Implementation of the CU/DU split functionality over OAI

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

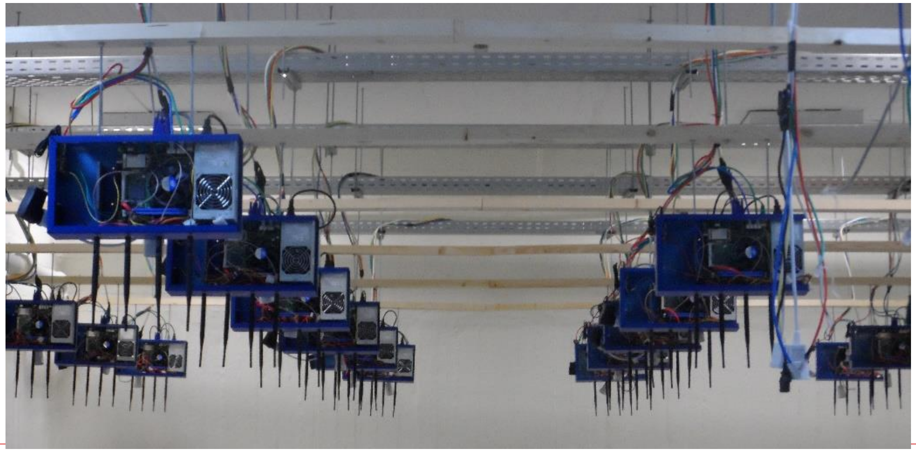
N. Nikaein
Eurecom

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

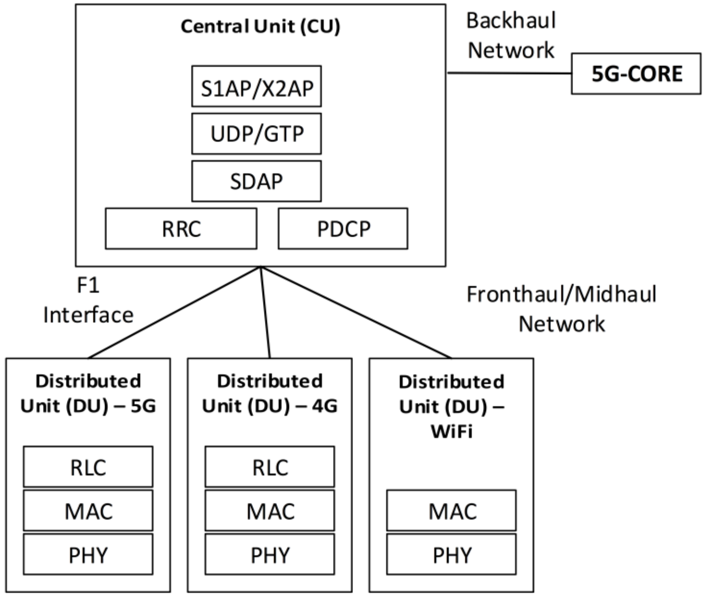
- ✓ Developed and operate NITOS, a research experimental facility that supports the research activity of the lab in EU level
- ✓ NITOS stands for "Network Implementation Testbed using Open Source tools"
- ✓ NITOS supports multiple technologies like wireless (Wi-Fi, 4G, mmWave, SDR), wired networks, SDN/NFV, 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



