iJOIN

Internal Report IR5.2

IR5.2: Revised Definition of iJOIN Architecture

Editors:	Peter Rost, Andreas Maeder, NEC
Deliverable nature:	Confidential
Suggested readers:	iJOIN GA
Due date:	June 30th, 2014
Delivery date:	July 14th, 2014
Version:	1.0
Total number of pages:	90
Reviewed by:	GA members
Keywords:	iJOIN
Resources consumed	21 PM

Abstract

This report provides an update on the iJOIN architecture including logical architecture, functional architecture, and physical architecture. Particular focus is given to the functional architecture and the interaction of individual technologies, their impact on the system performance, and how they interact. Furthermore, the physical architecture is comprehensively described. For each common scenario, a physical architecture is provided which shows how the logical entities and interfaces are mapped to physical entities and physical interfaces. Based on this, a detailed simulation campaign is prepared. In addition, this report provides an update on the functional split as well as joint RAN/BH operation, required interfaces, constraints, and preliminary results.

List of authors

Company	Author
CEA	Valentin Savin, Antonio de Domenico, Matteo Gorgoglione
HP	Marco Consonni, Marco Di Girolamo
IMC	Umer Salim
IMDEA	Luca Cominardi, Albert Banchs
NEC	Peter Rost, Andreas Maeder
SCBB	Massinissa Lalam
TI	Dario Sabella, Marco Caretti
TID	Ignacio Berberana
TUD	Vinay Suryaprakash, Jens Bartelt
UC3M	Carlos J. Bernardos, Antonio de la Oliva
UNIS	Emmanouil Pateromichelakis
UoB	Dirk Wübben

History

Modified by	Date	Version	Comments
Andreas Maeder	11.07.2014	1.0	Final version

Table of Contents

List of Figures

List of Tables

Abbreviations

3GPP	3rd Generation Partnership Project
ADSL	Asymmetric Digital Subscriber Line
AM	Acknowledged Mode
AMM	Anchoring and Mobility Management
API	Application Programming Interface
APN	Access Point Name
ARM	Advanced RISC Machines
ARQ	Automatic Repeat Request
ASIC	Application Specific Integrated Circuit
ATM	Asynchroneous Transfer Mode
BB	Base Band
BH	Backhaul
CAPEX	Capital Expenditures
CAS	Computational Aware Scheduler
CDF	Cumulative Distribution Function
CF	Compress and Forward
CoMP	Cooperative Multi-Point
CPE	Customer Premises Equipment
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
C-RAN	Centralized RAN
CS	Common Scenario
CSI	Channel State Information
CT	Candidate Technology
DCI	Downlink Control Information
DFT	Discrete Fourier Transform
DHCP	Dynamic Host Configuration Protocol
DL	Downlink
DSP	Digital Signal Processor
ECN	Early Congestion Notification
EE	Energy Efficiency
eNB	Evolved Node B
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FDD	Frequency Division Duplexing
FEC	Forward Error Correction
FER	Frame Error Rate
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GOPS	Giga Operations per Second
GFLOPS	Giga Floating Point Operations per Second
GPON	Gigabit Passive Optical Networking
GPP	General Purpose Processor
GWCN	Gateway Core Network
HARQ	Hybrid Automatic Repeat Request
IaaS	Infrastructure as a Service
IFFT	Inverse Fast Fourier Transform
iLGW	iJOIN Local Gateway
iNC	iJOIN Network Controller
IP	Internet Protocol
ISA	Instruction Set Architecture
iSC	iJOIN Small Cell
ISS	Industry Standard Server
IT	Information Technology

iTN	iJOIN Transport Node
ITU	International Telecommunications Unit
iveC	iJOIN virtual eNodeB Controler
JT	Joint transmission
KPI	Key Performance Indicator
KSR	Kendall Square Research
KVM	Kernel-based Virtual Machine
LOS	Line-of-Sight
LTE	Long Term Evolution
MAC	Medium Access Control
MARC	Multiple Access Relay Channel
MCS	Modulation and Coding Scheme
MIMO	Multiple Input Multiple Output
MIPS	Microprocessor without Interlocked Pipeline Stages
MME	Mobility Management Entity
MOCN	Multi-operator Core Network
MPI	Message Passing Interface
MPLS	Multi-Protocol Label Switching
MPTD	Multi-Point Turbo Detection
MRS	Maximum Rate Scheduler
MUD	Multi-User Detection
NEO	Network Energy Optimizer
OpenMP	Open Multi-Processing
OpenMPI	Open Message Passing Interface
OPEX	Operational Expenditures
OS	Operating System
OSS	Operations Support System
PA	Power Amplifier
PCP	Priority Code Point
PDCP	Packet Data Convergence Protocol
PDU	Packet Data Unit
PHY	Physical Layer
PLMN	Public Land Mobile Network
POSIX	Portable Operating System Interface for Unix
QCI	QoS Class Indicator
QinQ	Q-in-Q; refers to IEEE 802.11ad
QoE	Quality of Experience
QoS	Quality of Service
OSS	Operation Support System
RAM	Random Access Memory
RAN	Radio Access Network
RANaaS	Radio Access Network as a Service
RAP	Radio Access Point
RAT	Radio Access Technology
RF	Radio Frequency
RLC	Radio Link Control
RNC	Radio Network Controller
RoF	Radio over Fibre
RRC	Radio Resource Control
RRM	Radio Resource Management
RRH	Remote Radio Head
RTT	Round-Trip Time
SA1	System Architecture, WG 1
SDN	Software Defined Networking
S-GW	Serving Gateway
SISODQ	Soft-Input Soft-output Dequantizer
SMP	Symmetric Multi-Processor

SNR	Signal-to-Noise Ratio
SPTD	Single Point Turbo Detection
TCO	Total Cost of Ownership
TCP	Transport Control Protocol
UE	User Equipment
UK	United Kingdom
UL	Uplink
UM	Unacknowledged Mode
vCPU	virtual Central Processing Unit
veNB	virtual evolved Node B
VM	Virtual Machine
VPN	Virtual Private Networks
WG	Working Group
WP	Work Package

1 Introduction

The iJOIN project aims at providing a solution for heterogeneous small-cell based networks to incorporate partially centralized radio access network functionality. This centralization will improve the radio access performance through advanced processing such as joint transmission and reception. It will further improve the energy-efficiency through pooling gains at the central processor. In addition, the usage of a central processor based on commodity hardware will improve the cost-efficiency. Exploiting multi-user, traffic, and computational diversity further allows for improved utilization efficiency.

In order to implement the iJOIN vision, two main innovations need to be further developed, i.e. flexible functional split and joint RAN/BH operation and design. The previous deliverable D5.1 [4] provided a basic understanding for the requirements of both technologies and how the iJOIN architecture must be designed in order to allow for an efficient evolution towards the iJOIN system. In this report, the actual implementation of both innovations is put in focus. The main challenges for an implementation of the flexible functional split are the question for the right usage of different hardware options, how a RAN can be implemented in a virtualized environment, how 3GPP LTE RAN constraints impact the implementation of the functional split, and which functional splits should be preferred. In the case of joint RAN/BH operation, the main challenge is the definition of interfaces and to clarify how the individual components in RAN and backhaul will interact.

The iJOIN project further introduces a set of novel technologies which improve the individual performance objectives energy-efficiency, cost-efficiency, utilization-efficiency, and area throughput. While each technology on its own may improve the performance, it may also impact other technologies and deteriorate or emphasize their improvements. Hence, it is important to understand how these technologies interact, whether they are complementary and contradicting, how their gains will be combined, and how they are integrated in the two main concepts functional split and joint RAN/BH operation. This harmonization work is the main task of work package 5 and will use the output from work packages 2, 3, and 4 where novel technologies for physical layer, medium access and radio resource control layer, and for the network operation are derived, respectively. This report is the first step towards this harmonization.

At the end of the project, a comprehensive and consistent evaluation of the iJOIN system performance will be provided. In order to avoid loosely coupled results from individual candidate technologies, a harmonized simulation campaign is required. This could be achieved through different means, e.g. a joint simulation effort where all partners apply the same simulation framework, a joint calibration effort where all partners calibrate their individual simulation tools, or a joint parameter derivation where all partners apply the same set of parameters to both the novel technologies and the baseline system. In iJOIN, the last option has been chosen due to resource constraints. The first two options require substantial resources. In the case of iJOIN, relevant parameters for each main scenario are derived and all candidate technologies incorporate these parameters. In a next step, for each of these parameters a range of meaningful values has been defined and is used for the evaluation of each candidate technology. The comparison of candidate technologies is done based on a relative basis, i.e. each technology is compared to the baseline system and then relative gains across multiple candidate technologies are compared.

2 Summary and Contributions

In this report, the iJOIN project provides an update to the previous deliverable D5.1 [4]. The focus of D5.1 has been on a comprehensive analysis of the state-of-the-art and a detailed derivation of the iJOIN architecture including a logical, functional, and physical architecture. In this report, we focus rather on the two core innovations of iJOIN, i.e. functional split and joint RAN/BH operation. In addition, we detail how iJOIN will perform a project-wide evaluation of novel technologies.

2.1 Summary

The first part of this report, Section 3, provides an update of the logical iJOIN architecture which details how the logical entities are connected and which logical interfaces are considered. In comparison to D5.1, only minor changes have been applied as the focus has been on the implementation and application of the logical architecture. Further, Section 3 provides a summary of candidate technologies and which performance objectives are addressed. This is important in order to determine whether individual technologies may co-exist, whether they complement each other, and whether their gains add up. We further provide a summary of the impact of each candidate technology. Afterwards, a summary of main scenarios and their physical architecture is provided which is important to determine how the logical interfaces map to physical constraints and requirements. This allows for characterizing the individual logical interfaces. Finally, Section 3 gives a detailed overview how network sharing impacts the iJOIN architecture and how iJOIN facilitates network sharing, particularly in a small-cell environment.

Section 4 details different aspects of the functional split. First, it provides a comprehensive overview of implementation aspects and how different hardware options impact the implementation of RAN functionality. We further discuss a virtualized infrastructure which may have a significant impact on how algorithms are implemented, how they interact with each other, and how they can be scaled with the RAN. Virtualized environments hide resource constraints efficiently through virtualized interfaces. However, these constraints still need to be considered and shall be exploited in a centralized RAN environment. Load balancing is another option to exploit large-scale computing resources more efficiently. We further detail how well known concepts from cloud-computing will affect the RAN operation, e.g. how migration of virtual machines and therefore virtual eNodeBs can be implemented. Furthermore, constraints originating from computing platforms are linked with constraints originating from the RAN, e.g. latency and throughput requirements. In addition, Section 4 explores preferred splits of RAN functionality and how these splits may be implemented flexibly. In particular, the flexible split of RAN functionality needs to consider practical constraints in small-cell networks.

In Section 5, this report focuses on the second main iJOIN innovation, i.e. joint RAN and BH operation. First, this part elaborates on required interfaces which are required to support joint RAN/BH technologies. This is described individually for all candidate technologies and their interaction with other logical entities is explained. In addition, results for joint RAN/BH coding are provided to exemplify how joint RAN/BH operation can improve the system performance.

Finally, Section 6 defines main objectives and metrics which are applied in iJOIN, i.e. energy-efficiency, cost-efficiency, utilization-efficiency, and area throughput. In particular, the first three metrics were updated in order to provide a consistent framework for the final evaluation towards the end of the project. Also in addition to the previous report, a detailed overview of the performance evaluation campaign is provided. This report derives a set of parameters for each evaluation scenario which is supported by all candidate technologies which are applied to the respective scenario. For that, a minimum set of parameters has been derived and a meaningful range of parameter values has been defined.

2.2 Key Contributions

This report provides a major update compared to the previous report D5.1. Firstly, the functional architecture definition progressed significantly such that a first assessment of interaction across candidate technologies and work packages is provided. This includes the description of how candidate technologies impact different objectives, which is essential to perform a project-wide assessment at the end of the project.

This report further provides a detailed analysis of the functional split, including implementation aspects resulting from different hardware options, impact of virtualized infrastructure, and how data processing

complexity can be measured. Analytical results show how data processing complexity in a 3GPP LTE RAN system scales, how the centralization gain is reflected, and how the central processor can be dimensioned.

