OAI Webinar Chapter Two

Walkthrough 3GPP Specifications

Feb 21, 2022

Rohan Kharade
Agenda

- **Part 1: Overview 3GPP specifications**
  - Introduction, challenges, recommendations

- **Part 2: Implementing procedure/protocols from 3gpp specification**
  - Call flows, UML diagrams, struct/class (IEs)

- **Part 3: Some use cases**
  - Example case of pfcp, gtp1u etc.
Part 1: Overview 3GPP specifications

Introduction, challenges, recommendations
3GPP - 3rd Generation Partnership Project

- Project to produce the reports and specifications for 3GPP technologies.

- Seven SDOs worldwide (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC)

- Project Co-ordination Groups-
  - Radio Access Network
  - Service & System Aspects
  - Core Network & Terminals
3GPP - 3rd Generation Partnership Project

- Releases -> functionally frozen and ready for implementation

https://www.qualcomm.com/news/onq/2017/08/02/understanding-3gpp-starting-basics
3GPP - 3rd Generation Partnership Project

- Organised into organized into 16 specialized Working Groups (WGs)

https://www.3gpp.org/specifications-groups
3GPP - 3rd Generation Partnership Project

- Organised into 16 specialized Working Groups (WGs)

  - Overall architecture and service capabilities
  - Specifying terminal interfaces, terminal capabilities

TSG RAN
- Radio Access Network
- RAN WG1: Radio Layer 1 (Physical layer)
- RAN WG2: Radio Layer 2 and Radio Layer 3 Radio Resource Control
- RAN WG3: Radio Transmission and Reception and Related Network Interfaces
- RAN WG4: Radio Performance and Protocol Aspects
- RAN WG5: Mobile Terminal Conformance Testing
- RAN WG6: Application Enabler and Critical Communication Applications

TSG SA
- Service & System Aspects
- SA WG1: Services
- SA WG2: System Architecture and Services
- SA WG3: Security and Privacy
- SA WG4: Multimedia Coders, Systmes and Services
- SA WG5: Management, Orchestration and Charging
- SA WG6: Application Enabler and Critical Communication Applications

TSG CT
- Core Network & Terminals
- CT WG1: User Equipment - Core Network protocols
- CT WG2: Interworking with External Networks & Policy and Charging Control
- CT WG3: Core Network Protocols
- CT WG4: Core Network Protocols
- CT WG5: Short Range Communication Applications
- CT WG6: Smart Card Application Aspects

https://www.3gpp.org/specifications-groups
Why reading & understanding 3GPP specs is challenging?

- So many technologies
- Choosing right specification
- Some specifications are very big
- They are interlinked
- Lack of normative references
Recommended Practice