NITOS Indoor Testbed

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

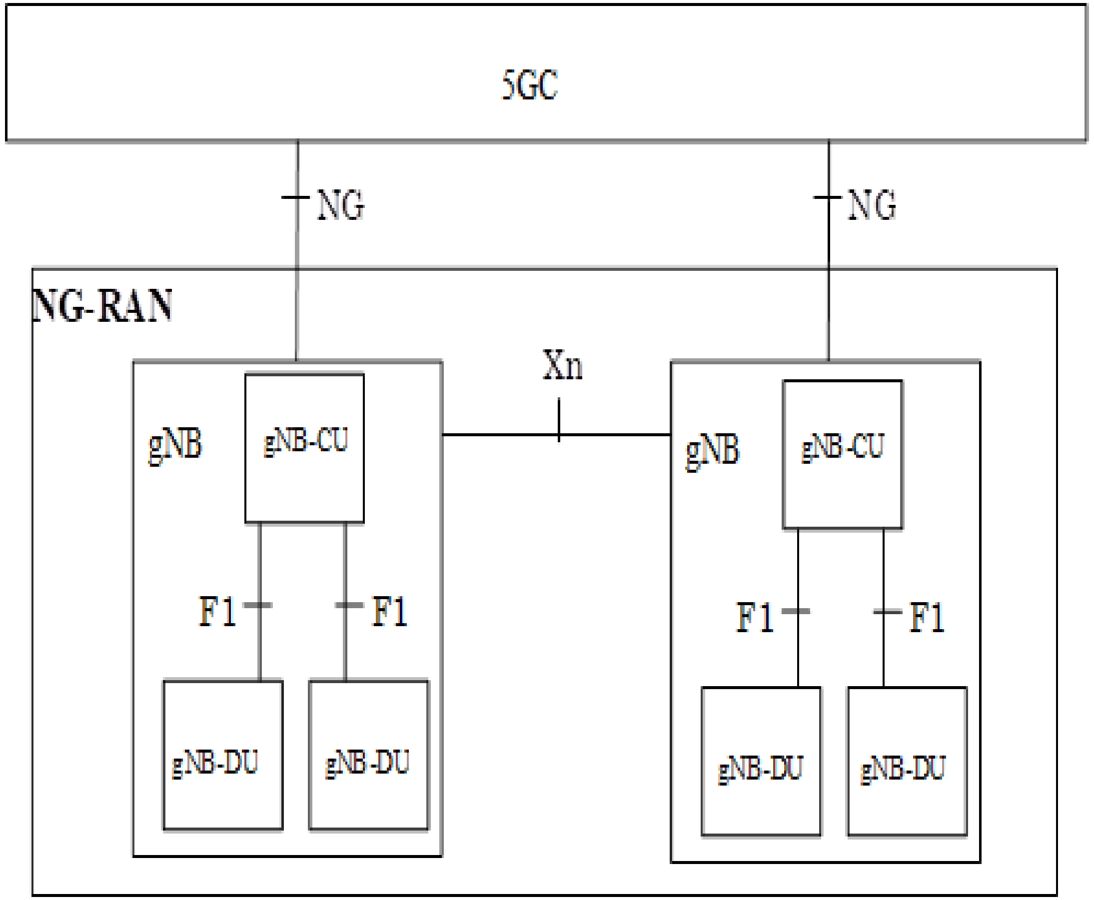
- ✓ Based on the current drafts for NG-RAN
- ✓ ETSI TS 38.470-475
- ✓ F1 interface introduced for the communication between CU and DUs
- ✓ F1AP protocol running over the F1 interface