Furthermore, a comprehensive overview of functional splits and practical constraints of a flexible implementation are detailed. Constraints from the cloud-computing platform and 3GPP LTE RAN are related through the required data processing capabilities for a given quality of service. Analytical results for 3GPP LTE RAN show that latency constraints at the interface of physical and medium access layer can be efficiently mitigated without performance loss. Also, results for joint RAN/BH coding show how coding across both domains, distributed IP anchoring, and network-wide energy optimization can improve the system-performance.

Finally, this report provides a detailed description of the simulation campaign, the individual objectives and of how they are going to be measured. Furthermore, first results for the utilization efficiency of cloud-RAN with higher layer split are provided.

3 iJOIN Architecture

3.1 Logical Architecture

In the past, the trend was to push the computation burden toward the last miles in order to reduce the round trip time and improve the reactivity of the system (e.g., see ARQ vs HARQ). With dense small cell deployments being a promising solution to answer the growing need of capacity, (partial) centralization is required to deal with complex interference situations. To cope with upcoming dense deployments of LTE-based small cells, iJOIN has proposed an evolutionary architecture in its deliverable D5.1 [4] which is shown in Figure 3-1.

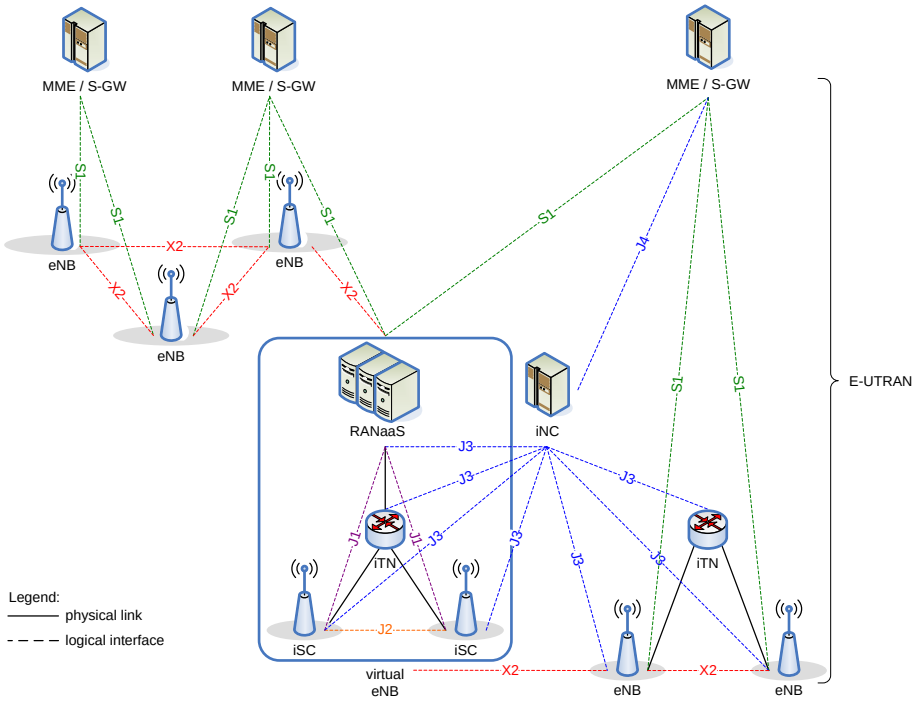


Figure 3-1: iJOIN Architecture

In this architecture, the RAN-as-a-Service (RANaaS) concept is introduced for dense small cell (iSC) deployments. The purpose of the RANaaS concept, based at a first glance on generic cloud computing platforms (see Section 4 for more insights), is to enable some RAN functionalities to be centrally executed on demand. This will allow for benefiting from centralization gains which will be of critical importance in dense small cell deployment. In practice, the RANaaS and iSCs entities appear as classical eNBs (3GPP terminology for LTE base stations) to the existing network. Therefore, such "virtual eNB" (veNB) entity can be seamlessly integrated in the existing architecture. The core network does not need to know that the RAN functionalities are effectively split between iSCs and RANaaS. It only needs to know where to forward/get user and control planes, which by default will be the RANaaS entity.

One veNB comprises the RANaaS instance running on a cloud platform (veNB upper domain) and one or several iSCs (veNB lower domain). Within one veNB, the iSCs and the RANaaS platform are connected through the J1 interface, while the iSCs can exchange information directly with each other using the J2 interface. Comparable to a legacy eNB, one veNB can setup one X2 connection with other (v)eNBs supporting the exchange of standardised 3GPP signals.

In order to optimize jointly the RAN part and the backhaul, a Network Controller (iNC) entity is also added to configure the routing among the backhaul Transport Node (iTNs) based on configurable constraints (e.g., RAN/backhaul load, user density, or mobility pattern). As illustrated, this software defined network (SDN)-based controller solution can be applied to the proposed RANaaS/iSCs setup or to a classical LTE deployment. To enable this function, the iNC will rely on the J3 interface connecting each of the involved entities as depicted in Figure 3-1. In addition, the iNC will also be able to dialog with the core network through the J4 interface for the purpose of routing, anchoring, and mobility.

3.2 Functional Architecture

The functional architecture defines the interaction of functional blocks implemented in the iJOIN architecture. In particular, the functional architecture defines the required input information for candidate technologies (CTs) and defines the output information provided by CTs. Furthermore, it allows for identifying the interaction across work packages (WPs) within the project. The deliverables D2.1 [35], D3.1 [26], and D4.1 [36] provide a detailed overview of the functional architecture from each individual WP perspective. In this section, we provide a brief summary of the functional architecture considered by each WP.

In iJOIN, four objectives are of particular interest: energy efficiency, area throughput, cost efficiency, and utilization efficiency. Each candidate technology aims at optimizing one particular objective. However, in addition, each candidate technology may also impact other objectives. Co-deploying multiple candidate technologies therefore may results in a set of contrary effects on multiple objectives.

The subsequent table shows the main objectives (**X**) and the potentially affected objectives (+) for each candidate technology.

Table 3-1: Main objectives for each CT

CT	Energy Efficiency	Area Throughput	Cost Efficiency	Utilization Efficiency
CT2.1	+	**X**		+
CT2.2		**X**		+
CT2.3		**X**		
CT2.4	+	**X**		
CT2.5		**X**		
CT2.6		**X**	+	
CT2.7	+	**X**	+	
CT3.1		**X**		
CT3.2		**X**		
CT3.3	**X**			
CT3.4		**X**		
CT3.5		**X**		
CT3.6				**X**
CT3.7		**X**		+
CT3.8		**X**		
CT3.9		**X**		
CT4.1				**X**
CT4.2	**X**			
CT4.3			**X**	**X**
CT4.4			+	**X**
CT4.5				**X**
CT4.6	+		**X**	

3.2.1 Interaction of CTs

In this subsection we detail the interaction of each candidate technology with other candidate technologies within the same WP. The description of the interactions across work packages will be harmonized in D5.2.

3.2.1.1 Work package 2 CTs

CT2.1 In-Network Processing

CT2.1 "In-Network Processing" interacts with CT3.8 "Radio Resource Management for In-Network Processing". CT2.1 defines the PHY processing for distributed detection in the uplink, whereas CT3.8 derives adapted RRM schemes that allow for scheduling several users on the same resource elements depending on the capability of the PHY processing. Interaction with other WP2 CTs:

- CT2.2 is an alternative approach for joint multi-user detection (MUD) in the uplink.
- CT2.3 is an alternative approach for cooperative detection of user signals in the uplink assuming orthogonal resource allocation.
- CT2.4 and CT2.5 are downlink approaches and can, thus, be implemented together with CT2.1.
- The functional split investigation in CT2.6 serves as an enabling technology for CT2.1
- The joint RAN/BH optimization and the mmWave transmission considered in CT2.7 serve as enabling technology for the used backhaul links between the iSCs.

CT2.2 Multipoint Turbo Detection

CT2.2 "MPTD" interacts directly with CT3.7 "scalable RRM for MPTD", the former dealing with the PHY processing, while the latter performs the RRM operation. Interactions with other WP2 CTs:

- CT2.1 is an alternative approach for joint multi-user detection in the uplink. Both CTs target uplink detection.
- CT2.3 is an alternative approach for cooperative detection of user signals in the uplink assuming orthogonal resource allocation.
- CT2.2 being uplink-oriented, it could be implemented together with CT2.4 and CT2.5 which are downlink-oriented (no side effect so far).
- CT2.2 can be implemented together with CT2.6 and CT2.7 as those CTs are dealing with backhaul links (no side effect so far).

Interactions with WP3:

- CT2.2 could be implemented together with CT3.1 "Backhaul Link Scheduling and QoS-aware Flow Forwarding", since CT3.1 deals with backhaul routing to the core network essentially. CT3.1 would be used to route traffic from users not involved in an MPTD processing (no side effect so far)
- CT2.2 could be implemented with CT3.2 "Partly decentralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments", since CT3.2 deals with cell (re) selection mechanisms. CT2.2 assumes the selection is done, while CT3.1 will act on the selection before CT2.2 has to be applied (no side effect so far).
- CT2.2 may not be compatible with CT3.3 "Energy-Efficient MAC/RRM at Access and Backhaul" which deals with discontinuous transmission of iSCs in the downlink. In CT2.2 the uplink is considered, but an acknowledgement from the downlink is always expected. If CT3.3 decided to cancel such acknowledgement, then CT2.2 may not operate properly if no coordination between CTs is performed.
- CT2.2 may not be implemented with CT3.4 "Computational Complexity and Semi-Deterministic Scheduling", performed at each iSC. CT2.2 works jointly with CT3.7 which is also a centralised RRM CT and is a "concurrent" algorithm of CT 3.4. Ideally if CT3.4 only deals with user equipments (UEs) not involved in MPTD, while CT3.7 operates on those specific UEs, then CT2.2 could be implemented together with CT3.4.
- CT2.2 could be implemented with CT3.5 "Cooperative RRM for Inter-Cell Interference Coordination in RANaaS" which deals with downlink RRM (no side effect so far).
- CT2.2 could be implemented with CT3.6 "Utilization and Energy Efficiency" which evaluates those metrics within the iJOIN context (no side effect so far).

- CT2.2 is a direct match to CT3.7 “Radio Resource Management for Scalable Multi-Point Turbo Detection”
- CT2.2 may not be compatible with CT3.8 “Radio Resource Management for In-Network-Processing”, which is the RRM part of CT2.1. Since CT2.2 and CT2.1 are alternative approaches, CT2.2 will not use CT3.8 output.
- CT2.2 could be implemented with CT3.9 “Hybrid local-cloud-based user scheduling for interference control” which deals with scheduling in the downlink, while CT2.2 operates in the uplink (no side effect so far).

There are no particular interactions with WP4 identified yet.

CT2.3 Joint Network-Channel Coding

CT2.3 “Joint Network-Channel Coding” interacts with CT3.2 “Partly Decentralized Mechanisms for Joint RAN and Backhaul Optimization in Dense Small Cell Deployments”. CT3.2 separates the small cell deployment within one veNB in a number of Multiple-Access Relay Channels (MARCs), while CT2.3 defines the joint network-channel coding strategy in the uplink for the MARC.

- CT2.1 and CT2.2 are alternative approaches for uplink detection, where several users transmit using the same physical resources
- CT2.4 and CT2.5 are downlink-oriented and do not affect CT2.3
- CT2.6 and CT2.7 are enabling technologies for the backhaul link.

CT2.4 Sum-Rate and Energy-Efficiency Metrics of DL COMP with backhaul constraints

CT2.4 “Sum-Rate and Energy-Efficiency Metrics of DL COMP with backhaul constraints” implements the downlink achievable rate based on compress-and-forward relay scheme by optimizing the compression rate at each iSC. It is compatible with all other uplink CTs and backhauling techniques such as CT 2.7. The CT does not consider the precoding at iSC, however, it is extendable to the case with precoding at iSCs and therefore it is potentially compatible with CT2.5.

