Omec gNB simulator

It is a tool that simulates gNodeB and UE by generating NAS and NGAP messages for the configured UEs and call flows.

Supported 3GPP procedures:

```
1. UE Registration
2. UE Initiated PDU Session Establishment
3. UE Initiated De-registration
4. AN Release
5. UE Initiated Service Request
6. N/W triggered PDU Session Release
7. UE Requested PDU Session Release
8. N/W triggered UE Deregistration
```

Installation types:

1. Build gNBSim (We will be following this method)
2. Build a docker image for gNBSim (To bring up docker images official docs can be followed)

Clone and build steps:

1. Clone the repository
   git clone https://github.com/omec-project/gnbsim.git
2. cd gnbsim
3. Build the image by running
   go build
4. A “gnbsim” executable is created.

Configure gNBsim:

- By default, the gNB Sim reads the configuration from /gnbsim/config/gnb.conf file. To provide a different configuration file, use config/gnbsim.yaml

Configure all the basic parameters in gnbsim.yaml

To Run then gnbsim using config file gnbsim.yaml:

```
./gnbsim --cfg config/gnbsim.yaml
```

Some important configs:

1. HTTP API to create a new profile. The Below configuration enables HTTP server in gNBSim.

```yaml
config:
  gnbsim:
    httpServer:
      enable: true #enable httpServer in gnbsim
      port: 6000
```

2. Executing all enabled profiles in parallel or in sequential order.

```yaml
config:
  gnbsim:
    yamlCfgFiles:
      gnb.conf:
        configuration:
            execInParallel: false #run all profiles in parallel
```

3. Profiles can be enabled and disabled by setting the “enable” flag.

```yaml
profiles: # profile information
  - profileType: register # profile type
    profileName: profile1 # uniqely identifies a profile within application
    enable: false # Set true to execute the profile, false otherwise.
    gnbName: gnb1 # gNB to be used for this profile
    startImsi: 001010000000001
    ueCount: 1
    defaultAs: "192.168.60.1" #default icmp pkt destination
    opc: "E8ED289DEBA952E4283B54E88E6183CA"
    key: "465B5CE8B199B49FAA5F0A2EE238A6BC"
```

4. Custom profiles can be created using the template.

```
customProfiles:
  customProfiles1:
    profileType: custom # profile type
    profileName: custom1 # uniqely identifies a profile within application
    enable: true # Set true to execute the profile, false otherwise.
    execInParallel: false #run all subscribers in parallel
    stepTrigger: false #wait for trigger to move to next step
    gnbName: gnb1 # gNB to be used for this profile
    startImsi: 001010000000001
    ueCount: 1
    defaultAs: "192.168.60.1" #default icmp pkt destination
    opc: "E8ED289DEBA952E4283B54E88E6183CA"
    key: "465B5CE8B199B49FAA5F0A2EE238A6BC"
    sequenceNumber: "16f3b3f70fc2"
    plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC><MNC>
      mcc: 001 # Mobile Country Code (3 digits string, digit: 0~9)
      mnc: 01 # Mobile Network Code (2 or 3 digits string, digit: 0~9)
    startiteration: iteration1
    iterations:
      #at max 7 actions
      - "name": "iteration1"
        "1": "REGISTRATION-PROCEDURE 5"
        "2": "PDU-SESSION-ESTABLISHMENT-PROCEDURE 5"
        "3": "UE-INITIATED-DEREGISTRATION-PROCEDURE 10"
        # "3": "USER-DATA-PACKET-GENERATION-PROCEDURE 10"
        #   "next":  "iteration2"
        # - "name": "iteration2"
        # "1": "AN-RELEASE-PROCEDURE 100"
        # "2": "UE-TRIGGERED-SERVICE-REQUEST-PROCEDURE 10"
        # "repeat": 5
        # "next":  "iteration3"
        # - "name": "iteration3"
        # "1": "UE-INITIATED-DEREGISTRATION-PROCEDURE 10"
        #"repeat": 0 #default value 0 . i.e execute once
        #"next":  "quit" #default value quit. i.e. no further iteration to run
```

Quick gNBsim Integration with magma:

Setup Magma:

1. git clone https://github.com/magma/magma.git
2. cd magma/lte/gateway
3. vagrant up magma
4. vagrant ssh magma
5. cd magma/lte/gateway
6. enable5gfeatures in gateway.mconfig
7. make run
8. Add subscriber
    - cd ~/magma/lte/gateway/python/scripts
    - magtivate
    - subscriber_cli.py add --lte-auth-key 465B5CE8B199B49FAA5F0A2EE238A6BC --lte-auth-opc E8ED289DEBA952E4283B54E88E6183CA IMSI001010000000001
    - subscriber_cli.py update --lte-auth-key 465B5CE8B199B49FAA5F0A2EE238A6BC --apn-config internet,9,1,0,0,3000,4000,0,,,, --apn-config oai.ipv4,9,1,0,0,3000,4000,0,,,, --apn-config INTERNET,9,1,0,0,3000,4000,0,,,, --lte-auth-opc E8ED289DEBA952E4283B54E88E6183CA IMSI001010000000001

Setup Omec gNB sim:

1. Clone the repository
   git clone https://github.com/omec-project/gnbsim.git
2. cd gnbsim
3. Build the image by running
   go build
4. A “gnbsim” executable is created.
5. Set the parameters in the config file manually.
   Or
   Fetch and replace the config file.
   https://github.com/shashidhar-p/integration-magma/blob/main/omec-gnbsim/config/gnbsim.yaml

6. Run gNB sim with the config file.
   ./gnbsim --cfg config/gnbsim.yaml

7. Enable the profiles to be executed.

8. To run the enabled profiles:
   curl -i -X GET 127.0.0.1:8080/gnbsim/v1/executeConfigProfile