gNB-CU

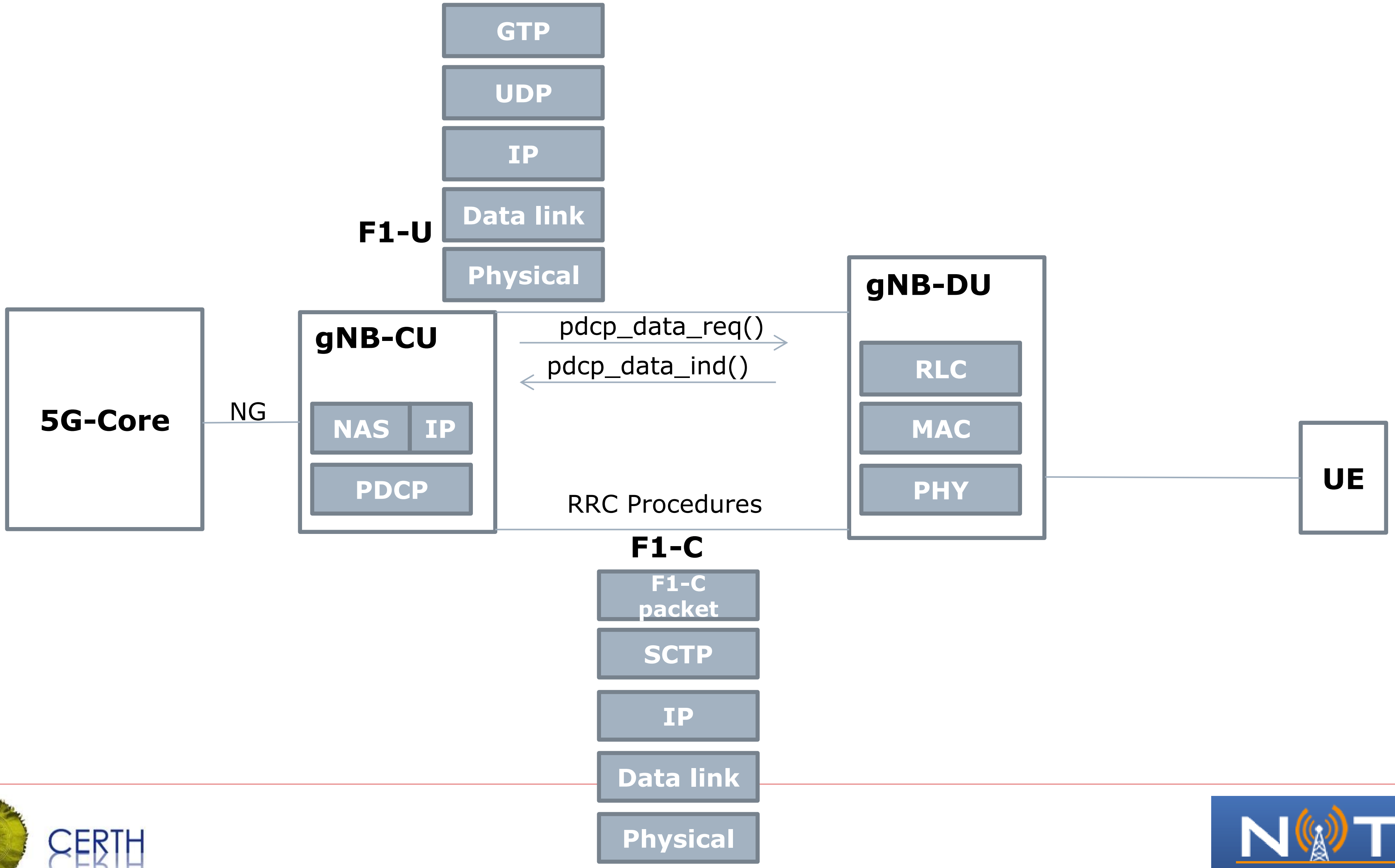
RRC
SDAP
PDCP

gNB-DU

RLC
MAC
PHY



F1AP procedures



Implementation details

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

Bits								Number of Octets
7	6	5	4	3	2	1	0	
PDU Type (=0)				spare				1
F1-U Sequence Number								3 (FFS)
Data								
Spare extension								0-4

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)

- **F1-U packet**
- **GTP**
- **UDP**
- **IP**
- **Data link**
- **Physical**

Current Architecture (Step 0/1)

- **F1-U packet + protocol context info**
- **UDP**
- **IP**
- **Data link**
- **Physical**

DL USER DATA format

Bits								Number of Octets
7	6	5	4	3	2	1	0	
PDU Type (=0)				spare				1
F1-U Sequence Number								3 (FFS)
frame								16
subframe								
RNTI								
SRB/DRB/LCID								
Spare extension								0-4