CT2.5 Partially Centralized Inter-Cell Interference Coordination

This CT studies the precoding with a hybrid architecture composed of both J1 and J2 links. It is compatible with all CTs focusing on the uplink and with the backhaul techniques. CT2.5 studies a similar problem as CT2.4, thus, it potentially interoperates with CT2.4.

CT2.6 Data Compression over RoF

CT2.6 “Data Compression over RoF” is based on a specific functional split of the PHY layer between RANaaS and iSCs with the goal of reducing the backhaul load. It can be applied in conjunction with any other CT, operating at both PHY and/or MAC/RRC level, that is compatible with a PHY functional split that entails the execution of the IFFT/FFT operations at the iSCs.

CT2.7 Millimetre wave backhauling

Since CT 2.7 deals with PHY-layer BH and is terminated after channel decoding, it should be compatible with any higher-layer CT. The joint RAN/BH schemes investigated apply only to the uplink, so the CT is also fully compatible with any downlink oriented CT. The compression schemes of CT2.6 are directly applicable to mmWave backhauling as well.

3.2.1.2 Work Package 3 CTs

Table 3-2 provides an overview of the inter-operability of CTs within work package 3. For a detailed discussion of CT interoperability in WP3 see Annex B.

Table 3-2: CT interoperability matrix for WP3

	CT 3.2	CT 3.3	CT 3.4	CT 3.5	CT 3.7	CT 3.8	CT 3.9
CT 3.1	Y	Y	Y	Y	Y	Y	Y
CT 3.2		Y	Y	Y	Y	Y	Y
CT 3.3			Y	Y	Y	Y	Y
CT 3.4				N	N	N	N
CT 3.5					Y	Y	N
CT 3.7						N	Y
CT 3.8							Y

Definitions:

"**Y**" – interoperable: CTs operate on different resources or in different operational domains. Algorithmic coordination and/or information exchange with iJOIN network entities (e.g. iNC or iveC) may be necessary. Different domains/resources include:

- Backhaul:
  - Channel resources (e.g. wireless, wired, …)
  - Link/Routing
- RAN:
  - RF transmission (transmit power)
  - Downlink radio resources
  - Uplink radio resources
  - Cell association

"**N**" – not interoperable: CTs operate on same resources and/or are based on different assumptions

3.2.1.3 Work package 4 CTs

CT4.1 Distributed IP Anchoring and Mobility Management

CT4.1 "Distributed IP Anchoring and Mobility Management" must interact with CT4.3 "Joint Path Management and Topology Control" in order to provide mobility. CT4.3 is an enabling technology for CT4.1. CT4.1 also interacts with CT4.2 "Network Wide Energy Optimisation" in case of decisions about switching on/off physical nodes where an anchor must be reassigned. CT3.2 interacts with CT4.1 since a Mobility Load Balancing command is triggered by the iSC serving the UE when the proposed algorithms decide if the UE must change its serving iSC based on the information in the measurement report provided by the UE.

CT4.2 Network Energy Optimization and CT4.5 Load Balancing and Scheduling

CT4.2: "Network Energy Optimization" tries to move UEs/flows from cellular base stations with low utilization to nearby ones, in order to switch-off underutilized nodes and conserve energy. On the other hand, CT4.5: "Load Balancing and Scheduling" tries to move flows from nodes with high utilization, to nodes with a lower one in order to distribute the traffic evenly in the network, and improve per user performance. Thus, the two CTs can be in conflict in some scenarios, and further consideration needs to be taken in order to tackle these two problems jointly. CT4.2 and 4.5 also interact with CT4.1 "Distributed IP Anchoring and Mobility Management". The latter CT tries to select the optimal anchor for the initial assignment of each UE/flow, while the former two might invoke CT4.1 to force handovers and change the original anchor for energy optimization or load balancing reasons. Thus, one should eventually take into account the potential

interactions between CT4.1 and 4.2/4.5 in order to minimize the amount of conflict and retain the intended performance gains of each module.

CT4.3 Joint Path Management and Topology Control

CTs that require the information of topology must interact with CT4.3 "Joint Path Management and Topology Control". Those CTs include: CT4.1: Distributed IP Anchoring and Mobility Management, CT4.2: Network Wide Energy Optimisation, CT4.4: Routing and Congestion Control Mechanisms, CT4.5: Network Wide Scheduling and Load Balancing. In some cases, CT4.3 may interact with other CTs in order to re-locate RANaaS and re-associate iSCs with the RANaaS: when some physical nodes are switched off/on due to energy management in CT4.2, or, when congestion notification is triggered in CT4.4. CT3.9 interacts also with CT4.3 since it develops new cooperative scheduling algorithms which efficiently exploit any backhaul topology available.

CT4.4 Routing and Congestion Control Mechanisms

CT4.4 Routing and Congestion Control Mechanisms should interact with CT4.3 in order to get information of the path and the functional splits employed for each information flow that traverse the controlled iTN. With respect to other CTs, it should be taken into account the way CT4.4 has to solve congestion issues. Basically, there are two potential sets of mechanisms that can be used: end to end mechanisms, involving the end users or their proxies (like those associated with the use of ECN on TCP), which would not have an impact on the other CTs, and in-net mechanisms, that try to solve congestion changing the characteristics of the information flows or changing routes. The latter mechanisms may have an impact on other CTs, as there may be contradictions on the decisions undertaken by them (e.g., for energy efficiency reasons) with respect to those taken by CT 4.4. It is believed that the latter should have priority over the former, but this is an issue for further study.

CT4.6 Backhaul Analysis based on Viable Metrics and "Cost" Functions using Stochastic Geometry

The main objective of CT 4.6 is cost efficiency. Since it provides a method to estimate deployment costs, it has no side effects on other CTs.

3.2.2 Qualitative Impact

In this subsection we report separately the qualitative impact on iJOIN objectives for each candidate technology.

CT2.1 In-Network Processing

CT2.1 aims to increase the *area throughput* by allowing several users to use the same physical resources to transmit their information and performing distributed MUD of these user signals. Signals are exchanged among iSCs over J2 links for this distributed processing in order to achieve the performance of centralized MUD but with limited traffic on the J1 links. Considering that the J1 link to the RANaaS covers a larger distance compared to the J2 links, the localization of BH traffic to the area of the veNB will potentially improve *utilization efficiency*.

CT2.2 Multipoint Turbo Detection

CT2.2 aims at increasing the *area throughput* by scheduling edge users on the same resources and exploiting the interference created as a source of information for the detection using an iterative approach. Thanks to the detection improvement, *utilisation* of the network may also be improved due to a possible increase of the data sent by the UEs (more bits in the pipe).

CT2.3 Joint Network-Channel Coding

CT2.3 aims at increasing the user *throughput*. It applies to the MARC, comprising two users that communicate with two small cells (one relay and one destination). Since the small cell deployment within one veNB is separated in a number of MARCs (by CT3.2), increasing the users' throughput within the MARC will results in an increase of the overall *area throughput* within the veNB.

CT2.4 Sum-Rate and Energy-Efficiency Metrics of DL COMP with backhaul constraints

CT2.4 performs joint transmission (JT) CoMP to explore the processing capability of the RANaaS platform and/or the backhaul network capabilities to improve the system sum rate. This target is achieved by optimizing the compression rate at each iSC based on the compress-and-forward relay scheme with a constrained backhaul. Correspondingly, this CT addresses the objective *area throughput* and can potentially address the objective *energy efficiency*.

CT2.5 Partially Centralized Inter-Cell Interference Coordination

CT2.5 aims at improving the *area throughput* when the channel state information (CSI) can only be imperfectly exchanged among nodes. It will provide a gain which depends on the quality of the CSI available at each iSC and the transmit signal-to-noise ratio (SNR).

CT2.6 Data Compression over RoF

CT2.6 aims to reduce the backhaul rate compared to the baseline, which reflects in an increase of the number of served users per unit area for the same backhaul rate (i.e. *area throughput*). For some PHY functional splits envisaged by CT2.6 the backhaul rate improvement comes also from the statistical multiplexing gain of the backhaul traffic generated by different iSCs connected to the same backhaul network. Considering that in principle the backhaul capacity is linked to the backhaul cost, the backhaul rate reduction provided by CT2.6 may also potentially affect the *cost efficiency*".

CT2.7 Millimetre wave backhauling

CT 2.7 aims to provide a BH technology that is able to meet the other CTs requirements in terms of capacity, range, latency and reliability. In that regard it is an *enabler* for other CTs, especially those aiming to increase *utilization* of hardware by centralized processing.

With the use of mmWave BH, the *cost* of a network can be decreased, since mmWave links do not require earthworks and thus, the deployment cost will be lower as compared to fibre. Compared to traditional microwave links they are also cheaper in terms of licensing as only "light licensing" (70-90 GHz) or no licensing (60 GHz) is required. By the proposed removal of an additional encoder, the hardware of the mmWave links will also be simpler resulting in lower hardware cost and less energy consumption.

By increasing the reliability of the BH link, the range can be extended, further lowering the *costs* as less links per area are required, or enabling otherwise impossible topologies. An increased reliability can also be traded off to reduce transmit power, increasing the overall *energy efficiency* of the network, or to achieve a higher *throughput* when facing unfavourable channel conditions.

CT3.1 Backhaul Link Scheduling and QoS-aware Flow Forwarding

This CT proposes the efficient BH link scheduling (activation / de-activation) in a millimetre-wave small cell BH environment so as to ensure high throughput and low latency small cell backhaul taking into account the traffic demand for the access per iSC and the users' QoS requirements for different types of traffic. Moreover, this CT might improve the utilization efficiency by optimizing the small cell BH resource usage.

CT 3.2 Partly decentralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments

This CT aim to improve the overall system throughput by implementing a coordinate load balancing mechanism amongst neighbouring iSCs. By varying the cell association at mobile UEs, bottleneck due to the RAN and BH capacity can be avoided. Moreover, this CT also improve the overall network utilisation efficiency by increasing the number of active iSCs as well as the radio access and BH resource usage.

CT 3.3: Energy-Efficient MAC/RRM at Access and Backhaul

This CT aim to improve the radio access and backhaul network energy efficiency by dynamically activating and deactivation BH and access links. Non urgent data (according to the QoS requirements) is buffered at the RANaaS while iSCs and BH links are idle. When needed the nodes are activated by the RANaaS and data is transmitted towards the corresponding UE.

CT 3.4 Computational Complexity and Semi-Deterministic Scheduling

This CT proposes a partially distributed multi-level scheduling algorithm, which introduces robustness against imperfect channel state information (CSI). The proposed scheme aims for increasing the spectral efficiency by satisfying the proportional fair metric and a fixed outage probability at the same time.

CT 3.5 Cooperative RRM for Inter-Cell Interference Coordination in RANaaS

This CT proposes a graph-based ICIC mechanism for a dense small cell network. CT3.5 aims to enhance small cell's spectral efficiency / throughput by jointly scheduling users of different cells and at the same time to mitigate inter-cell interference by keeping the outage probability in low levels.

CT 3.6 Utilization and Energy Efficiency

This CT proposes new metrics and does therefore not have any functional dependencies on other CTs.

CT 3.7 Radio Resource Management for Scalable Multi-Point Turbo Detection

This CT will schedule users on the same resources and exploiting the created interference as an additional source of information. Therefore, it is expected that the area throughput should be increased in theory, addressing the first objective defined in iJOIN.

CT 3.8 Radio Resource Management for In-Network-Processing

This CTs aims to increase the area throughput by scheduling of several UEs to the same physical resources. INP can also be applied to uplink signals of orthogonally scheduled UEs, thus an SNR gain can be achieved, allowing for a reduction of UE transmit power, and therefore improving energy efficiency.

CT 3.9 Hybrid local-cloud-based user scheduling for interference control

This CT aims at increasing the area throughput via coordination of the scheduling decisions at the different iSCs. In contrast to conventional approaches which require the exchange of the totality of the CSI, this coordination is either realized solely on the basis of the statistical information or through the exchange of a few coordination bits, when a backhaul link is available to exchange within a short delay (some ms) these coordination bits.

CT 4.1 Distributed IP Anchoring and Mobility Management

The main goal of CT4.1 is to select dynamically the optimal anchor for each UE. By selecting an optimal anchor is possible to avoid the redirection of the traffic along sub-optimal paths in the backhaul network. Such behaviour leads therefore a *utilization efficiency* improvement of the network.

