OpenAirCN-5G Project
Implementation Status and Current Activities

Dincer BEKEN, BLACKNED
Olivier CHOISY, B<>COM
Keliang DU, BUPT
Tien Thinh NGUYEN, EURECOM
Luhan WANG, BUPT

contact@openairinterface.org
contact@opensource5g.org_wluhan@bupt.edu.cn
Outline

- Introduction to OAI CN 5G Project
  - Introduction to 5G Core Network
    - Architecture Overview
    - High Level Functions
  - Implementation Status of 5GC Components
  - Roadmap
  - A Prototype Implementation (SMF)
Introduction

OAI CN-5G project

- To provide an open source implementation of 5G Core Network (5GC) as specified by 3GPP (SA)
- Main partners
  - Blackned, Bupt, B-Com, Eurecom, and ng4T
- License
  - 5G Core network license is OAI Public License v1.1
  - Contribution is opened to any people who signs the license agreement
- Project management
  - Synchro meeting every 2 weeks
  - Trello available
Outline

- Introduction to OAI CN 5G Project
- *Introduction to 5G Core Network*
  - Architecture Overview
  - High Level Functions
- Implementation Status of 5GC Components
- Roadmap
- A Prototype Implementation (SMF)
5G - Service Trends and Usage Scenarios

Source: ITU

Source: ETSI
Service-Oriented 5G Core Network

- Next Generation Network: To meet the needs of the range of services envisioned for 5G, with diverse performance requirements, across a wide variety of industries: *Flexible, Scalable, and Customizable*

- Service Based Architecture
  - support a modularized service, flexible and adaptable, with fast deployment cycles and updates for launching services on demand in the network
  - a set of network functions (NFs) providing services to other authorized NFs to access their services
5G System Architecture

- Access and Mobility Management Function (AMF)
- Session Management Function (SMF)
- User plane function (UPF)
- Policy Control Function (PCF)
- Authentication Server Function (AUSF)
- Unified Data Management (UDM)
- Network Exposure function (NEF)
- NF Repository function (NRF)
- Network Slice Selection Function (NSSF)
- Service based interfaces (SBI): Namf, Nsmf, Nudm, Nnrf, Nnssf, Nausf, Nnep, Nsmsf, Nudr, Npcf
5G System Architecture: From 4G to 5G Networks

CUPS: Control and User Plane Separation

MME: Mobility Management Entity
PCRF: Policy and Charging Rules Function
PGW: Packet Data Network (PDN) Gateway

SGW: Serving Gateway
HSS: Home Subscriber Server

UE

RAN

Control Plane
User Plane

PGW

PDN

EPC before CUPS

SGW-C

MME

HSS

SGW-U

PCRF

MME

HSS

PCRF

EPC after CUPS

SGW-U

PGW-U

PGW-C

SGW-C

PDN

SMF

UPF

NRF

NEF

NSSF

SMF

PGW-C

HSS

SMF

AUSF

UDM

UPF

PGW-U

PCRF

DN

4G-5G CN functions mapping

UE

AN

NRF

NEF

NSSF

SMF

PGW-C

SGW-C

MME

HSS

PCRF

PGW-U

PGW-U

PDN

DN
5G Core Network Functions

- **High level features**
  - **Network Access Control**
    - Network selection, Identification and authentication, Authorization, Access control and barring, Policy control, Lawful Interception
  - **Registration and Connection Management**
    - Registration Management, Connection Management, UE Mobility
  - **Session Management**
    - Support SSC mode 1,2,3
    - Support different PDU Session Type
    - Support for Local Area Data Network
  - **User Plane Management**
    - UE IP Address Management
    - CN Tunnel Info Management
    - Traffic Detection
    - UP Tunnel Management
  - **Network Slicing**
Service-based Architecture - NF Service Framework

- Protocol stack
  - HTTPv2: Mandatory Http custom header: 3gpp-sbi-message-priority

- Network Function (NF) Service Framework
  - Interactions between NF Service Consumer and NF Service Producer: Request-Response, Subscribe-Notify
  - NF service authorization
  - NF service registration, de-registration
  - NF service discovery
Service-based Architecture - NF Service Framework