DL Data Delivery Status packet

- ✓ 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

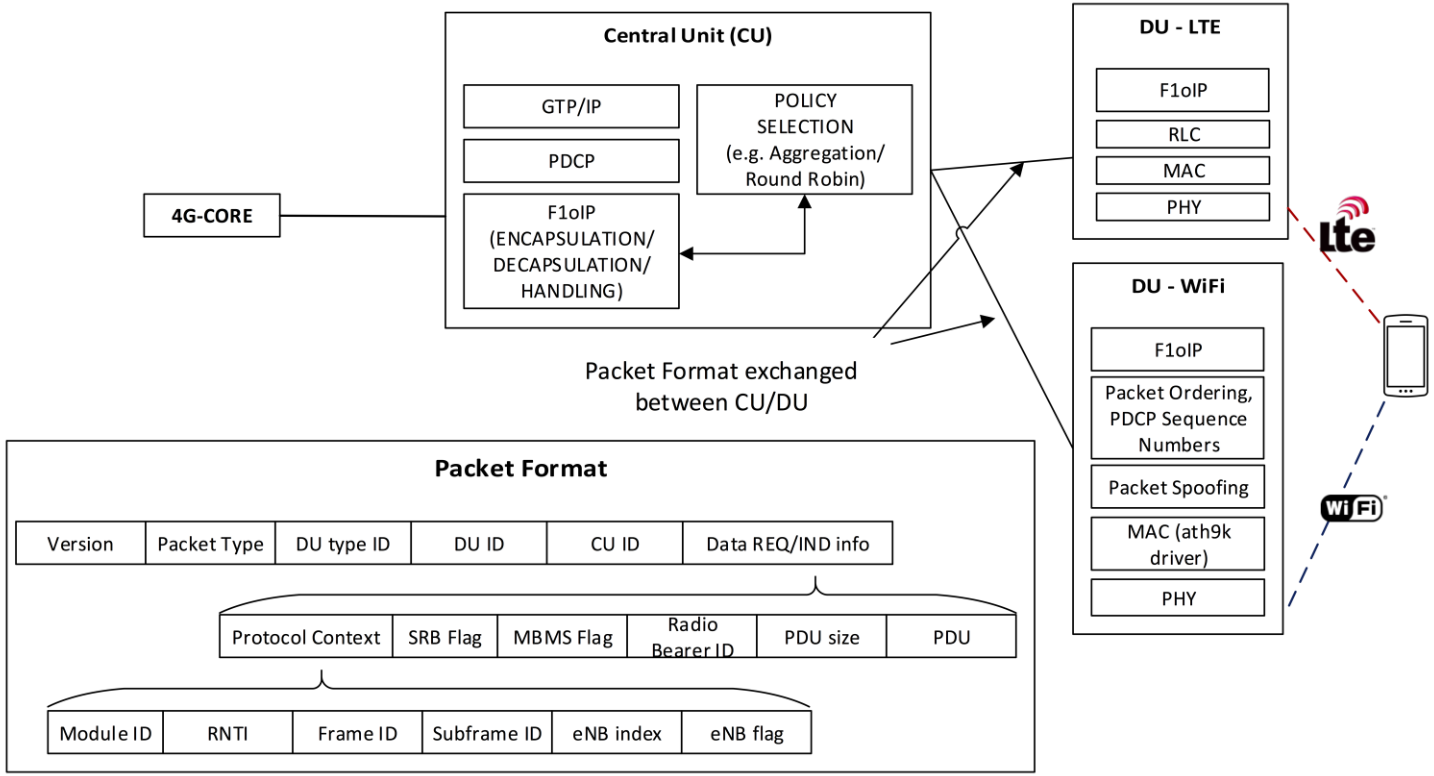
Bits								Number of Octets
7	6	5	4	3	2	1	0	
PDU Type (=1)				Spare		Final Frame Ind.	Lost Packet Report	1
Highest successfully delivered PDCP Sequence Number								3(FFS)
Desired buffer size for the data radio bearer								4
Minimum desired buffer size for the UE								4
Number of lost F1-U Sequence Number ranges reported								1
Start of lost F1-U Sequence Number range								4*(Number of reported lost F1-u SN ranges) (FFS)
End of lost F1-U Sequence Number range								
Spare extension								0-4