CT4.2 Network Energy Optimization

The objective of CT4.2 is to minimize the *energy* consumption that the cellular base stations and backhaul links/nodes spend, by reducing the amount of power wasted in idle (or almost idle) cells, while maintaining a desired user QoE. This also leads to *cost* reductions, as power consumption constitutes a major OPEX for operators.

CT4.3 Joint Path Management and Topology Control

The main goal of CT 4.3 is to position (location) and dimension (CPUs) RANaaS platform inside the EPC network and how to associate each RANaaS with a particular set of iSCs. By optimally

positioning/dimensioning RANaaS platform results in more efficient path computation (u*tilization efficiency*) and more economic RANaaS deployment (*cost efficiency*).

CT4.4 Routing and Congestion Control Mechanisms

The objective of CT4.4 is to allow the support over the same transport infrastructure of simultaneous information flows that implement different functional split options. This may allow a greater flexibility in the deployment and operation of the network and a *cost* reduction of the infrastructure installed.

CT4.5 Load Balancing and Scheduling

The main goal of CT 4.5 is twofold: (i) to balance the traffic load among base stations and backhaul links, in order to improve the *utilization* of each element, and ensure resources are spent where needed; (ii) to propose smart scheduling policies that can accommodate a given load. Also, the user QoE, e.g. service delay, latency is improved.

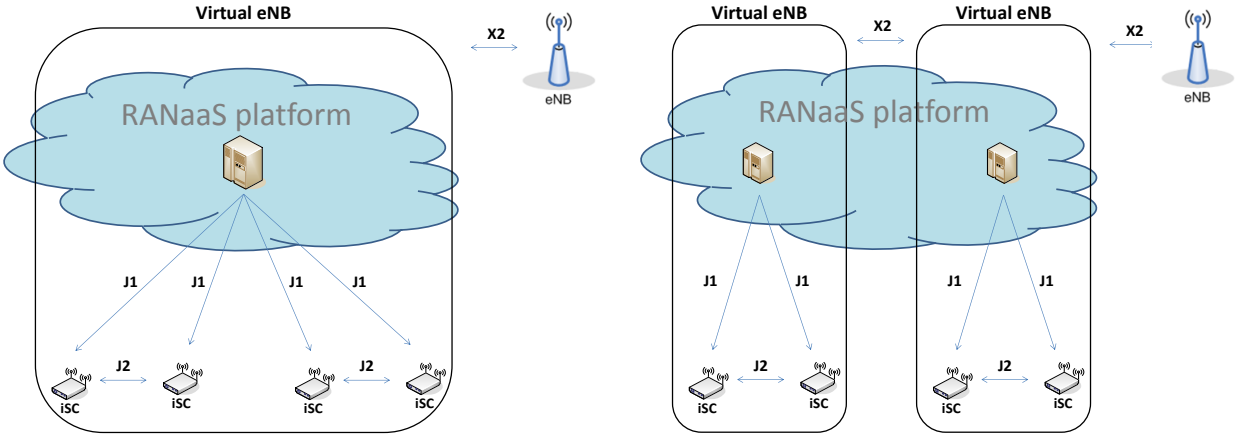
CT4.6 Backhaul Analysis based on Viable Metrics and "Cost" Functions using Stochastic Geometry

The main objective of this CT is lay down the basics for estimating deployment cost. This framework allows comparing the *cost* of implementing iJOIN technologies with other newer technologies or with technologies that are already in use.

3.3 Physical Architecture and Common Scenarios

The following subsections describe the four physical architectures identified by respective Common Scenarios defined in iJOIN (see also D5.1 [4]). In general we can notice that every physical architecture differs in terms of deployment scale, number of nodes and particular placement of physical interfaces (realizing logical connections in different ways). Moreover, in all scenarios a RANaaS instance is coordinating iSCs and its implementation in cloud platform may consist of many Virtual Machines (VMs) representing the baseband processing units of the coordinated iSCs.

According to the definition of the veNB, a set of cells (each one with corresponding ID cell) belongs to the same veNB. Figure 3-2 exemplarily shows two scenarios with and without inter-veNB communication in the same RANaaS platform.



(a) A single virtual eNB in the RANaaS platform

(b) Several virtual eNBs in parallel at the RANaaS platform

Figure 3-2: RANaaS and virtual eNodeB configuration options

If more than one virtual eNB (Figure 3-2(b)) is executed at the same RANaaS platform, it may need to involve the X2 interface in order to realize coordination. Each veNB is seen from the core network as a eNB and can communicate with other (v)eNBs through X2. In this case, in the view of realizing fast coordination among the cells, also X2 protocol limitations should be taken into account.

3.3.1 Common Scenario 1: Stadium

This scenario is formed by a stadium covering a wide area, in the order of 50.000 m^2, and containing several thousands of spectators. The average number of spectators that can be taken into account is 40.000. Multiple iSCs are installed to provide the needed coverage and capacity in the area. Multiple macro cells can be present to provide sufficient overage also outside the stadium, as depicted in Figure 3-3.

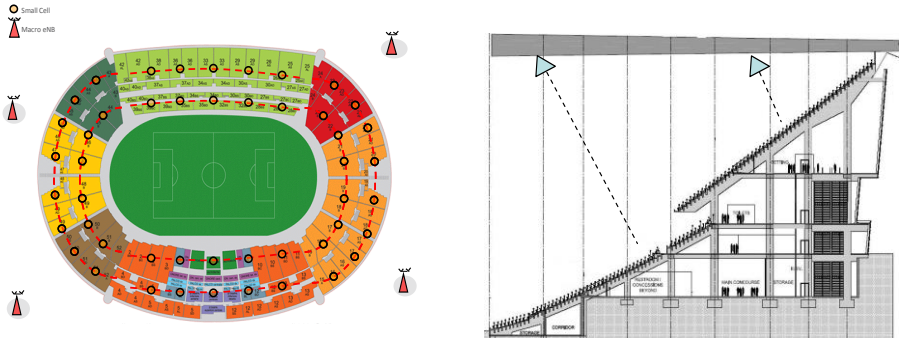


Figure 3-3: Stadium – iSCs and macro cell positions (left) and details on iSCs antenna tilt (right)

The key characteristics of this scenario are:

- Multiple rings of iSCs providing coverage in the stadium (2 rings are envisaged in Figure 3-3).
- All iSCs and the macro eNBs are coordinated by one iNC node controlled by the same RANaaS data centre.
- A tight coordination is envisaged among the iSCs. A loose coordination between macro and iSC layers can be considered under the control of iNC node.

Based on these characteristics, Figure 3-4 illustrates a possible physical deployment.

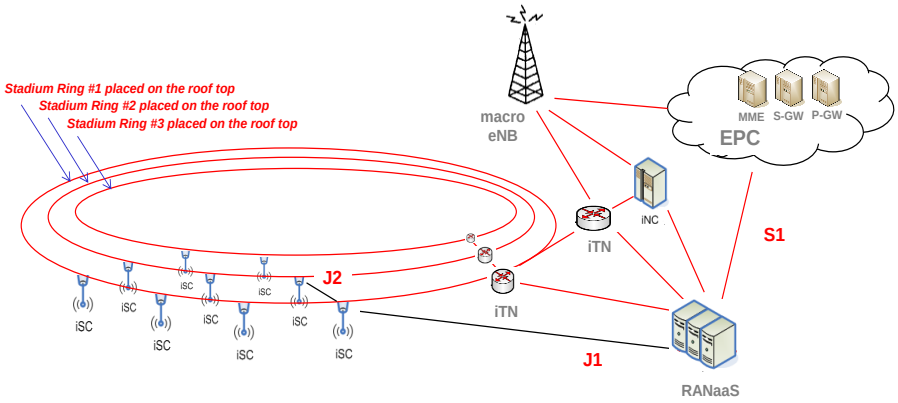


Figure 3-4: Stadium – Physical deployment example

3.3.2 Common Scenario 2: Square

This scenario is a typical square where multiple iSCs cover a wide area (in the order of 15.000 m^2) to meet coverage and capacity demands in a high user dense environment. Figure 3-5 illustrates an example square deployment. The key characteristics of this deployment are:

- The RAN deployment for the square is based on a dense random deployment of iSCs.
- All deployed iSCs within one area are connected to the same RANaaS data centre. Hence, all iSCs from the square are processed at the same RANaaS data centre.

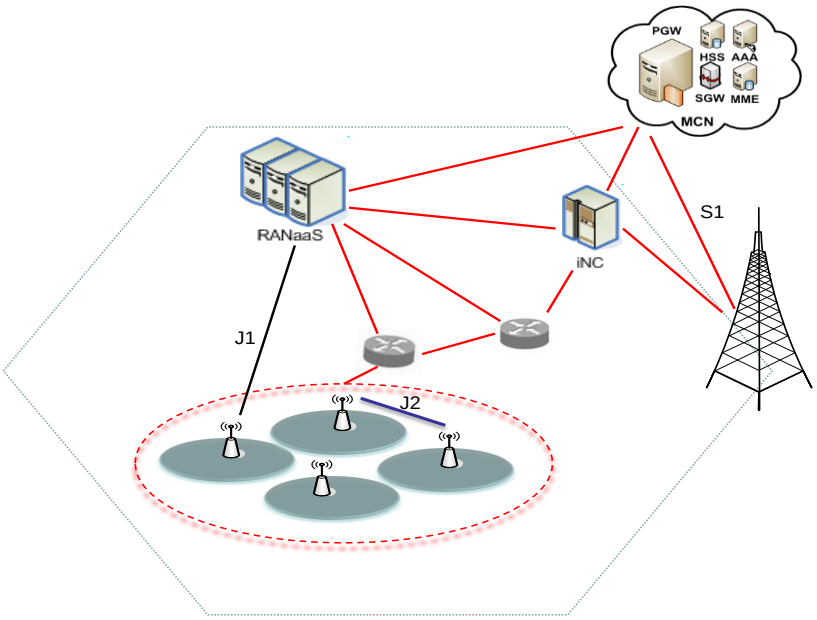


Figure 3-5: Square - Physical Deployment example

3.3.3 Common Scenario 3: Wide-area continuous coverage

In this scenario, iSCs are used to provide continuous coverage over a wide area, up to several square kilometres, preferably in an urban environment. This layer serves as an additional layer to the macro-cell layer in order to offload traffic. The key characteristics of the scenario are:

- iSCs are expected to be deployed taking into account the topography and morphology of the area to be covered.
- Different backhaul supporting technologies may be employed, e.g. taking advantage of deployed fibre infrastructure or potential line-of-site (LoS) links with macrocells. Wireless inter-iSCs links can be considered as well.
- Connecting the iSCs with the RANaaS infrastructure may require an aggregation network, which may impose limitations regarding the supported functional split.