- Know what to read

<table>
<thead>
<tr>
<th>General Information (not used)</th>
<th>00 series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>21 series</td>
</tr>
<tr>
<td></td>
<td>41 series</td>
</tr>
<tr>
<td></td>
<td>01 series</td>
</tr>
<tr>
<td>Service aspects (&quot;Stage 1&quot;)</td>
<td>22 series</td>
</tr>
<tr>
<td></td>
<td>42 series</td>
</tr>
<tr>
<td></td>
<td>02 series</td>
</tr>
<tr>
<td>Technical realization (&quot;Stage 2&quot;)</td>
<td>23 series</td>
</tr>
<tr>
<td></td>
<td>43 series</td>
</tr>
<tr>
<td></td>
<td>03 series</td>
</tr>
<tr>
<td>Signaling protocols (&quot;Stage 3&quot;) - user equipment to network</td>
<td>24 series</td>
</tr>
<tr>
<td></td>
<td>44 series</td>
</tr>
<tr>
<td></td>
<td>04 series</td>
</tr>
<tr>
<td>Radio aspects</td>
<td>25 series</td>
</tr>
<tr>
<td></td>
<td>45 series</td>
</tr>
<tr>
<td></td>
<td>05 series</td>
</tr>
<tr>
<td>CODECs</td>
<td>26 series</td>
</tr>
<tr>
<td></td>
<td>46 series</td>
</tr>
<tr>
<td></td>
<td>06 series</td>
</tr>
<tr>
<td>Data</td>
<td>27 series</td>
</tr>
<tr>
<td></td>
<td>47 series (none exists)</td>
</tr>
<tr>
<td></td>
<td>07 series</td>
</tr>
<tr>
<td>Signaling protocols (&quot;Stage 3&quot;) - (R2S-CN) and OAM&amp;P and Charging (overflow from 32 - range)</td>
<td>28 series</td>
</tr>
<tr>
<td></td>
<td>48 series</td>
</tr>
<tr>
<td></td>
<td>08 series</td>
</tr>
<tr>
<td>Signaling protocols (&quot;Stage 3&quot;) - intra-fied-network</td>
<td>29 series</td>
</tr>
<tr>
<td></td>
<td>49 series</td>
</tr>
<tr>
<td></td>
<td>09 series</td>
</tr>
<tr>
<td>Programme management</td>
<td>30 series</td>
</tr>
<tr>
<td></td>
<td>50 series</td>
</tr>
<tr>
<td></td>
<td>10 series</td>
</tr>
<tr>
<td>Subscriber Identity Module (SIM / USIM), IC Cards, Test specs.</td>
<td>31 series</td>
</tr>
<tr>
<td></td>
<td>51 series</td>
</tr>
<tr>
<td></td>
<td>11 series</td>
</tr>
<tr>
<td>OAM&amp;P and Charging</td>
<td>32 series</td>
</tr>
<tr>
<td></td>
<td>52 series</td>
</tr>
<tr>
<td></td>
<td>12 series</td>
</tr>
<tr>
<td>Access requirements and test specifications</td>
<td>13 series (1)</td>
</tr>
<tr>
<td></td>
<td>15 series (1)</td>
</tr>
<tr>
<td>Security aspects</td>
<td>33 series (2)</td>
</tr>
<tr>
<td></td>
<td>11 series</td>
</tr>
<tr>
<td>UE and USIM test specifications</td>
<td>34 series (2)</td>
</tr>
</tbody>
</table>

Series Number

38.413 NR
23.501 NGAP

Technical Realization (Stage 2)
5GS Arch.

part of series number

V = version e.g. 16.0.0.

https://www.3gpp.org/specifications/79-specification-numbering
https://wirelessbrew.com/5g-nr/guide-to-reading-3gpp-specifications/
Recommended Practice

- Know what to read

5G System
- TS 23.501: System Architecture for the 5G System, Stage 2, Rel 16.0.0, 2019-03
- TS 23.502: Procedures for 5G System, Stage 2, Rel 16.0.0, 2019-03
- TS 29.500: 5G System, Technical Realization of Service Based Architecture, Stage 3, Rel 16.0.0, 2019-03
- TS 29.501: 5G System, Principles and Guidelines for Services Definition, Stage 3, Rel 16.0.0, 2019-03
- TS 29.571: Common Data Types for Service Based Interfaces, Stage 3, v16.0.0, 2019-03

Security
- TS 33.102 - Security architecture, v16.0.0, 2020-07-10

NAS:

NGAP
- TS 38.413, NG Application Protocol (NGAP), v16.0.0, 2019-12

PFCP
- TS 29.244, Interface between the Control Plane and the User Plane nodes, v16.0.0, 2019-06-13
- TS 23.503, Policy and Charging Control Framework for the 5G System, Stage 2, v16.0.0, 2019-03

SMF
- TS 29.502 - Session Management Services, Stage 3, v16.0.0, 2019-03
- TS 29.508 - Session Management Event Exposure Service, Stage 3, v16.0.0, 2019-03

AMF
- TS 29.518 - Access and Mobility Management Services, Stage 3, v16.0.0, 2019-03

UDM/AUSF
- 3GPP TS 33.501 V16.0.0 (2019-09): Security architecture and procedures for 5G system
- UDM: TS 29.503 - Unified Data Management Services, Stage 3, v16.0.0, 2018-09

NRF
- TS 29.510 - Network Function Repository Services, Stage 3, v16.0.0, 2019-03

PCF
- TS 29.507 - Access and Mobility Policy Control Service, Stage 3, v16.0.0, 2019-03
- TS 29.512 - Session Management Policy Control Service, Stage 3, v16.0.0, 2019-03

