Architecure overview

- Etrib5gc makes a few changes to the 5GPP mobile core architecture.
- AMF now is decomposed into three network functions: DAMF, PRAN, and AMF.
- DAMF is a default AMF that does UE authentication and search for an AMF that can serve the UE. In 3GPP architecture, AMF can act as a default AMF. Decomposition reduce complexity of AMF implementation.
- DAMF having known of UE's slices must query NSSF to find suitable AMF for serving UE. In current system, NSSF is simply hard-coded into DAMF's configuration.
- PRAN stands for Proxy RAN. It is the part of AMF that interface to gnB through NGAP (N2 interface). PRAN converts the N2 interface to gnB into service based-interface (Restful). PRAN stand in the middle of gnB and AMF. The idea of introducing PRAN is moving forward to the new architecture where RAN-CORE get converged: gnB and PRAN get merged together.

The new AMF now does not has N2 interface. Instead it has a service-based interface to PRAN.

Setting up the system:

Configuration :

NF's agent configuration

- Configuration for each NF consist of two parts: NF-specific part, and service mesh part (agent). The later are similar among NFs and described below.
- All NFs (except the service controller) listen to port 7888 for SBI requests. Agents running on NFs listen to port 8889 for messages from service controller. The service controller listen on port 8888 for requests from agents.
- Listening IP address is defaulted to "0.0.0.0" if left empty. That means listening is on all network interfaces. Unless multiple network interfaces are avalable, just leave it empty.

```
"mesh": {
        "sbi": {
            "ip": "0.0.0.0",
            "port": 9100
        },
        "registry": {
            "agent": {
                "ip": "",
                "port": 7100
            },
            "controller": {
                "ip":"192.168.0.100",
                "port":1234
            }
        },
        "labels": {
            "app":"b5gc",
            "nf": "amf",
            "amfid":"112233"
        }
    }
```

- If running multiple NFs on a single host, listening ports must be assigned to different values.
- Controller IP address is the only item to be configured.
- Each NF is described with labels which is a key-value map in a similar way a POD is labled in Kubenetes. Indeed, this part must left empty in Kubenetes deployments. Labels are used to assigned NFs to services (defined at the service controller).

Controller configuration

- Default port listening address is 0.0.0.0:8888
- Controller defines services running on the core network.
- A service has a name
- A service has a selector for matching to labels of NFs. Matched NFs are serving the service.

```
{
      "id": "smf.208-93.2-030201",
      "selectors": {
        "nf": "smf",
        "slice": "2-030201"
      },
      "groups":{
          "v1": {
              "selectors": {
                  "version": "v1"
              }
          },
          "v2": {
              "selectors": {
                  "version": "v2"
              }
          }
      },
      "routes":[{
          "match":{
              "headers":{
                  "usertype":"tester"
              }
          },
          "destinations":[
            {
                "group": "v1",
                "weight":90,
                "lb": 0
            },
            {
                "group": "v2",
                "weight":10,
                "lb": 0
            }
          ]
      }]
}
```

- NFs in a service can be further grouped into subgroup with selectors. Subgroups are used for describing traffic splitting among NFs of the same service.
- Traffic routing for a service is defined by routes.
- A route composes of a matching rule and a set of destinations (subgroup). Traffic spliting and load balaning policy are defined on this destination set.

NF-spefic configuration

- all NFs must specify a Plmn Id (3GPP format)

```
"plmnId": {
        "mcc": "208",
        "mnc": "93"
    },
```

- PRAN and DAMF is identitied with a regional identity. Their service names include the identity (pran.208-93.daejeon; damf.208-93.daejeon). PRAN will select default AMF in the same region.

```
...
    "id": "daejeon"
...
```

- AMF must have an AmfId (3GPP format) and is is a part of the AMF service name (amf.209-93.112233) A PRAN must be configed with a list of AmfId that it can request.

```
...
    "amflist": ["112233","332211"]
...
```

- NGAP configuration at PRAN can be left empty. In such case default listening address is 0.0.0.0:38412

```
    "ngap": {
        "ip":"0.0.0.0",
        "port":32000
    },
```

- Important: PRAN is configured with a list of User plane Ran interfaces. They match to the definition in UPF topology configuration. When UE request a PDU session, this information is fowarded to SMF for selecting a UPF path from the topology.

```
    "rannets": ["an1"],
```

Running the system

- All NFs can run on a single machine. UPFs should be run on a separate machine
- Make sure to edit the topo.json file to reflect the deployment topology
- When running all control plane NFs on a single machine, make sure there will be no listening address confliction.
- all NFs can run with this command:

```
./amf -c amf.json
```

- except for the SMF that need an UPF topology:

```
./smf -c smf.json -t topo.json
```

Running order

1. Controller should be execute first.
2. Then all NFs (any order)
3. Execute UERANSIM's nr-gnb. It should connect to PRAN

4. Execute UERANSIM's nr-ue. It should established a PDU session to UPF

- Note that UPF and nr-ue needs to run with root user.
- Implementation of signaling procedures at AMF and SMF is still faulty. There are many corner cases that are not handled yet. So usually SMF and AMF need to be re-start every time UE make a connection to the network.

Running in Docker

- Docker should be able to run with non-root user (by adding the user to docker group and reboot):

```
sugo usermod -aG docker your-username
```

- Build docker images:

```
make docker
```

- The command will build all NFs from etr5gc, the UPF from free5gc and UERANSIM
- Use docker in interactive mode to setting up the whole system.
- The nr-ue and UPF must be run with NET_ADMIN capbability enabled

```
docker run --cap-add=NET_ADMIN -it b5gc-upf:latest -- sh
```