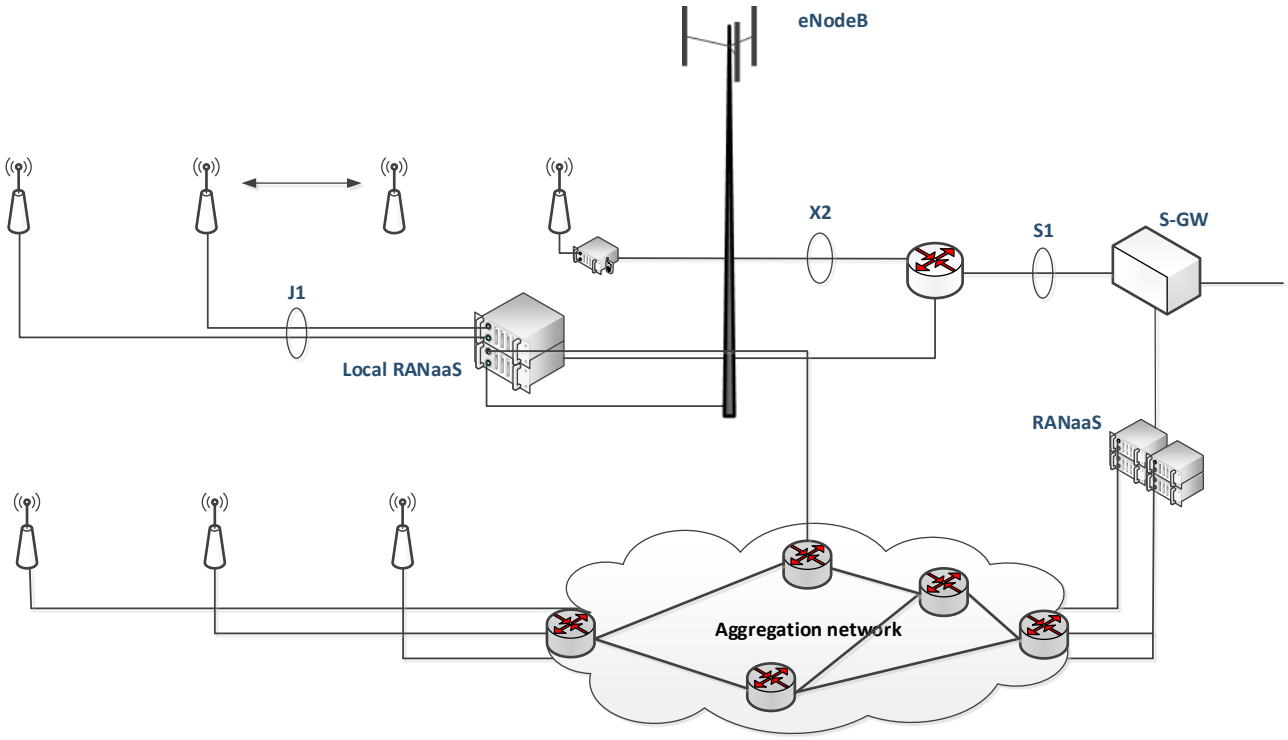


Figure 3-6: Square - Physical Deployment example

3.3.4 Common Scenario 4: Shopping Mall / Airport

Figure 3-7 shows one feasible deployment of the iJOIN architecture for shopping malls or airports. The main characteristics are:

- All deployed iSCs within one place are connected to the same RANaaS platform.
- One gateway to connect to the EPC (through RANaaS data centre)
- All iSCs within one building are connected through a heterogeneous backhaul including Ethernet, wireless, or GPON.
- Line-of-sight between iSC and user terminal, and iTN and iSC may be feasible.

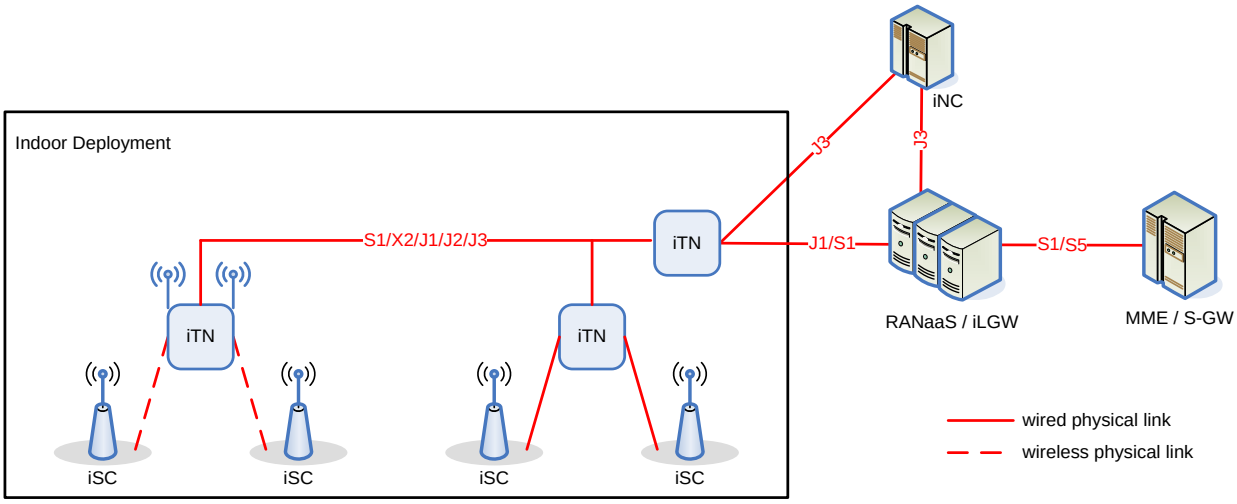


Figure 3-7: Shopping Mall / Airport: Physical deployment example

3.4 Network Sharing Enablers

As the traditional model of single ownership of all network layers and elements is being challenged, network sharing is emerging as a mechanism for operators to substantially improve network costs and to efficiently utilize network capacity. More and more operators are adopting network sharing as a means of cutting the heavy Capital Expenditure (CAPEX) and Operating Expenditure (OPEX) costs involved in the initial roll-out and operation of mobile networks.

The main motivations for operators adopting network sharing schemes are:

- Increased rollout speed
- Quickly expand coverage to meet customer demand for wider coverage
- Sharing low-traffic areas leads to long-term cost advantages
- Sharing high-license obligations
- Cost efficiency (CAPEX and OPEX)
- Joint effort to offer availability of services at more affordable price

3.4.1 Network Sharing in the context of 3GPP

3GPP has been working on providing standardised solutions for different alternatives of RAN sharing. The main milestones are collected in Figure 3-8.

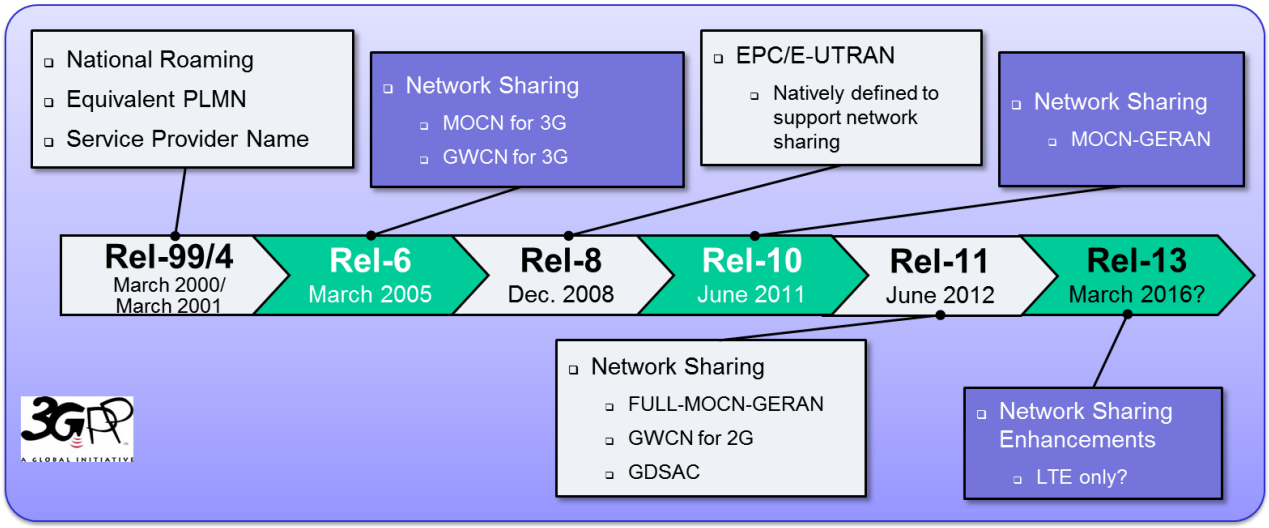


Figure 3-8: 3GPP support of network sharing

In general, the solutions supported differ in terms of the level of infrastructure integration between operators, from roaming agreements to complete network (both access and core) sharing. Most of the benefits are usually associated to the RAN sharing (where iJOIN is focused), which is responsible of most of CAPEX and OPEX.

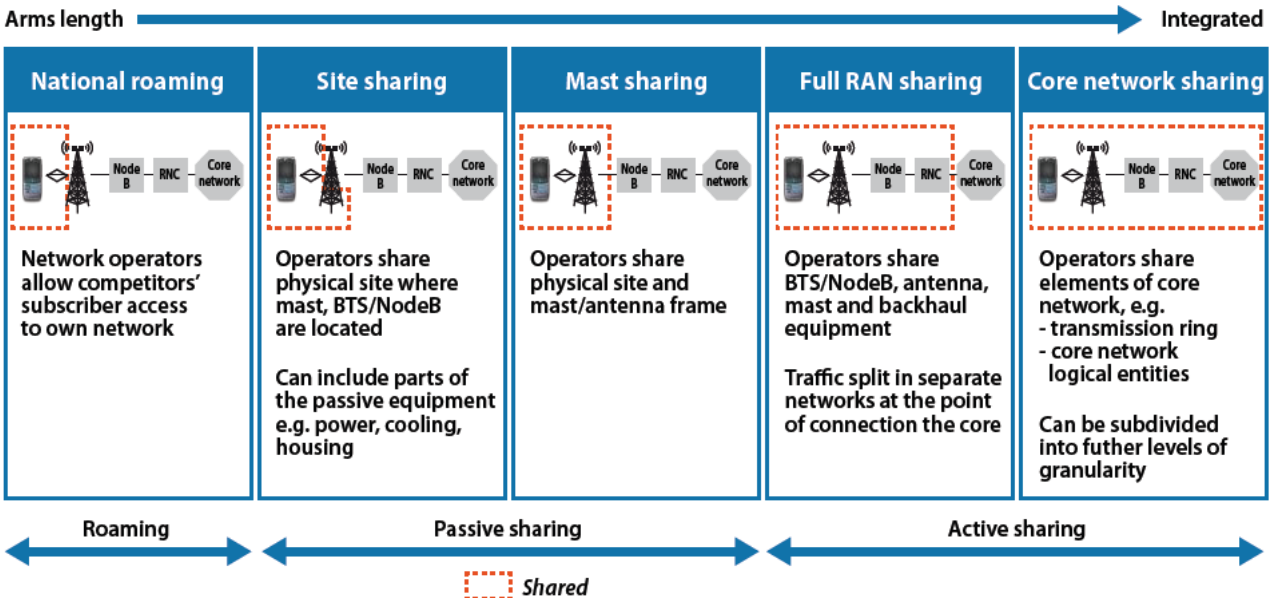


Figure 3-9: Degrees of integration in network sharing solutions

In the case of RAN sharing the 3GPP Services WG SA1 in [16] specifies five main use cases for RAN sharing:

- **Sharing a common RAN**: but not the radio frequencies (Release 99). In this case the operators connect directly to their own dedicated carrier layer in the shared radio network controller (RNC) in the shared RAN.
- **Operator collaboration to enhance coverage**: where two or more operators with individual frequency licenses cover different parts, e.g. of a country, but together provide coverage of the entire country.
- **Sharing coverage on specific regions**: where one operator provides coverage in a specific geographical area, with other operators allowed using this coverage for their subscribers. Outside such RAN sharing area, coverage is provided by each of the operators independently.

- **Common spectrum sharing**: considering the following two variations: i) one operator has a frequency license and shares the allocated spectrum with other operators ii) a number of operators decide to pool their allocated spectra and share the total spectrum.
- **Multiple RANs share a common core network**: where the multiple RANs can belong to different PLMNs and network operators. Due to operators' deployment choices, different nodes or part of the common core network can be shared.

Active RAN sharing enables partitioning or pooling of radio resources enhancing the overall RAN utilization. At the same time, investments for installing new infrastructure may be reduced as well. In 3GPP, WG SA1 conducted a study on RAN sharing which analyses a set of use cases and derives business requirements [17] . This study aims to outline ways for sharing RAN resources, maintaining and sharing policies, and providing flexibility in RAN resource sharing on-demand within shorter time periods. The architecture and operations that enable different mobile operators with a separate core network to share the RAN are specified by the 3GPP Architecture WG SA2 in [18], detailing the following two approaches:

- **Multi-Operator Core Network (MOCN)**, where each operator has its own EPC providing a strict separation among the core network and RAN. This enables certain benefits regarding service differentiation and interworking with legacy networks. Shared eNBs are connected to core network elements of each different operator, i.e. Mobility Management Entity (MME) and Serving-Gateway (S-GW), using a separate S1 interface, allowing load balancing policies to be provided within each operator's core network.
- **Gateway Core Network (GWCN)**, where operators share additionally the MME. This approach enables further cost savings compared to MOCN, but at the price of reduced flexibility, i.e. no mobility for inter-Radio Access Technology (RAT) scenarios and no Circuit Switching (SC) fall-back for voice traffic.