Next steps for the implementation

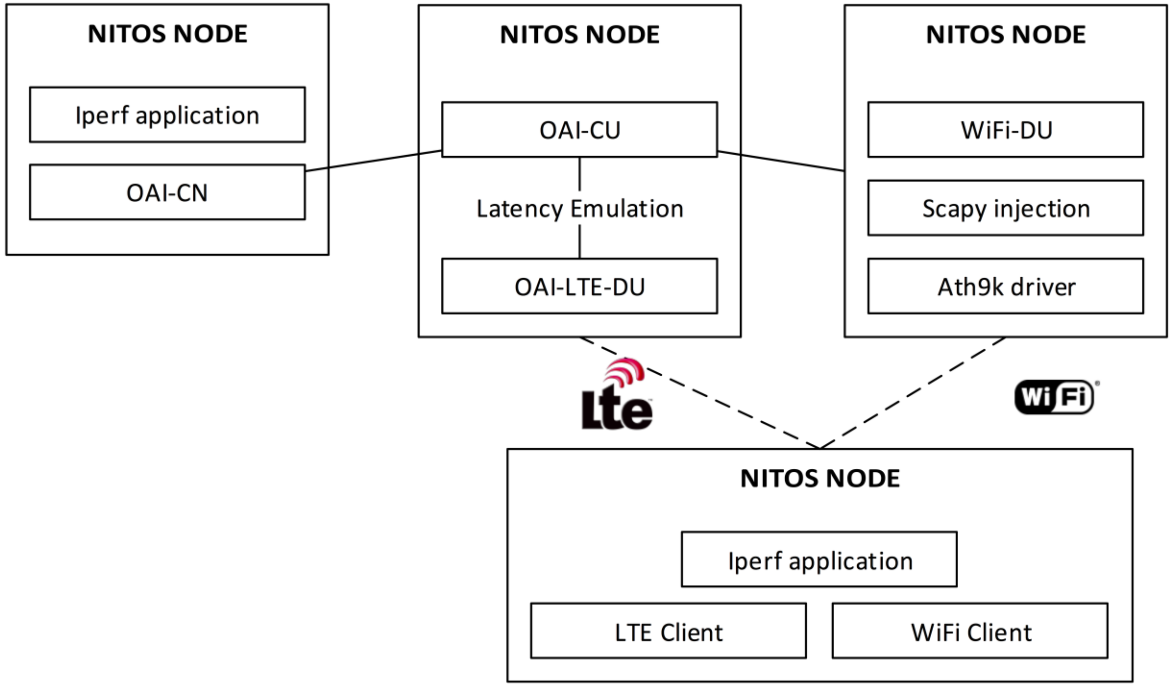
- ✓ Develop the respective interfaces for the F1-C packets
- ✓ Replicate the appropriate structures that are needed from PDCP to RLC and vice-versa (e.g. PDCP/RLC hash tables)
- ✓ Transfer the RRC messages over the F1-C packets
- ✓ Handle them transparently at the DU side if are intended for MAC
- ✓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

