3.1 Założenia

Każdy skrypt należy uruchomić z podniesionymi uprawnieniami, chyba, że zaznaczono inaczej. Skrypty *auto.sh* uruchamiane są przez *cron* przy uruchomieniu systemu.   
Jeśli nie zadziała którakolwiek z akcji, może to być spowodowane wykonaniem danego polecenia zanim maszyna zdążyła uruchomić odpowiednie daemony. Skrypt można uruchomić wtedy ponownie. W środowiskach produkcyjnych działania tych skryptów powinny zapewnić ich poprawne wykonanie.

## 3.2 UE

### auto.sh

#!/bin/bash

ip addr add 192.168.68.6/30 dev enp0s3

Skrypt ten ustawia odpowiedni adres dla UE.

### start.sh

#!/bin/bash

UERANSIM/build/nr-ue -c UERANSIM/config/open5gs-ue.yaml

Skrypt ten uruchamia symulowany terminal końcowy UE oraz tworzy interfejs uesimtun0 do połączenia na zewnątrz.

### setup\_network.sh

#!/bin/bash

ip route add 10.45.0.0/16 dev uesimtun0

ip route add default via 10.45.0.1

Skrypt ten konfiguruje połączenie z gNB oraz Core. Musi być wykonany po poprawnym połączeniu, by można było z niego korzystać.

## 3.3 gNB

### auto.sh

#!/bin/bash

ip link set enp0s8 up

ip addr add 192.168.68.2/30 dev enp0s8

ip addr add 192.168.68.5/30 dev enp0s3

Skrypt ten ustawia odpowiednie adresy dla gNB.

### start.sh

#!/bin/bash

UERANSIM/build/nr-gnb -c UERANSIM/config/open5gs-gnb.yaml

Skrypt ten konfiguruje połączenie z Core

Jako jedyny należy uruchomić bez podnoszenia uprawnień!

## 3.4 Core

### auto.sh

#!/bin/bash

**# konfiguracja interfejsu Core-gNB**

ip link set enp0s8 up

ip addr add 192.168.68.1/30 dev enp0s8

**# restart wszystkich elementów**

systemctl restart mongod open5gs-webui open5gs-{amf,ausf,bsf,hss,mme,nrf,nssf,pcf,pcrf,scp,sgwc,sgwu,smf,udm,udr,upf}d

**# ustawienie routingu dla ruchu wychodzącego przez rdzeń**

iptables -t nat -A POSTROUTING -s 10.45.0.0/16 ! -o ogstun -j MASQUERADE

iptables -P FORWARD ACCEPT

sysctl -w net.ipv4.ip\_forward=1

# Możliwości testowanego rdzenia

Uzyskanie połączenia internetowego sprawdzonego przy pomocy sniffer.  
Dodanie abonenta przez WebUI.