- HTTP methods - Request/Response communication (CRUD)
  - Create/Read/Update/Delete a resource
    - Create a resource using HTTP POST
    - Read a resource using HTTP GET
    - Update a resource using HTTP PUT
    - Delete a resource using HTTP DELETE
Service-based Architecture - NF Service Framework

- **HTTP methods - Subscribe/Notify communication**
  - Notifications will be sent to NF Service Consumer when the change/event occurs at the Service Provider (Provider ~ Client, Consumer ~ Server)
  - Create/Modify/Delete subscriptions using HTTP POST, PUT/PATCH, DELETE

  ![Diagram](image)

  **Create a subscription using HTTP POST**
  1. POST `/xyz_subscriptions (XyzSubscription)`
  2. 201 Created (XyzSubscription)

  **Send a notification using HTTP POST**
  1. POST `{callback_ref} (Notification)`
  2. 204 No Content

  **Modify a subscription using HTTP PUT**
  1. PUT `/subscriptionId (XyzSubscription)`
  2. 204 No Content or 200 OK (XyzSubscription)

  **Delete a subscription using HTTP DELETE**
  1. DELETE `/subscriptionId ()`
  2. 204 No Content
Mandatory
- Access and Mobility Management Function (AMF)
- Session Management Function (SMF)
- User plane function (UPF)

Others
- NF Repository function (NRF)
- Authentication Server Function (AUSF)
- Unified Data Management (UDM)
- Policy Control Function (PCF)
- Network Exposure function (NEF)
- Network Slice Selection Function (NSSF)
- Service based interfaces (SBI):
  Namf, Nsmf, Nudm, Nnrf, Nnssf, Nausf, Nnrf, Nsmsf, Nudr, Npcf
Outline

- Introduction to OAI CN 5G Project
- Introduction to 5G Core Network
  - Architecture Overview
  - High Level Functions
- Implementation Status of 5GC Components
- Roadmap
- A Prototype Implementation (SMF)
Overall Implementation Status

- **Phase 1 (on-going):** To deploy a simple workflow e.g., UE registration/de-registration processes (connection and registration procedures)
  - AMF, SMF (+ UDM/AUSF)

- **Phase 2 (on-going):** Support PDU session-related procedures: create/modify/release PDU session (session management procedures)
  - AMF, SMF, UPF, UDM/AUSF

- **Phase 3:** Support further functionalities e.g., paging, mobility, Ipv6, authentication/authorization by DNN, multiple PDU session anchors, etc.

- **Main components:** AMF, SMF, UPF, UDM/AUSF, NRF (+gNB, UE)
  - Objective: testing with CI/CD (with ng4T tester)
OAI EPC - Current Implementation Status

- OAI Core network source code
  - [https://github.com/OPENAIRINTERFACE/openair-cn](https://github.com/OPENAIRINTERFACE/openair-cn)
  - [https://github.com/OPENAIRINTERFACE/openair-cn-cups](https://github.com/OPENAIRINTERFACE/openair-cn-cups)

- MME
  - Written in C
  - Separation between EMM and ESM
  - Stabilization:
    - Intensively tested with ng4T tester (including burst attaches, out of coverage scenarios with real eNBs, etc.)
    - Valgrind testing: Fixed all known memory leaks

- New SPGW (CUPS)
  - Written entirely in C++
  - Easy-to-use

- HSS (contributed by Sprint)
  - Rel14 compliant based on previous OAI HSS.
  - Written in C++
5GC Components: AMF Implementation Status