Similar Work – F1oIP

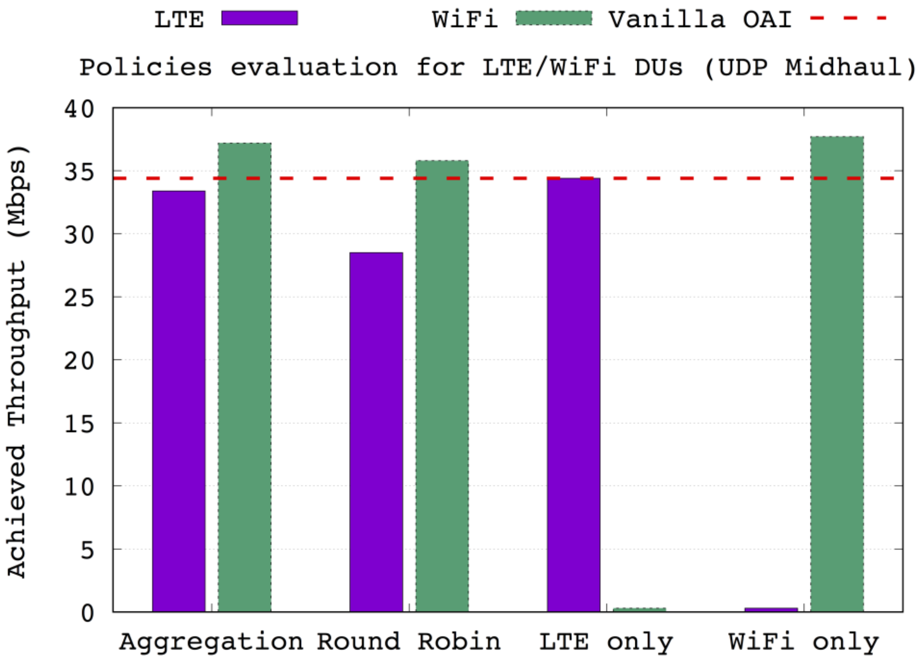
✓ Introduced a protocol for the similar intercommunication between CU/DUs