In general, MOCN is more expensive but more flexible, addressing conventional operators' needs. In both cases, the UE can distinguish up to six different operators that share the RAN infrastructure based on broadcast information, i.e. Public Land Mobile Network (PLMN)-ID, and can signal to obtain connectivity or perform a handover irrespective of the underlying RAN sharing arrangement. Specifically, the S1 interface supports the exchange of PLMN-IDs between eNBs and MMEs in order to assist the selection of the corresponding core network [19]. The X2 interface supports a similar PLMN-ID exchange among neighbouring eNBs for handover purposes [20]. Considering broadcasting, the Uu interface supports the PLMN-IDs enabling the UEs to perform the network selection [21].

3.4.2 Benefits of Network Sharing

In the framework of the iJOIN project, it is worth understanding which can be the benefits in considering network sharing in conjunction with the main enablers defined by the project, RANaaS implementation and joint access/backhauling design. At this respect some considerations must be made:

- The main objective of sharing is to reduce costs, both capital and operational – if sharing does not result in a cost efficient solution, it probably should not be pursued. In other words, if, as expected, iJOIN technological solutions result in a reduced cost for operators, they may become inhibitors for RAN sharing.
- The possibility of sharing the spectrum is usually precluded by regulators. This may preclude the realization of some potential advantages by adopting iJOIN architecture.

On the other hand, it is clear that the iJOIN architecture may open up new possibilities to overcome some of the issues associated to network sharing, such as the reduced flexibility for operators to differentiate from a technical viewpoint. In this sense, operators may be able to contract different network services from the RANaaS and backhaul (iNCs and iTNs) elements (e.g. supporting different functional splits and associated network services, different transport services, etc.). It may be also possible for operators to implement their own processing procedures on top of the RANaaS, even if it is shared with other operators.

In order to identify whether there are new technical requirements for iJOIN enablers for supporting network sharing or not, it is proposed to analyse a number of network sharing scenarios with different levels of integration between cooperating operators.

However some conclusions can be extracted from a preliminary analysis:

- Sharing of the RANaaS infrastructure should not be problematic, because cloud infrastructure and technologies are specifically designed to allow for sharing the processing tasks to be carried out. However, it is not clear which may be the advantage of sharing it.
- Supporting flows from different operators when reusing the same transport (backhaul infrastructure should be taken into account in the protocol design. In WP4, it has been proposed to use IEEE 802.1ad QinQ mechanism to differentiate flows. It should be verified that the mechanism can be adapted in the case of multi-operator support.
- The iJOIN architecture may allow operators to contract different services from a backhaul provider, as well as different control capabilities. The SDN based iJOIN architecture may allow this to happen, but in this case the iNC should provide some kind of northbound open interface so operators can configure the backhaul services they want to be provided (e.g. implementing different security mechanisms or using different local breakout points). But it should not be feasible for an operator to enhance its performance at the expense of the other, i.e. congestion control corrective procedures should be under the control of the backhaul operator.

4 Functional Split Implementation Aspects of RANaaS

4.1 Implementation aspects of RANaaS hardware

4.1.1 Implementation Options

Hybrid solution for virtualization

The C-RAN (Centralized RAN) approach allows for offloading cellular processing from based stations to centralized servers. The most common solution proposed for supporting virtualized baseband processing with a C-RAN architecture are based on the combination of general purpose processors (GPP) with hardware accelerators or co-processors that implement specialized digital signal processing functionalities [47]. The former can be implemented by means of DSPs, FPGAs, ASICs or a combination of them. GPP can be based on ARM, MIPS or x86 ISAs (Instruction Set Architectures). Co-processors communicate with the CPU using a standard interface (e.g., PCIexpress). The approach is illustrated in Figure 4-1. Algorithmic bottlenecks that prevent a pure-software implementation running on GPP can be eliminated by the use of custom hardware accelerators that offload data processing from the CPU. The same approach has been followed to support in an integrated way graphic processing capabilities (combination of CPU and GPU) or packet processing capabilities. The next logical step is to define a programming model for the co-processor that is also GPU-reminiscent, akin to DirectX or OpenGL's abstraction of a computer's graphics subsystem.

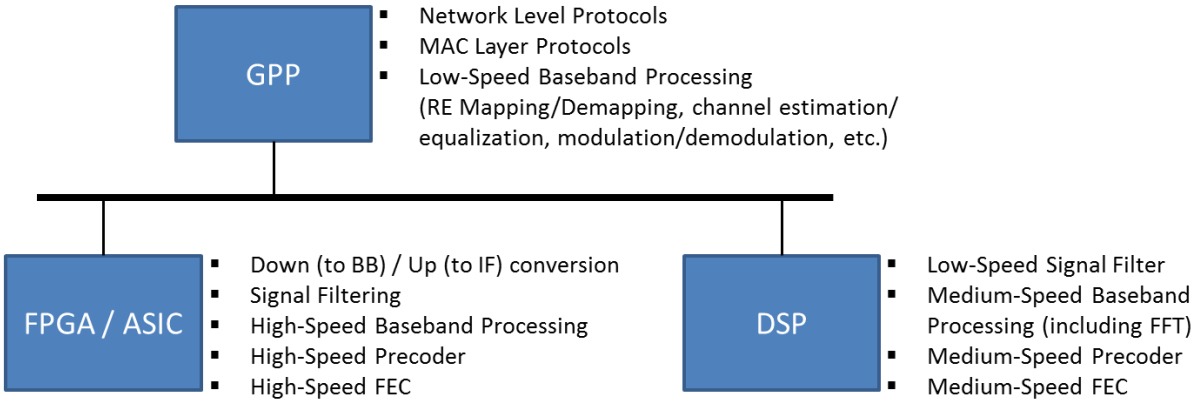


Figure 4-1: Example of splitting of digital signal processing across GPP, DSP, and FPGA

The following functions are proposed by the Israeli company ASOCS [46] to be implemented by the co-processor:

- **FEC (decoder)**: Accelerator for decoding of turbo and convolution codes.
- **Demapper**: Extracts soft bit values from QAM signals, (LLR values), slicer decisions and slicing error values.
- **Arithmetic**: Performs all the intrinsic arithmetic functions required by the co-processor, e.g., matrix and scalar multiplications, windowing, and frequency correction.
- **DFT/FFT**: Supports orthogonal transforms (FFT and DFT). Frequency correction can be done on the input to the FFT/DFT unit.
- **Logic operations**: Specialized for scrambling, pseudo random bit stream generation, encryption solutions, and various encodings (e.g., convolutional and turbo). It may operate on hard or soft bits and can also perform interleaving and arithmetic operations.
- **Data rearranging**: Its main purpose is to support interleaving and data manipulation. The unit can move and interleave large volumes of data, as well as handle IR (incremental redundancy), puncturing and simple decoding at high rates.

It can be noticed that functionalities identified as “software” (e.g. channel estimation or MIMO processing) are not supported by the co-processor. It can be argued that the same kind of approach is already followed by

the solutions provided for baseband processing in commercial base stations. However, two main differences between base station solutions and virtual RAN solutions should be noticed:

- In most baseband processing units for base stations, GPP responsibilities are limited to scheduling and coordination of DSP functionalities carried by specialized hardware, while in the virtualization solution, a significant part of the processing is carried out by GPPs – as discussed in the previous section.
- Solutions for the virtualized architectures should support resource virtualization as understood in the Information Technologies (IT) realm, while the base station solutions are intended as dedicated elements.

The main reason for using this kind of hybrid solutions with co-processors is the fact that the full implementation of radio interface baseband processing by means of GPPs may be suboptimal in terms of required investment, energy consumption and other performance parameters. On the other hand, centralization of conventional baseband processing units does not allow for an easy virtualization of the resources and the reuse of IT solutions.

Implementation of RANaaS in iJOIN

The support of RANaaS and flexible functional split introduce a new level of complexity, as it may require the solution to support different levels of processing without penalizing the network TCO (Total Cost of Ownership). In this sense, it should be noticed that the solutions previously described are expected to be deployed on the cloud infrastructure, while the distributed elements are iSCs, which only support a limited set of baseband processing functionalities (depending on the functional split).

In the context of the iJOIN architecture, the iSC may implement different levels of baseband processing, depending on the functional split – from only RRH (radio remote head) functionalities (like in a Centralized-RAN scenario) to full support of the whole radio interface protocol stack (like a conventional base station). The same operating scenarios are applicable to the RANaaS infrastructure.

The requirements for an ideal solution would be the following:

- The same solution should be reusable for both iSCs and RANaaS infrastructure, in such a way that processing elements may be moved from the iSC to the RANaaS and vice versa.
- It should be possible to switch off those processing units (CPU cores, DSP co-processors) that are not required for the functional split selected.
- It should be possible to virtualize the capabilities of the processing elements, in such a way that the functionalities they implement may be decoupled from their locations. For instance, the processing elements of an iSC may be used for processing connections of other iSCs, if the backhaul infrastructure is efficient enough.
- It should be possible to reuse the same solution for the virtualization of other network elements, not necessarily of the mobile network (e.g., virtualization of CPEs, implementation of virtual switches, etc.).

4.1.2 Virtualization infrastructure

In iJOIN, the initial target chosen for the RANaaS implementation is a cloud computing platform delivering general purpose computational resources. The platform implements an Infrastructure as a Service (IaaS) model where resources are provided on a “as a Service” paradigm meaning that resources are allocated and deallocated on demand.

The resources provided by an IaaS platform can be classified as follows:

- computation resources: Virtual Machines (VMs), running an operating system and application software;
- storage resources: Virtual Volumes, storage elements that can be attached to VMs;
- networking resources: Virtual Networks objects like Virtual Level 2 (L2) trunks, subnets, DHCP services, etc.

IaaS platforms usually provide resources using virtualization. Virtualization aims to simulate the existence of a piece of hardware which is “materialized” by a software layer running on top of the physical device. The

idea is that the actual hardware is hidden to the applications and partially or temporarily used for "impersonating" the role of a virtual piece of similar hardware. The actual computation happens at the physical level but physical resources and applications are not tightly bound to each other. This makes it easier to reuse the physical infrastructure for several purposes, usually at different times.

As described in Section 4.1.1, common solution designs for supporting virtualized baseband processing use a combination of general purpose processors with hardware accelerators or co-processors for implementing specialized digital signal processing functionalities. Mapping this kind of architecture into an IaaS platform raises the problem of how to distribute the related workload on a virtualized platform. Specific attention must be paid for parallel computation which is traditionally obtained using specialized hardware devices (e.g. DSPs or FPGAs).

IaaS platforms implement server virtualization where more than one virtual server runs on top of a single physical computer. This is implemented by using a hypervisor which runs on the physical hardware and which takes care of running several virtual servers. Figure 4-2 summarizes the concept.

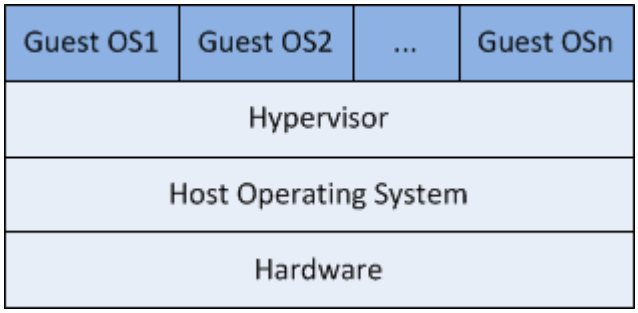


Figure 4-2: Server Virtualization

At the bottom of the stack, the physical hardware provides the actual computational resources (e.g., CPUs and RAM), an operating system is installed on the bare metal and it is integrated with the hypervisor. Each virtual server appears as an autonomous computer having its own (virtual) hardware. Users access virtual servers via network connections. Similar techniques are available for implementing storage and network virtualization.