NSSF
- TS 29.531 - Network Slice Selection Services; Stage 3, v16.0.0 2019-03, 2019-09-23

https://www.3gpp.org/specifications/79-specification-numbering
https://wirelessbrew.com/5g-nr/guide-to-reading-3gpp-specifications/
Recommended Practice

- Read respective specification first then references

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document:

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
[7] 3GPP TS 23.640: "Technical realisation of the Short Message Service (SMS)".

https://www.3gpp.org/specifications/79-specification-numbering
Recommended Practice

- Search for normative references
  - Blogs, tutorials, use cases scenarios
  - Whitepapers

https://www.3gpp.org/specifications/79-specification-numbering
Recommended Practice ……………. !!!

- Take printout *when necessary*
Recommended Practice ................ !!!

- Take printout *when necessary*
- Always good to ask
Part 2: Implementing procedure/protocols from 3gpp specification

Call flows => UML diagrams => IEs etc.
5CN Communication Mechanism

- Communication within NFs (SBIs) -
  - Transport layer -> TCP
  - Application layer -> http (serialization with json)
  - Openapi -> interface definitions

- Communication with User Plane (Protocols)-
  - NAS
  - NGAP (serialization with ASN1.c)
  - SCTP (at lower layer)
  - PFCP
  - gtpv1u
3GPP Procedures (Call Flows)

- 3GPP TS 23.502 -> Procedures for the 5G System
3GPP Procedures (Call Flows)

- 3GPP TS 23.502 -> Procedures for the 5G System
3GPP Procedures (Call Flows - Create detailed call flow)

- 3GPP TS 23.502 -> Procedures for the 5G System

![Diagram of PFCP Session Establishment Request]

### Table 7.5.2.1-1: Information Elements in a PFCP Session Establishment Request

<table>
<thead>
<tr>
<th>Information elements</th>
<th>P</th>
<th>Condition / Comment</th>
<th>Applicability</th>
<th>IE Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>M</td>
<td>This IE shall contain the unique identifier of the sending Node.</td>
<td>X X X X</td>
<td>Node ID</td>
</tr>
<tr>
<td>CP F-SEID</td>
<td>M</td>
<td>This IE shall contain the unique identifier allocated by the CP function identifying the session.</td>
<td>X X X X</td>
<td>F-SEID</td>
</tr>
<tr>
<td>Create PDR</td>
<td>M</td>
<td>This IE shall be present for at least one PDR to be associated to the PFCP session.</td>
<td>X X X X</td>
<td>Create PDR</td>
</tr>
<tr>
<td>Create FAR</td>
<td>M</td>
<td>This IE shall be present for at least one FAR to be associated to the PFCP session.</td>
<td>X X X X</td>
<td>Create FAR</td>
</tr>
<tr>
<td>Create URR</td>
<td>C</td>
<td>This IE shall be present if a measurement action shall be applied to packets matching one or more PDR(s) of this PFCP session.</td>
<td>X X X X</td>
<td>Create URR</td>
</tr>
</tbody>
</table>

See Table 7.5.3.1

See Table 7.5.3.1

See Table 7.5.3.1
3GPP Procedures (Call Flows - UML Class Diagram)

### Session Establishment Request

```
+ M:node_id : node_id_t
+ M:cp_f_seid : cp_f_seid_t
+ M:create_pdr : create_pdr_t
+ M:creat_far : creat_far
+ C:create_uurr : create_uurr
+ C:create_qer : create_qer
+ O:create_bar : create_bar
+ C:create_traffic_endpoint : create_traffic_endpoint_t
+ C:pdn_type : pdn_type_t
+ C:dn_type : dn_type_t
+ C:sgw_c_fq_csid : fq_csid_t
+ C:mme_fq_csid : fq_csid_t
+ C:pgw_c_fq_csid : fq_csid_t
+ C:epdg_fq_csid : fq_csid_t
+ O:user_plane_inactivity_timer : user_plane_inactivity_timer_t
+ O:user_id : user_id_t
+ O:trace_information : trace_information_t
+ O:apn/dnn : apn_t/dnn_t
+ C:create_mar : create_mar_t
+ get_()
+ set_()
```
3GPP Procedures (Call Flows - UML Class Diagram)