- Main partners: Blackned, Bupt, B-com, Eurecom
- Git repository: https://gitlab.eurecom.fr/oai/oai-cn5g-amf
- Programming language: C/C++
- **Current status:** under development
  - **Reuse the code from the latest OAI MME** (https://github.com/OPENAIRINTERFACE/openair-cn/tree/bearer_mod)
  - SBI interface implementation (done)
  - Integrate with new NGAP/NAS libraries (https://github.com/OPENAIRINTERFACE/openair5g-cn)
  - Libnas (https://github.com/OPENAIRINTERFACE/oai-libnascodec-cpp)
    - Library for coding/decoding NAS messages from UE
  - Libngapcodec (https://github.com/OPENAIRINTERFACE/oai-libngapcodec)
    - Library for coding/decoding values of protocol NGAP
  - Based on code generated by modifier asn1c generator (https://github.com/velichkov/asn1c)
  - Libngmessage
    - Implements higher functions to manipulate NGAP messages
- To be testbed with ng4T tester
### 5GC Components: AMF Conformance Interfaces

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Status</th>
<th>Comments</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>√</td>
<td>Communicate with UE via NAS message</td>
<td>NAS</td>
</tr>
<tr>
<td>N2</td>
<td>√</td>
<td>Communicate with gNB via NGAP message</td>
<td>NGAP</td>
</tr>
<tr>
<td>N11(Client/Server)</td>
<td>√</td>
<td>Interface to/from SMF (e.g., N1N2MessageTransfer)</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N8</td>
<td>√</td>
<td>Interface to UDM (e.g., retrieve UE subscription data)</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N15</td>
<td>×</td>
<td>To PCF, not implemented</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N14</td>
<td>×</td>
<td>Between 2 AMFs, not implemented</td>
<td>RESTful/HTTP</td>
</tr>
</tbody>
</table>
## 5GC Components: AMF Conformance Functions

### 3GPP TS 23.501 V16.1.0 §6.2.1

<table>
<thead>
<tr>
<th>Classification</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Termination of RAN CP interface (N2)</td>
<td>✓</td>
<td>Communicate with gNB via NGAP message</td>
</tr>
<tr>
<td>Termination of NAS (N1)</td>
<td>✓</td>
<td>Communicate with UE via NAS message</td>
</tr>
<tr>
<td>NAS ciphering and integrity protection</td>
<td>✗</td>
<td>Under development</td>
</tr>
<tr>
<td>Registration/Connection/Mobility/Reachability Management</td>
<td>✗</td>
<td>Under development</td>
</tr>
<tr>
<td>Provide transport for SM messages between UE and SMF</td>
<td>✗</td>
<td>Under development</td>
</tr>
<tr>
<td>Access Authentication/Authorization</td>
<td>✗</td>
<td>Under development</td>
</tr>
</tbody>
</table>
5GC Components: SMF Implementation Status

- Main partners: Eurecom, Bupt
- Git repository: https://gitlab.eurecom.fr/oai/oai-cn5g-smf
- Programming language: C++
- **Current status**: under development
  - **Reuse the code from OAI PGWC** (https://github.com/OPENAIRINTERFACE/openair-cn-cups)
  - NAS library (done)
  - NGAP library (80%)
  - SMF SBI interfaces (done)
  - Core SMF functionalities (under development)
- To be testbed with ng4T tester
## 5GC Components: SMF Conformance Interfaces

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Status</th>
<th>Comments</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>N11(Server)</td>
<td>√</td>
<td>Interface from AMF (SMF PDU Session Service)</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N11(Client)</td>
<td>√</td>
<td>Interface to AMF (Namf_N1N2MessageTransfer)</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N10(Client)</td>
<td>√</td>
<td>Interface to UDM (Nudm_SubscriberDataManagement)</td>
<td>RESTful/HTTP</td>
</tr>
<tr>
<td>N4(PFCP)</td>
<td>√</td>
<td>PFCP association; PFCP heartbeat connection and PFCP Session Establishment Request/Response</td>
<td>PFCP/UDP</td>
</tr>
<tr>
<td>N4(GTPU)</td>
<td>×</td>
<td>Data forwarding from/to UPF</td>
<td>GTPU/UDP</td>
</tr>
<tr>
<td>N7</td>
<td>×</td>
<td>To interface PCF, not implemented</td>
<td>RESTful/HTTP</td>
</tr>
</tbody>
</table>
## 5GC Components: SMF Conformance Functions