Hypervisors are designed to minimize the processing overhead and to allow for almost the same performance as non-virtualized environments. In addition, in case a VM is assigned a certain number N of (virtual) CPUs, it can (virtually) execute up to N processes/threads in parallel. It's important to say that, when a VM is started, it is possible to define the number of virtual CPUs that the VM will use. This number defines a virtual parallelism that becomes real parallelism only when the number of physical CPUs dedicated to the execution of the VM corresponds to the number of real CPUs dedicated to it. This aspect is regulated by the overbooking factor.

Overbooking can be defined as the ability of running a number of virtual servers requiring more hardware resources than available. For example, a physical server with 8 CPUs can run a certain number of virtual machines allocating a total of 10 virtual CPUs (i.e. vCPUs). This is possible because usually not all virtual machines are running at the same time. Therefore, when a VM is waiting for a "slow event" (i.e., an interrupt), the real CPUs are used for running concurrent VMs.

Similarly, a physical server with 64 GB RAM can accommodate a number of virtual machines requiring 200 GB RAM. In such a case, memory swapping techniques are used for transferring main memory "chunks" on the mass storage. Overbooking can have strong impact on the performance because it may happen that, under certain conditions, the actual workload of a virtual server cannot be supported by the available physical resources and some VMs are randomly suspended independently on the priority of the applications they are running. In this case, a simple software configuration should take care to prevent overbooking.

This is an important aspect to consider when designing a real-time application and, specifically, parallel algorithms: actual parallelism is obtained by allocating sufficient virtual CPUs for the VM hosting the algorithm. Furthermore, the amount of virtual resources allocated to the execution of a VM can only be defined at start-up and cannot be changed throughout the entire VM lifetime. Consequently, when running a parallel algorithm on a single VM, it is fundamental to allocate a sufficient number of CPUs to support the required level of parallelism. Then, if overbooking is disabled, the cloud computing platform (not the hypervisor) selects a physical server where this condition is satisfied and starts the VM on top of it.

Through virtualization, IaaS platforms provide the low-level building blocks for implementing systems for hosting parallel programming. In fact, each VM works similar to physical machines (i.e., servers) and, as described above, can utilize multiple processors of the hosting physical machine for elaborating the assigned workload. In addition, several VMs can be activated and can collaborate for expanding the computing power dedicated to the workload at hand. In this manner, parallelization can be implemented either within a single VM or across several collaborating VMs. Figure 4-3 summarizes the concept.

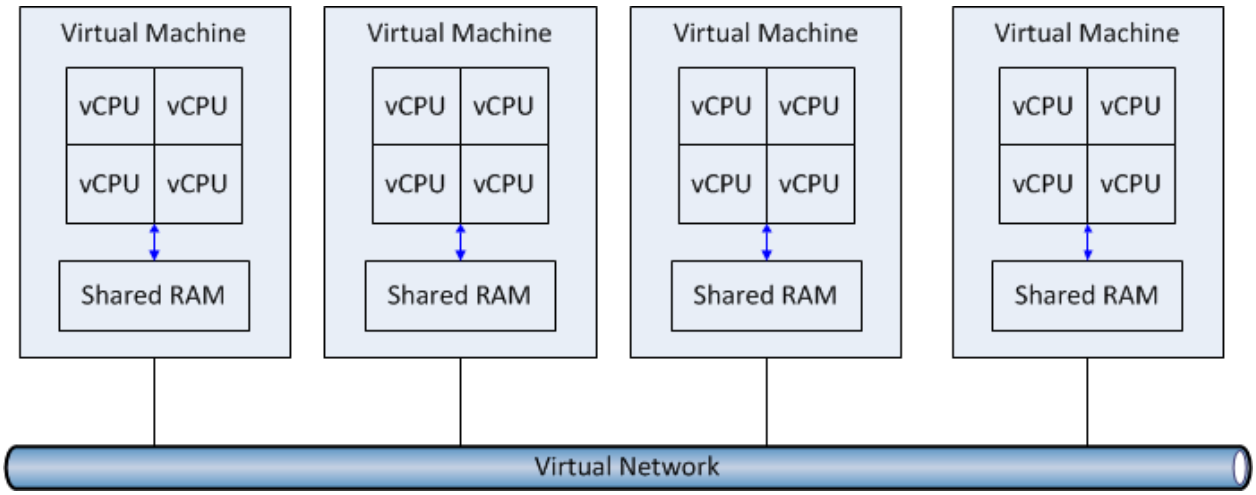


Figure 4-3: Virtual Machine Cluster

Every single virtual machine works as a Symmetric Multi-Processor machine (SMP), a computer with multiple processors and cores which all share a single address space. SMP units, in turn, can be connected through a virtual network giving the possibility of creating parallel computer clusters for further parallelization of algorithms. It is important to mention that every single VM runs inside the boundaries of a physical node. On the other hand, two different VMs can be hosted in a single node. In the latter case, the communication of two VMs co-located on a single physical node is obtained through a virtual network.

Through virtualization and, more in general, IaaS platforms, it is possible to implement various cluster topologies for supporting parallel computing. Specifically, parallelism can be alternatively supported within every single node or by distributing the algorithm across several VMs. Therefore, it is important to analyse the advantages and disadvantages of implementing parallelism within a single element of the cluster (i.e., a VM) versus the advantages and disadvantages of implementing parallelism activating several collaborating VMs. These indications provide the guidance for finding the best balance that fits the problem at hand.

In a single VM, all vCPUs access the memory as global address space, multiple processors can operate independently but share the same memory resources and changes in a memory location caused by one processor are visible to all other processors. From a programming point of view, the global address space facilitates data sharing between parallel tasks and access to data is both fast and uniform. On the other hand, single VMs lack scalability. As mentioned above, the number of vCPUs can only be decided at start time and cannot be changed throughout the VM lifecycle. In addition, adding more vCPUs can geometrically increase traffic on the shared memory-CPU path, and for cache coherent systems, geometrically increase traffic associated with cache/memory management. From a programming perspective, synchronization for ensuring the correct access to global memory must be explicitly indicated by the programmer through constructs such as semaphores, barriers, and queues. On this side, standard software libraries such as POSIX threads (Portable Operating System Interface for Unix threads or Pthreads, for short [41]) or OpenMP (Open Multi-Processing [37]) can significantly facilitate the programmer.

Considering all the VMs in a cluster, it might be noticed that each VM has its own memory address that does not map to other VMs. Each VM operates independently and changes to its local memory have no effect on the memory of other VMs, even in the case they run on the same physical node. When a VM needs access to data in another VM, it is usually up to the programmer to explicitly define how and when data is communicated. Elasticity is the main advantage of this approach because resources (e.g. processors and RAM) can be easily adapted to the workload changes. Whenever the workload changes, VMs are provisioned/de-provisioned following a so-called scale-out paradigm (or horizontal scalability) that spreads the workload over several virtual machines, possibly running on different physical computers. The flip side is that the programmer is responsible for the details associated with data communication between tasks

running on different VMs, in some cases it could be difficult to map the data structures used by the algorithm on a distributed memory architecture (this strongly depends on the problem at hand) and there is a non-uniform memory access times (i.e., data residing on remote nodes takes longer to access than node local data). In addition, starting a new VM for supporting workload may take up to several minutes.

As mentioned above, IaaS platforms provide tools for creating VM clusters that, from the programmer view point, can be considered and treated as clusters of 'real' machines. In this perspective, the programmer can take advantage of software technologies specifically designed for implementing parallel programming on top of server clusters. These technologies come as software libraries or compiler directives that permit the programmer to distribute the tasks on the servers participating to the cluster. Every technology implements a parallel programming model that, theoretically, can be deployed independently on the underlying cluster topology.

For example, it is possible to implement a shared memory model where all the tasks access a virtually shared memory even when they run on different machines of a distributed topology (e.g. Kendall Square Research (KSR) ALLCACHE [42]). In such a case, it is up to the used technology to create the illusion of a single shared memory, hiding the underlying complexity for implementing that. From the other end of the spectrum, it is possible to implement a pure distributed memory model where each task has its own private memory and communicates with the other tasks only through a message passing mechanisms (e.g., OpenMPI [38]) even when the two tasks operate on the same machine.

It is important to say that there is not a unique programming model to apply because most of the times it strongly depends on both the problem to solve and the parallel algorithm to implement. Usually hybrid solutions, where the two models are applied at the same time, permit a better usage of the underlying infrastructure at the price of some additional complexity. Figure 4-4 shows an example where different parallel programming technologies are used on a VM cluster created on an IaaS platform.

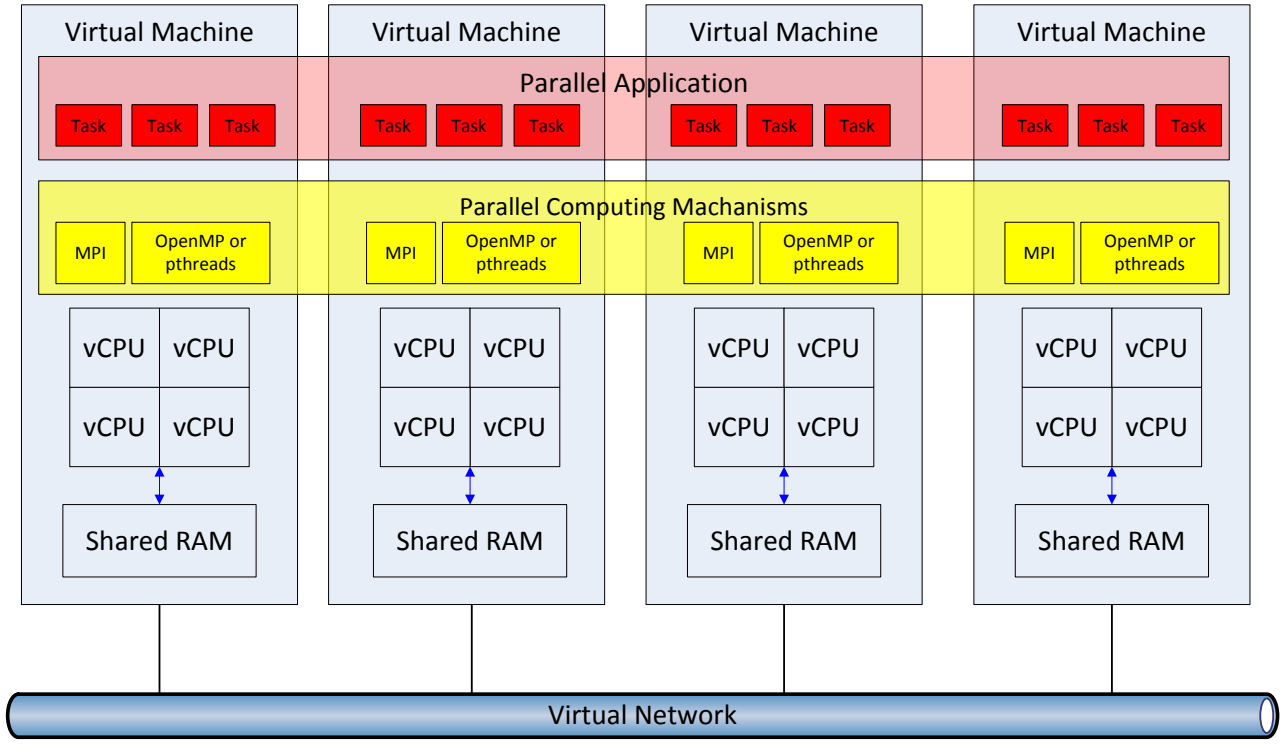


Figure 4-4: Hybrid Programming Model on an IaaS VM Cluster

The figure shows a parallel application distributed on a cluster of VMs, each VM hosts a certain number of tasks. A software technology such as OpenMP [37] or Pthreads is used for managing the tasks running within a single VM: it provides mechanisms for creating, coordinating and synchronizing tasks all sharing the memory of the VM. Another software technology, such as a Message Passing Interface (MPI) implementation, is used for managing tasks running on different VMs.

The software technologies used for implementing parallel computing constitutes a layer shown as the Parallel Computing Mechanisms layer in Figure 4-4. The following paragraphs report a short description of the most relevant characteristics.