### Class Name

- M: node_id
- M: cp_f_seid
- M: create_pdr
- M: create_far
- C: create_uerr
- C: create_gei
- C: create_bar
- C: create_traffic_endpoint
- C: pdn_type
- C: dn_type
- C: sgw_c fq_csid
- C: mme fq_csid
- C: pgw_c fq_csid
- C: epdg fq_csid
- C: tawn fq_csid
- O: user plane inactivity timer
- O: user id
- O: trace information
- O: apn dnn
- C: create mar

### Class Attributes

- + get_() 
- + set_()
OAI CN Repository Structure

- Cmake build directory, BM install scripts
- Scripts used for CI/CD validation
- Dockerfiles for Ubuntu and RHEL
- Documentation (feature set)
- Config files
- Some init scripts
- Implementation of 3gpp procedures and protocols

https://gitlab.eurecom.fr/oai/cn5g
https://gitlab.eurecom.fr/oai/cn5g/oci-cn5g-amf/-/wikis/home
OAI CN Repository Structure

/src

3gpp protocols

https://gitlab.eurecom.fr/oai/cn5g
OAI CN Repository Structure

```
/srch/<nf_name>_app/
```

3gpp procedures

https://gitlab.eurecom.fr/oai/cn5g
OAI CN Repository Structure

/src/common

3gpp common data types, enumerations, constants

https://gitlab.eurecom.fr/oai/cn5g
OAI CN Repository Structure

/src/oai_<nf_name>/

- Parse config parameters
- Initiate application layer
- Initiate thread manager (ITTI or Task Manager)
- Initiate HTTPv1 & HTTPv2 server

generic main.cpp

```c
int main(int argc, char** argv) {
    // Config
    nrf_cfg.load(Options::getLibconfigConfig());
    nrf_cfg.display();
    // Event subsystem
    nrf_event ev;
    // NRF application layer
    nrf_app_inst = new nrf_app(Options::getLibconfigConfig(), ev);
    // Task Manager
    task_manager tm(ev);
    std::thread task_manager_thread6(task_manager::run, 6tm);
    // PID file
    // Currently hard-coded value: TODO: add as config option.
    string pid_file_name = get_exe_absolute_path("/var/run", nrf_cfg.instance);
    if (!is_pids_file_lock_success(pid_file_name.c_str())) {
        Logger::nrf_app().error("lock PID file %s failed\n", pid_file_name.c_str());
        exit(-1);
    }
    // NRF Pistache API server (HTTP1)
    Pistache::Address addr{
        std::string(int_16bit)*((struct in_addr) nrf_cfg.sbl.addr4),
        Pistache::Port(nrf_cfg.sbl.port))};
    api_server = new NRFApi2Server(addr, nrf_app_inst);
    api_server->init();
    std::thread nrf_manager_thread6(NRFApi2Server::start, api_server);
    // NRF NGHTTP API server (HTTP2)
    nrf_api_server_2 = new nrf_http2_server(
        convertTostring(nrf_cfg.sbl.addr4), nrf_cfg.sbl_http2 port, nrf_app_inst);
    std::thread nrf_http2_manager_thread6(NRFHttp2Server::start, nrf_api_server_2);
    nrf_manager.join();
    nrf_http2_manager.join();
}
```

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-nrf
Part 3: Use cases

Example case of pfcp, gtp1u etc.
Example case of PFCP

- TS 29.244 - Interface between the Control Plane and the User Plane nodes
- Sx/N4 interface
- UDP based, Port 8805

Figure 5.2.1-1: Packet processing flow in the UP function
Example case of PFCP

- TS 29.244 - Interface between the Control Plane and the User Plane nodes
- Sx/N4 interface
- UDP based, Port 8805

---

**Figure 7.2.1-1: PFCP Message Format**
Example case of PFCP

Node Related Messages

Session Related Messages

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/common/3gpp_29.244.h
Example case of PFCP

