Implementation of the CU/DU split functionality over OAI

N. Makris, P. Basaras, T. Korakis
University of Thessaly & CERTH

N. Nikaein
Eurecom
Network Implementation Testbed Laboratory

- 5 faculty members
- 30 researchers (research engineers, postdocs, PhD students, master students)
- Research activities in the field of wired and wireless networking, cloud, smart cities
- Strong participation in EU projects
- Website: nitlab.inf.uth.gr
NITOS Wireless Testbed

- Developed and operate NITOS, a research experimental facility that supports the research activity of the lab in EU level
- NITOS stands for “Network Implementation Testbed using Open Source tools”
- NITOS supports multiple technologies like wireless (Wi-Fi, 4G, mmWave, SDR), wired networks, SDN/NFV, cloud, sensors)
- NITOS offers over 100 nodes for wireless experimentation available in indoor/outdoor setups
- NITOS is the main testbed facility of multiple EU projects (Onelab, Openlab, FIBRE, Content, SmartFIRE, Fed4FIRE, FLEX, 5G-Xhaul, 5G-PICTURE)
- NITOS is constantly upgraded with beyond the state-of-the-art hardware and software
CU/DU split use cases

- Currently involved in the CU/DU split over the OAI stack
- Using as a reference stack the LTE stack
- Develop a protocol for the full communication between the CU and DU units of the network
- Target use cases:
  - multi-RAT behaviour (e.g. 5G/4G/WiFi)
  - multi-vendor application (e.g. OAI CU and third party DUs)
  - modelling of the midhaul/fronthaul network
  - multi-tier splits (e.g. CU/DU/RRU)
Centralized Units – Distributed Units

- Based on the current drafts for NG-RAN
- ETSI TS 38.470-475
- F1 interface introduced for the communication between CU and DUs
- F1AP protocol running over the F1 interface
F1AP procedures

5G-Core

F1-U

5G-UE

gNB-CU

NAS IP PDCP

GTP UDP IP Data link Physical

pdcp_data_req() pdcp_data_ind()

F1-C

F1-C packet SCTP IP Data link Physical

UE

RRC Procedures

gNB-DU

RLC MAC PHY

NG
Implementation details

- Implementation of the DL User Data packet for F1-U
- Using Google Protocol Buffers

<table>
<thead>
<tr>
<th>Bits</th>
<th>PDU Type (=0)</th>
<th>spare</th>
<th>F1-U Sequence Number</th>
<th>Data</th>
<th>Spare extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>0-4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Using Google Protocol Buffers

<table>
<thead>
<tr>
<th>Bits</th>
<th>Octet of Num</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

F1-U Sequence Number: 3 (FFS)
DL-User-Data packet

✓ We piggyback information that is traditionally exchanged between the PDCP and RLC
  ✓ Information to build the protocol context (e.g. RNTI, SRB, frame, subframe, lcid)
  ✓ Replicate the PDCP Hash table to the RLC layer and update it accordingly with the values that we receive over the fronthaul/midhaul interface
  ✓ For accessing the fronthaul/midhaul interface, we use the asynchronous interface (TCP, we have extensions for UDP, SCTP) – no GTP for the moment
  ✓ Config file for getting initial values (e.g. CU/DU address)
  ✓ One to many relationship between CU/DU
DL User Data packets

Target Architecture (Step 2)  Current Architecture (Step 0/1)

DL USER DATA format

<table>
<thead>
<tr>
<th>Bits</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PDU Type (=0)</td>
<td>spare</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F1-U Sequence Number

- frame
- subframe
- RNTI
- SRB/DRB/LCID
- Spare extension

(FFS)
Developed the DL Data Delivery Status

Will be used for the transferring of UL information?

Piggyback the information needed to update the PDCP entity with the received information

Branch: feature-127-protocol-split
Next steps for the implementation

- Develop the respective interfaces for the F1-C packets
- Replicate the appropriate structures that are needed from PDCP to RLC and vice-versa (e.g. PDCP/RLC hash tables)
- Transfer the RRC messages over the F1-C packets
- Handle them transparently at the DU side if are intended for MAC
- Fully decouple the stack to two separate binaries
Similar Work – F1oIP

- Introduced a protocol for the similar intercommunication between CU/DUs
Experimental Setup

Experimental Setup Diagram:

- NITOS NODE
  - Iperf application
  - OAI-CN

- NITOS NODE
  - OAI-CU
  - Latency Emulation
  - OAI-LTE-DU

- NITOS NODE
  - WIFI-DU
  - Scapy injection
  - Ath9k driver

- NITOS NODE
  - Iperf application
  - LTE Client
  - WiFi Client

Institute Affiliations:

- CERTH, Greece
- Eurecom, France
- Yale Institute for Network Science, USA
Policy Evaluation

Experimental Evaluation of Functional Splits for 5G Cloud-RANs

Nikos Makris, Pavlos Basaras, Thanasis Korakis, ... Greece
Informatics & Telematics Institute (CERTH), Greece
Eurecom, France
Yale Institute for Network Science, USA

Policies evaluation for LTE/WiFi DUs (UDP Midhaul)

Policies evaluation for LTE/WiFi DUs (TCP Midhaul)
Delay Results

Evaluation for UDP Midhaul vs delay

Evaluation for TCP Midhaul vs delay
Thank you for your attention!

Nikos Makris (UTH & CERTH)
nimakris@iti.gr, nimakris@uth.gr