### 3GPP TS 23.501 V16.1.0 §6.2.2

<table>
<thead>
<tr>
<th>Classification</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Management, including tunnel maintain between UPF and AN node</td>
<td>✓</td>
<td>Only Session Establishment available</td>
</tr>
<tr>
<td>UE IP address allocation &amp; management</td>
<td>✓</td>
<td>Only support static ipv4 address allocation</td>
</tr>
<tr>
<td>DHCPv4/DHCPv6 functions</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Selection of UPF function</td>
<td>✓</td>
<td>Local configuration in SMF</td>
</tr>
<tr>
<td>Termination of interfaces towards Policy control functions</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Determine SSC mode of a session</td>
<td>✓</td>
<td>Only support SSC mode 1</td>
</tr>
</tbody>
</table>
5GC Components: UPF, UDM Implementation Status

- **UPF (and PFCP):** under development
  - Main partner: Eurecom
  - Programming language: C++
  - PFCP integrated into SMF implementation
  - Tested with SMF

- **UDM/AUSF:** under development
  - Main partner: Eurecom
  - Programming language: C++
  - *Reuse OAI HSS* ([https://github.com/OPENAIRINTERFACE/openair-cn](https://github.com/OPENAIRINTERFACE/openair-cn))
### 5GC Components: UPF Conformance Interfaces

<table>
<thead>
<tr>
<th>Interfaces</th>
<th>Status</th>
<th>Comments</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>N4</td>
<td>√</td>
<td>PFCP association; PFCP heartbeat connection and PFCP Session Establishment Request/Response</td>
<td>PFCP/UDP</td>
</tr>
<tr>
<td>N3</td>
<td>√</td>
<td>Interface with R(AN)</td>
<td>GTP (*)</td>
</tr>
<tr>
<td>N6</td>
<td>√</td>
<td>Interface with DN</td>
<td></td>
</tr>
<tr>
<td>N9</td>
<td>×</td>
<td>Interface between 2 UPFs</td>
<td>GTP (*)</td>
</tr>
</tbody>
</table>

(*) with header extensions for 5G (under development)
### 5GC Components: UPF Conformance Functions

**3GPP TS 23.501 V16.1.0 §6.2.3**

<table>
<thead>
<tr>
<th>Classification</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocation of UE IP address/prefix in response to SMF request</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>External PDU Session point of interconnect to Data Network</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Packet routing and forwarding</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>QoS handling for user plane (e.g. UL/DL rate enforcement, Reflective QoS marking in DL)</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Uplink Traffic verification (SDF to QoS Flow mapping)</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>Downlink packet buffering and downlink data notification triggering</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>
5GC Components: NRF Implementation Status

- Main partners: b<>com
- Operational NRF usable
  - NNRF-NFM [https://5g.labs.b-com.com/swagger/nrf-nfm/v1](https://5g.labs.b-com.com/swagger/nrf-nfm/v1)
  - NNRF-DISC [https://5g.labs.b-com.com/swagger/nrf-disc/v1](https://5g.labs.b-com.com/swagger/nrf-disc/v1)

**Current status**
- Based on 3GPP Rel 15.2
- Programming language: Java (Source code to be published)
- Integration with Jaeger + Opentracing (traces), Prometheus (monitoring)
- Error management
- Discovery and Management (notification, registration) implemented
- HTTPv2 and HTTPs supported (HTTP used in deployment currently)
- NRF integrated with b<>com AMF/SMF micro-services deployment (Kubernetes)

**Next steps**
- HTTPv2 in deployment
- Heartbeat feature
- External persistence mechanism
5GC Components: B<>com Integrated Solution

- Kubernetes based solution (micro-services architecture, multi services/pods) with NRF, b<>com SMF, b<>com AMF:
  - Automated deployment (helm based)

- AMF integrated with new NGAP/NAS libraries from b<>com

- 2 call flows implemented
  - gNodeb Attachment (gNodeb simu, AMF) [available]
  - UE PDUSessionEstablishmentRequest (UE simu, AMF, SMF, UPF) [available 12/2019]
    - Currently, 5G messages parsed successfully with Wireshark
    - Without authentication