Generic PFCP header
Example case of PFCP

Parent class for pfcp header - Members

```
class pfcp_msg_header : public stream_serializable {
private:
  Message_PFCP_MSG_HEADER_MTN_SIZE_B;
  union {
    struct {
      uint8_t s : 1;
      uint8_t mp : 1;
      uint8_t sequence : 3;
      uint8_t version : 3;
    } bf;
    uint8_t b;
  } u11;
  uint8_t message_type;
  uint16_t message_length;
  uint64_t seid;
  uint32_t sequence_number;
  union {
    struct {
      uint8_t spare : 4;
      uint8_t message_priority : 4;
    } bf;
    uint8_t b;
  } u2;
}
```

Figure 7.2.2.3-1: PFCP message Header for session related messages
Example case of PFCP

Parent class for pfcp header - Methods (get/set)

```cpp
void set_ssid(const uint64_t &s) {
    ssid = s;
    if ((u1_bf.s == 0) {
        U1.BF.s = 1;
        message_length += 8;
    }
}

bool has_ssid() const { return (u1_bf.s == 1); }

uint64_t get_ssid() const { return ssid; }

void set_message_type(const uint8_t &t) { message_type = t; }

uint8_t get_message_type() const { return message_type; }

void set_message_length(const uint16_t &l) { message_length = l; }

uint16_t get_message_length() const { return message_length; }

void set_sequence_number(const uint32_t &s) {
    sequence_number = s & 0x00FFFFFF;
}
```

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.hpp
Example case of PFCP

Parent class for pfcp header - Methods (serialization/deserialization)

- **Deserialization** *(is.read())*

```cpp
virtual void load_from(std::istream& is) {
    is.read(reinterpret_cast<char*>(u1.b), sizeof(u1.b));
    is.read(reinterpret_cast<char*>(&message_type), sizeof(message_type));
    is.read(reinterpret_cast<char*>(&message_length), sizeof(message_length));
    if (u1.bf.s) {
        is.read(reinterpret_cast<char*>(&sequid), sizeof(sequid));
        sequid = be64toh(sequid);
    } else {
        sequid = 0;
    }
    uint8_t sn[3];
    is.read(reinterpret_cast<char*>(&sn), 3);
    sequence_number =
        (((uint32_t) sn[0]) << 16) | (((uint32_t) sn[1]) << 8) | sn[2];
    is.read(reinterpret_cast<char*>(u2.b), sizeof(u2.b));
}
```

- **Serialization** *(is.write())*

```cpp
virtual void dump_to(std::ostream& os) {
}
```

![Figure 7.2.2.3-1: PFCP message Header for session related messages](https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.hpp)
Example case of PFCP

- **IE NODE ID** - ID of PFCP peer endpoints

### Figure 8.2.38-1: Node ID

<table>
<thead>
<tr>
<th>Octets</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 2</td>
<td>8</td>
</tr>
<tr>
<td>3 to 4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6 to 0</td>
<td>5</td>
</tr>
<tr>
<td>m to (n+4)</td>
<td>4, 3, 2, 1</td>
</tr>
</tbody>
</table>

- Type = 60 (decimal)
- Length = n
- Spare
- Node ID value

These octet(s) is/are present only if explicitly specified.

### Table 8.2.38-2: Node ID Type

<table>
<thead>
<tr>
<th>Node ID Type Value (Decimal)</th>
<th>Node ID Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>1</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>2</td>
<td>FQDN</td>
</tr>
<tr>
<td>3 to 15</td>
<td>Spare, for future use.</td>
</tr>
</tbody>
</table>

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.hpp
Example case of PFCP

IE NODE ID

```cpp
// IE NODE_ID
class pfcp_node_id_ie : public pfcp_ie {
public:
    union {
        struct {
            uint8_t node_id_type : 4;
            uint8_t spare1 : 4;
        } b;
    uint8_t b;
    } u1;

    struct in_addr ipv4_address;
    struct in6_addr ipv6_address;
    std::string fqdn;
}
```

### Figure 8.2.38-1: Node ID

<table>
<thead>
<tr>
<th>Octets</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 2</td>
<td>8: Type = 60 (decimal)</td>
</tr>
<tr>
<td>3 to 4</td>
<td>7: Length = n</td>
</tr>
<tr>
<td>5</td>
<td>6: Spare</td>
</tr>
<tr>
<td>6 to 0</td>
<td>5: Node ID Type</td>
</tr>
<tr>
<td>m to (n+4)</td>
<td>4: Node ID value</td>
</tr>
</tbody>
</table>