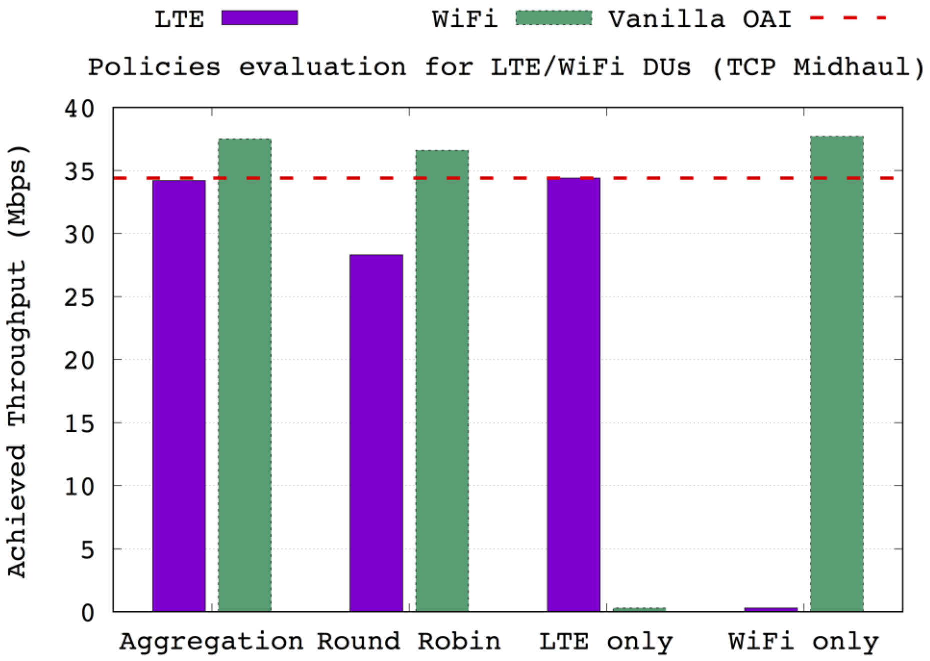


Experimental Setup

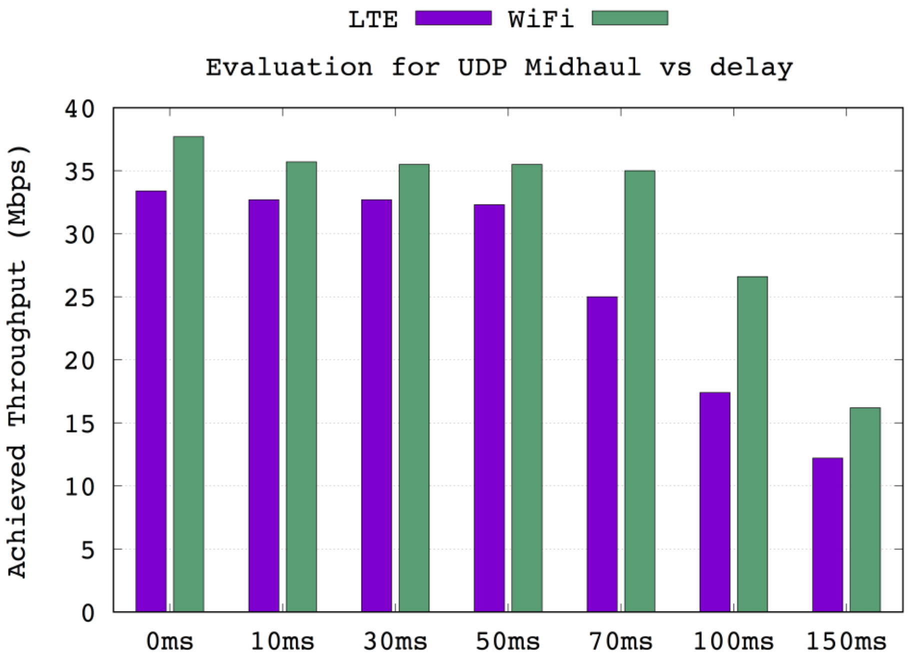


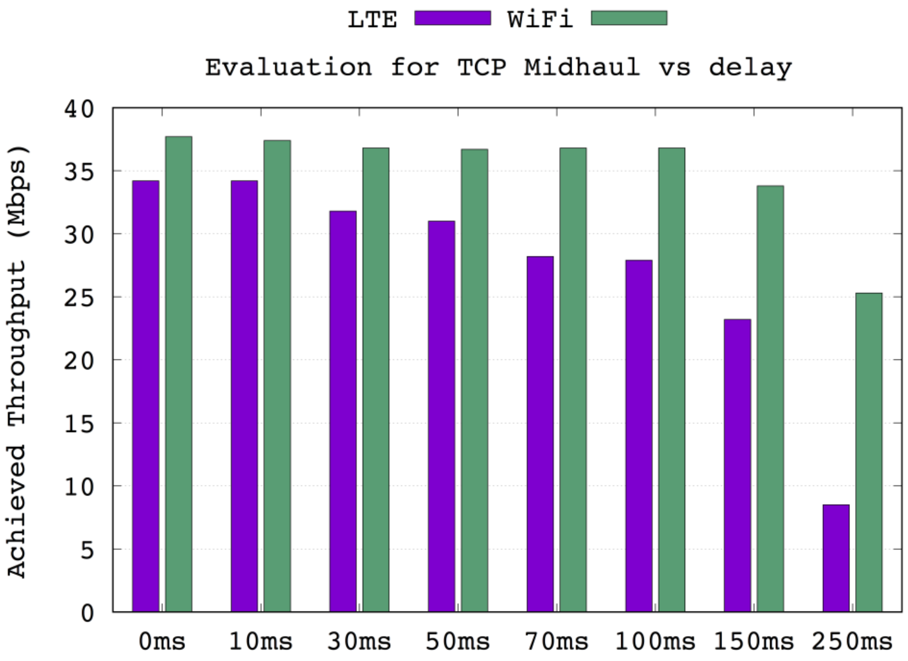
Policy Evaluation





Delay Results





UNIVERSITY OF THESSALY

CERTH

NETWORK IMPLEMENTATION TESTBED LABORATORY

Thank you for your attention!

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

UNIVERSITY OF THESSALY

CERTH
CENTRE FOR RESEARCH & TECHNOLOGY HELLAS