- NFs management integrated with NRF (registration/notification/discovery)

- Next Steps
  - UE de-registration (close datapath session)
  - UE authentication mechanism
Outline

- Introduction to OAI CN 5G Project
- Introduction to 5G Core Network
  - Architecture Overview
  - High Level Functions
- Implementation Status of 5GC Components
- Roadmap
- A Prototype Implementation (SMF)
First version available Q1 2020 (phase 1,2)

- Phase 1, 2
  - Focusing on functional parts:
    - gNodeB Attach/Detach (AMF)
    - UE registration/de-registration wo authentication (AMF)
    - UE PDU Session create/modify/release (AMF, SMF, UPF)
    - NRF main features available
  - Testing with CI/CD using Ng4T tester

- Phase 3
  - Implementing further functionalities
  - Performance testing, load testing
Outline

- Introduction to OAI CN 5G Project
- Introduction to 5G Core Network
  - Architecture Overview
  - High Level Functions
- Implementation Status of 5GC Components
- Roadmap
- A Prototype Implementation (SMF)
A Prototype Implementation (SMF) - deployment

- Repository Of SMF
  - git clone https://gitlab.eurecom.fr/oai/oai-cn5g-smf.git

- SMF → build & run
  - cd oai-cn5g-smf
  - git checkout develop
  - cd build/scripts
  - ./build_smf -I -f
  - ./build_smf -c -V -b Debug -j
  - ./smf_conf.sh
  - sudo smf -c /usr/local/etc/oai/smf.conf -o

- UDM → build & run
  - cd oai-cn5g-smf
  - git reset --hard fb47b34e
  - cd src/test/udm
  - mkdir build
  - cd build
  - cd src
  - ./smf
  - sudo smf -c /usr/local/etc/oai/smf.conf -o

- UPF → build & run
  - cd oai-cn5g-smf
  - git checkout develop
  - cd build/scripts
  - ./build_upgw -I -f
  - ./build_upgw -c -V -b Debug -j
  - ./upgw_conf.sh
  - sudo upgw -c /usr/local/etc/oai/upgw.conf -o

- AMF http client
  - git clone http://forwards@gitlab.opensource5g.org:8888/forwards/HttpClient.git
  - cd HttpClient
  - git checkout go-http-client
  - cd src
  - make
  - cd server
  - git reset --hard a4a9a0832b99f8ac228654cae5c03a352e995fa6
  - cd smf/client
  - ./main-server "http://127.0.0.1:8080/nsmf-pdusession/v1/sm-contexts" -txtfile "/home/smf/oai-cn5g-smf/sm_encode_establishment_request.txt"

- AMF Server → build & run
  - cd oai-cn5g-smf
  - git checkout develop
  - cd src/server/amf
  - make
  - cd build
  - cd server
  - ./amf

- AMF http client
  - ./main-server "http://127.0.0.1:8080/nsmf-pdusession/v1/sm-contexts" -txtfile "/home/smf/oai-cn5g-smf/sm_encode_establishment_request.txt"

Figure. Deploy with several VNICS in one host
A Prototype Implementation (SMF) - flow chart

UE-triggered PDU Session Establishment Procedure

1. UE sends PDU Session Request (PDU SESSION ESTABLISHMENT REQUEST)
2. NGAP MM Context Created Request
3. NGAP MM Context Create Error
4. Check PDU Session Request Type
5. PDU Session Request Initial (PDU SESSION REQUEST)
6. Create SM Context and associated with SSID
7. Create NAS Session and associated with Supi
8. Create NAS Message Transfer Request
9. Create NAS Message Transfer Response
10. Decode NAS Context and associated with Supi
11. Decode NAS Context and associated with DNN
12. Check NAS Message Transfer Request
13. Encode NAS Message Transfer Response
14. SMF Service Endpoints
15. N11_HTTP SERVER(smf service endpoints)
16. Task SMF_APP
17. Task SMF_MM
18. Task SMF_N1
19. N11_HTTP SERVER(UEsmf SM)
20. UE