These octet(s) is/are present only if explicitly specified

### Table 8.2.38-2: Node ID Type

<table>
<thead>
<tr>
<th>Node ID Type Value (Decimal)</th>
<th>Node ID Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>1</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>2</td>
<td>FQDN</td>
</tr>
<tr>
<td>3 to 15</td>
<td>Spare, for future use.</td>
</tr>
</tbody>
</table>

IE included in the message

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.hpp
Example case of PFCP

Deserialization

### Table 8.2.38-2: Node ID Type

<table>
<thead>
<tr>
<th>Node ID Type Value (Decimal)</th>
<th>Node ID Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>1</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>2</td>
<td>FQDN</td>
</tr>
<tr>
<td>3 to 15</td>
<td>Spare, for future use.</td>
</tr>
</tbody>
</table>

Figure 8.2.38-1: Node ID

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.hpp
Example case of PFCP

Sample node related message

7.4.4.5 PFCP Association Release Request

Table 7.4.4.5-1: Information Elements in a PFCP Association Release Request

<table>
<thead>
<tr>
<th>Information elements</th>
<th>P</th>
<th>Condition / Comment</th>
<th>IE Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node ID</td>
<td>M</td>
<td>This IE shall contain the unique identifier of the sending Node.</td>
<td>Node ID</td>
</tr>
</tbody>
</table>

PFCP Message

IE included in the message

Table 8.2.38-2: Node ID Type

<table>
<thead>
<tr>
<th>Node ID Type Value (Decimal)</th>
<th>Node ID Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IPv4 address</td>
</tr>
<tr>
<td>1</td>
<td>IPv6 address</td>
</tr>
<tr>
<td>2</td>
<td>FQDN</td>
</tr>
<tr>
<td>3 to 15</td>
<td>Spare, for future use.</td>
</tr>
</tbody>
</table>
Example case of PFCP

Sample node related message

- IE Association Release request -

```cpp
pfcp_msg::pfcp_msg(const pfcp_association_release_request& pfcp_ies) :
    pfcp_msg_header() {
    les = {};
    set_message_type(PFCP_ASSOCIATION_RELEASE_REQUEST);
    if (pfcp_les.node_id.first) {
        std::shared_ptr<pfcp_node_id_le> s1e{
            new pfcp_node_id_le(pfcp_les.node_id.second)};
        add_ie(s1e);
    }
}
```

Adding IE

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-smf/-/blob/master/src/pfcp/3gpp_29.244.cpp
Example case of PFCP

Sample node related message

- IE Association Release request -

```cpp
pfcp_msg::pfcp_msg(const pfcp_association_release_request& pfcp_ies)
    : pfcp_msg_header() {
    ies = {};
    set_message_type(PFCP_ASSOCIATION_RELEASE_REQUEST);
    if (pfcp_ies.node_id.first) {
        std::shared_ptr<pfcp_node_id_ie> sie{
            new pfcp_node_id_ie(pfcp_ies.node_id.second)};
        add_ie(sie);
    }
```
Summary

- Short overview of 3GPP specifications for 5G CN
- Recommended practice for studying specs
- Implementation in OAI CN stack
- Examples of encoding/decoding IEs
Example case of gtp1u

- GTP - GPRS Tunneling Protocol
- 3GPP TS 29.281 - User Plane (GTPv1-U)
- S1-U, S11-U, S2a, S2b, X2, S4, S5, S8, S12 (EPS)
- F1-U, Xn, N3, N9 and N19 (5GS)

![Diagram of GTP header and original UE packet]

**Figure 5.1-1: Outline of the GTP-U Header**

---

NOTE 0: (*) This bit is a spare bit. It shall be sent as '0'. The receiver shall not evaluate this bit.

NOTE 1: 1) This field shall only be evaluated when indicated by the S flag set to 1.

NOTE 2: 2) This field shall only be evaluated when indicated by the PN flag set to 1.

NOTE 3: 3) This field shall only be evaluated when indicated by the E flag set to 1.