A Prototype Implementation (SMF) - results

- **Wireshark**

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>12:59:40.45348</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>PFCP</td>
<td>75</td>
<td>PCRF Association Setup Request</td>
</tr>
<tr>
<td>115</td>
<td>12:59:47.734</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>PFCP</td>
<td>79</td>
<td>PFCP Association Setup Response</td>
</tr>
<tr>
<td>245</td>
<td>17:08:32.231</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>PFCP</td>
<td>68</td>
<td>PFCP Heartbeat Request</td>
</tr>
<tr>
<td>248</td>
<td>17:09:59.676</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>PFCP</td>
<td>68</td>
<td>PFCP Heartbeat Response</td>
</tr>
<tr>
<td>269</td>
<td>18:49:49.0374</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>HTTP</td>
<td>1157</td>
<td>POST /api/session/v1/su-contex HTTP/1.1 (application/json)</td>
</tr>
<tr>
<td>281</td>
<td>18:50:09.0568</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>HTTP</td>
<td>222</td>
<td>GET /v2/0f932345619000/00-data HTTP/1.1</td>
</tr>
<tr>
<td>285</td>
<td>18:50:30.386</td>
<td>172.55.5.102</td>
<td>172.55.5.101</td>
<td>HTTP</td>
<td>652</td>
<td>HTTP/1.1 300 OK</td>
</tr>
<tr>
<td>291</td>
<td>18:50:57.1348</td>
<td>172.55.5.101</td>
<td>172.55.5.101</td>
<td>HTTP</td>
<td>134</td>
<td>HTTP/1.1 501 Created</td>
</tr>
<tr>
<td>299</td>
<td>18:51:31.6998</td>
<td>172.55.5.101</td>
<td>172.55.5.101</td>
<td>PFCP</td>
<td>114</td>
<td>PFCP Session Establishment Response</td>
</tr>
<tr>
<td>311</td>
<td>18:53:11.2381</td>
<td>172.16.1.102</td>
<td>172.16.1.102</td>
<td>HTTP</td>
<td>370</td>
<td>POST /api/session/v1/su-contex HTTP/1.1 (application/json)</td>
</tr>
<tr>
<td>316</td>
<td>18:53:48.892</td>
<td>172.16.1.102</td>
<td>172.16.1.102</td>
<td>HTTP</td>
<td>140</td>
<td>HTTP/1.1 200 OK</td>
</tr>
</tbody>
</table>

- **N4 association between SMF and UPF**

```
[2019-11-19T16:05:25.183646] [spgwu] [spgwu_sx] [start] Starting...
[2019-11-19T16:05:25.189686] [spgwu] [pfcp] [trace] Sending PFCP ASSOCIATION SETUP REQUEST, seq 11189661
[2019-11-19T16:05:25.191539] [spgwu] [spgwu_sx] [trace] Started
[2019-11-19T16:05:25.192214] [spgwu] [udp] [debug] Creating new listen socket on address 192.168.248.159 and port 2152
```

```
[2019-11-19T16:05:25.199296] [spgwu] [smf_n4] [info] handle_receive(31 bytes)
[2019-11-19T16:05:25.199871] [spgwu] [smf_n4] [info] handle_receive(16 bytes)
[2019-11-19T16:05:25.202598] [spgwu] [pfcp] [trace] Sending PFCP ASSOCIATION SETUP RESPONSE, seq 11189661
[2019-11-19T16:05:30.212901] [spgwu] [smf_n4] [info] handle_receive(16 bytes)
[2019-11-19T16:05:30.213235] [spgwu] [smf_n4] [trace] handle_receive_pfcp_msg msg type 1 length 12
[2019-11-19T16:05:30.214263] [spgwu] [pfcp] [trace] Sending HEARTBEAT RESPONSE, seq 11189662
```
A Prototype Implementation (SMF) - results

- N4 heartbeat connection

- PFCP Session Establishment Request/Response
A Prototype Implementation (SMF) - results

- Retrieve Session Management Subscription data from UDM

- Assign ipv4 address and create a pdu session
Thank you for your attention!