NOTE 4: 4) This field shall be present if and only if any one or more of the S, PN and E flags are set.
Example case of gtp1u
Example case of gtp1u
/src/gtpv1u/3gpp_29.281.hpp

```cpp
class gtpv1u_msg_header : public stream_serializable {
private:
#define GTPV1U_MSG_HEADER_MIN_SIZE 8
union {
    struct {
        uint8_t pn : 1;
        uint8_t s : 1;
        uint8_t e : 1;
        uint8_t spare : 1;
        uint8_t pt : 1;
        uint8_t version : 3;
    } br;
    uint8_t b;
} ui;
uint8_t message_type;
uint16_t message_length;
uint32_t teld;
uint64_t sequence_number;
uint8_t npdu_number;
uint8_t next_extension_header_type;
};
```

<table>
<thead>
<tr>
<th>Octets</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Version</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Message Type</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length (1st Octet)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Length (2nd Octet)</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tunnel Endpoint Identifier (1st Octet)</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tunnel Endpoint Identifier (2nd Octet)</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tunnel Endpoint Identifier (3rd Octet)</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tunnel Endpoint Identifier (4th Octet)</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sequence Number (1st Octet)</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sequence Number (2nd Octet)</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N-PDU Number</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Next Extension Header Type</td>
</tr>
</tbody>
</table>

NOTE 0: (*) This bit is a spare bit. It shall be sent as '0'. The receiver shall not evaluate this bit.
NOTE 1: 1) This field shall only be evaluated when indicated by the $S$ flag set to 1.
NOTE 2: 2) This field shall only be evaluated when indicated by the $PN$ flag set to 1.
NOTE 3: 3) This field shall only be evaluated when indicated by the $E$ flag set to 1.
NOTE 4: 4) This field shall be present if and only if any one or more of the $S$, $PN$ and $E$ flags are set.

Figure 5.1-1: Outline of the GTP-U Header
Example case of gtp1u

/src/gtpv1u/3gpp_29.281.hpp

- Serialize stream

```cpp
virtual void dump_to (std::ostream &os) {
    ui32 spare = 0;
    os.write(reinterpret_cast<const char*>(&ui3_l), sizeof(ui3_l));
    os.write(reinterpret_cast<const char*>(&message_type), sizeof(message_type));
    auto be_message_length = htonl(message_length);
    os.write(reinterpret_cast<const char*>(&be_message_length), sizeof(be_message_length));
    auto be_tid = htonl(tid);
    os.write(reinterpret_cast<const char*>(&be_tid), sizeof(be_tid));
    if (ui1.bl.s) {
        auto be_sequence_number = htonl(sequence_number);
        os.write(reinterpret_cast<const char*>(&be_sequence_number), sizeof(be_sequence_number));
    }
    if (ui1.b & 0x01) {
        os.write(reinterpret_cast<const char*>(npdu_number), sizeof(npdu_number));
        os.write(reinterpret_cast<const char*>(next_extension_header_type), sizeof(next_extension_header_type));
    }
}
```

- Deserialize stream

```cpp
virtual void load_from (std::istream &is) {
    // Load data from stream
    // ...
Example case of AMF Reselection

3GPP TS 23.502, R16.0.0., Section 4.2.2.2.3

Slice Selection during Registration with AMF re-allocation
[When requested NSSAI not present in subscribed NSSAI]

3GPP TS 23.502, R16.0.0., Section 4.2.2.2.3

Slice Selection during registration
[AMF Reselection- When requested NSSAI is not supported by AMF]
Call Flow - AMF Reselection

AMF to NSSF

Nssf_NSSelection (GET)

Nssf_NSSelection (RESPONSE 200 OK)

Authorized_Network_Slice_Info

3GPP TS 29.531, R16.0.0., 5.2.2.2.2

https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-nssf/-/blob/master/docs/oai_cn5g_nssf_dev_guide.pdf
UML diagram - AMF Reselection (nnnsf_get)

URI query parameters

- M: nf-Type: NF_Type (string)
- M: nf-id: Nf_InstanceID (string)
- C: slice-info-request-for-registration: Slice_info_for_registration_t
- C: home-plmn-id: PlmnId_t
- C: tai: Tai_t
- C: supported-features: Supported_Features (string)