Open Multi-Processing (OpenMP) is a standard Application Program Interface (API) that allows the implementation of portable, shared memory applications [37]. Through compiler directives or explicit library calls, a programmer implements multi-threaded, shared memory applications. The programmer defines the portions of code that are to be executed in parallel as well as synchronization points for coordinating parallel computation streams. At run-time, the compiled program runs as a process of the underlying operating system split into light-weight threads all sharing the memory address space of the 'hosting' process. OpenMP run-time allocates the processors (i.e. CPUs or cores, depending on the underlying hardware architecture) to thread execution and takes care of maximizing the "actual" parallelism. OpenMP is available for C/C++ and Fortran programming languages and runs under several operating systems, e.g., Solaris, AIX, HP-UX, Linux, Mac OS X and Windows.

Pthreads is an alternative mechanism for implementing multi-threaded/shared memory applications [41]. It comes as a set of library calls originally implemented for Unix operating systems and currently available under other platforms such as Linux. Using Pthreads, the programmer implements a parallel program as a single operating system process providing the same computational resources such as the memory, the network connections, and the file system. Inside a process, parallel threads can be created and managed by explicitly invoking suitable Pthreads library calls. It's up to the Pthreads run-time library to ensure that concurrent threads are actually executed on different processors in order to obtain the maximum level of parallelism. In these aspects, Pthreads and OpenMP are very similar. Pthreads is available for C/C++ binding and provides, in addition to functions for managing threads, also functions for coordinating their executions and concurrent access to memory (i.e., mutual exclusion management, condition variable management, synchronization).

Message Passing Interface (MPI) is a standard definition for a software library that allows for the implementation of parallel programs realizing a pure distributed memory [38], message passing model. It was originally designed for clusters of single CPU computers communicating through network connections but has been more recently adapted to run on clusters of multi-processor computers. The underlying programming model relies on distributed memory, meaning that each parallel task accesses its own private chunk of memory and, in case two or more tasks need to share some data, they need to explicitly exchange messages. However, some MPI implementations optimize the mechanism for exchanging messages. They use shared memory if the tasks run on the same machine and network messages if they do not run on the same machine (either physical or virtual). MPI provides several library calls for sending and receiving messages among tasks and, more importantly, implements a cluster 'concept' where several nodes can be considered as a unique computing platform for distributing the computational workload. The programmer writes parallel programs as monolithic entities but has complete control on the location where parallel tasks will be executed (i.e. on which node of a cluster). MPI is available for many programming languages such as C/C++, Fortran, Java, Perl, Python, and runs under several computing platforms ranging from PC to supercomputers equipped with a range of operating systems, e.g., Unix, Linux, Mac OS.

The combination of the two parallel programming models, shared-memory/multi-threaded model and distributed-memory/message-passing model, opens the possibility of taking advantage of the best characteristics of both. The former model is more suitable to situations where parallel tasks need to exchange data with minimum communication overhead. The latter can be used successfully for addressing scalability issues. How these two technologies are used strongly depends on the problem at hand and the parallel algorithm that solves the problem.

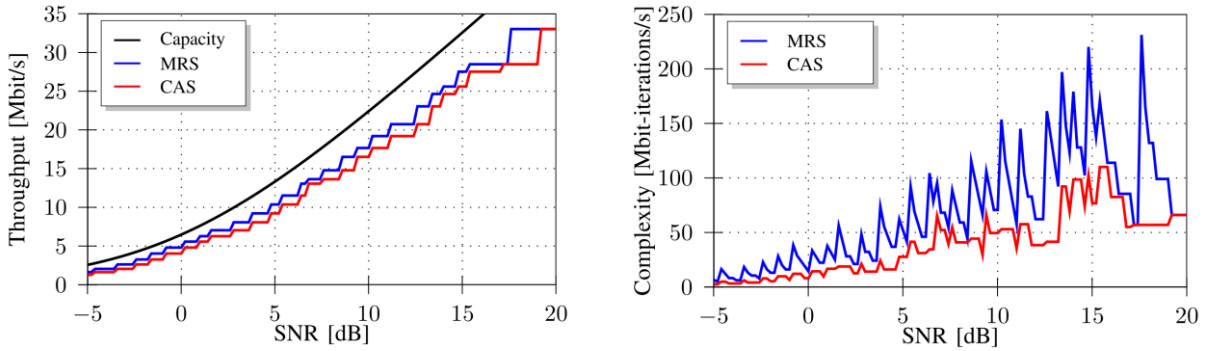
For example, in case a certain algorithm shall be executed with real-time constraints (i.e. the response must be guaranteed within strict time constraints), the best solution is to 'deploy' the related tasks on a single virtual machine of the cluster. This avoids unpredictable or unacceptable delays caused by network communications between tasks running on different machines.

4.1.3 Computational Outage

As explained, a virtualized infrastructure is able to provide an abstract interface between the network platform (RAN) and the underlying physical computational resources. However, these resources are still limited which may lead to computational outage of the RAN rather than channel outage due to difficult channel conditions. For example, a computational outage would occur if the employed Forward Error Correction (FEC decoding software would require more computational resources, such as following from a high number of decoding iterations, than can be provided by the virtualized infrastructure. Computational outage leads to a waste of spectral resources and loss of throughput, in much the same way as a channel

outage. In the case of small-cell deployments where each small-cell needs to implement the full RAN protocol stack, computational outage may even dominate channel outage due to very limited computational resources available at a small-cell base-station.

Therefore, it may not be throughput-optimal if the scheduler only considers the raw throughput while ignoring the required computational resources (we refer to this as Maximum Rate Scheduling (MRS)). By contrast, a Computationally Aware Scheduler (CAS) would consider both the achievable rate as well as the required computational resources. Figure 4-5 (a) shows the achievable raw throughput of both schedulers. Obviously, the MRS achieves a slightly higher raw throughput than the CAS. However, Figure 4-5 (b) shows the required computational resources. As can be seen, the required computational resources of the MRS may be up to 4 times higher than the required resources of the CAS. In this simple example, the CAS has been limited to maximum 2 iterations while the MRS may perform up to 8 iterations. There are more complex schedulers possible which take the current computational load into account and adjust the maximum number of iterations according to the available resources. In Section 4.5.2, we provide further detailed results on the computational outage using a detailed system level uplink simulation employing actual an LTE decoder.



(a) Raw throughput **(b) Computational complexity**

Figure 4-5: Raw throughput and computational effort for rate-maximizing and computationally aware scheduler

Centralizing multiple base-stations and their associated computational load allows for exploiting computational diversity gains. Figure 4-5 shows the strongly varying computational complexity depending on the actual channel quality and the corresponding MCS (Modulation and Coding Scheme). These fluctuations can be well exploited in a centralized system where multiple base-stations share the same pool of resources. As a consequence, the variance of required computational resources is reduced. More formally, let $C_{outage,N}(\varepsilon)$ be the outage complexity for N centralized base stations and an outage probability ε. Outage complexity is the required amount of computational resources such that the probability for an outage due to computational limitations does not exceed ε. Computational diversity gain is now defined by the ratio:

$$c(N) = \frac{NC_{outage,1}(\varepsilon)}{C_{outage,N}(\varepsilon)}. \tag{4.1}$$

This measure gives overprovisioning ratio in the case of a distributed implementation compared to a centralized implementation. It further gives us an indication of the utilization of system because the centralized system is able to perform the operations with only $1/c(N)$ of the resources. In Section 4.5.3, we provide preliminary numerical results for the computational diversity gain.

4.1.4 Load balancing

The previously explained computational diversity allows for a computational load balancing. The goal of load balancing is to distribute the dynamic workload across the multiple processing nodes, to achieve optimal resource utilization and to avoid computational overload. It prevents bottlenecks of the system that may occur due to overburdened nodes, and further helps in promoting equal availability of computational resources. One of the most important challenges in implementing load balancing algorithms for the Cloud-RAN comes from the highly variable computational complexity of the various functionalities that have to be implemented.

Load balancing algorithms follow different classifications, according to whether the workload is distributed between the processing nodes in a static, dynamic, or adaptive manner [7]. In the static approach, the load balancing is defined when the system is implemented. The dynamic approach takes into account the current state of the system during load balancing decisions. The adaptive approach further allows dynamically changing the properties of the implemented functionality (e.g. switch from an optimal to a suboptimal algorithm) according to the state of the system when the load balancing decisions are made. The adaptive approach seems more appropriate in the Cloud-RAN context, since the computational load can vary significantly in time, due to fluctuations in the traffic load. Dynamically adapting the implemented functionalities to the computational load allows for further optimizing the use of the available resources in the Cloud-RAN.

Load balancing can also be used to implement failover [6] – the continuation of a service after the failure of one or more of its components. The components are monitored continually, and when one becomes non-responsive the load balancer is informed and no longer sends traffic to it. This is an inhered feature from grid-based computing for cloud-based platforms.

Another issue that can be addressed by using load balancing algorithms is related to energy optimization [8]. In traditional server cluster systems, the workload is distributed in an equal fashion in order to achieve the best possible performance and scalability. However, distributing the work across many servers may result in low levels of utilization, thus yielding excessive energy consumption with respect to the amount of useful work done. The reason is that the power consumption of current systems is not proportional to how much work they are doing, with low levels of utilization incurring disproportionate amounts of energy. In [5] , it has been pointed out that it may be possible to rewrite load balancing algorithms to be more energy aware and introduce the concept of "load-skewing". If servers were continually allocated work while they have resources remaining, then we would be able to power down unused servers and therefore save on energy consumption. Switching off or powering down components and entire systems effectively when not in use can be considered a key area of energy aware computing. However, the effect and extent of these power state transitions requires careful consideration. For example, powering down a CPU can be an effective means of saving energy. Suspending also the system cache, memory and controllers will save even more energy, but at the penalty of increase cost and time to return the system to a useful state [8]. A balance must be achieved between energy savings and system performance.

4.1.5 Migration of Virtual eNodeBs

The virtualization of physical resources allows for a virtually unlimited availability of resources and the possibility to provide resource on-demand. Although the user of a virtualized environment does not need to care about the actual physical deployment and assignment of virtual machines, the operator of the cloud-platform needs to take care of it. In particular, the following events require special attention:

1. Maintenance of physical resources which requires to turn off part of the infrastructure,
2. Increased resource demand by one particular virtual machine,
3. Failure of equipment.

All three events require a migration of virtual machines across physical resources, i.e. the assignment of physical resources such as memory and CPUs to virtual machines needs to be changed. During such migration, the system may not be used which implies that the downtime needs to be reduced to a minimum. Furthermore, all relevant data must be migrated such that the virtual machine can continue its service seamlessly for the user of the virtual machine and the operator of it. The process of migration may introduce dependencies between source and target virtual machines. Those dependencies may prohibit to finish the migration process but require to keep the source virtual machine active. Hence, those dependencies should be minimized, in particular in the time domain.

One possibility to implement a migration is to copy the full virtual machine. In this case, while the source virtual machine is still running, it is copied (i.e., memory pages) towards the destination. As soon as this process is completed, the source virtual machine is stopped. Then, two possibilities exist. Firstly, all remaining changes may be copied before the target virtual machine is started. Alternatively, the target virtual machine is started and remaining changes are copied on-demand. While the latter choice allows for very quick migration, the former one reduces dependencies after migration and is therefore more deterministic. Finally, after starting the target virtual machine, re-routing of incoming traffic needs to arranged, i.e., a

buffer needs to hold all incoming traffic, after start of the virtual machine the buffer needs to be emptied, and all future incoming packets needs to be routed to the target virtual machine. This migration method is applied particularly for long-term events such as maintenance and under-/overloaded physical resources which may be turned off/on.

Copying a complete virtual machine requires significant resources for the migration and it does not allow for scaling the actual resources. Alternatively, virtual machines may be grouped while maintaining a common abstract interface to the user of this group. In terms of virtual eNBs, one virtual eNodeB may be composed of multiple virtual machines which appear as one black box. In this case, it may be possible to increase or decrease the amount of consumed resources through adding or removing virtual machines. For this process, a template of new virtual machines and there state is required. Furthermore, the hypervisor of the cloud-platform must be able to support this scaling. This method is usually applied for short-term events when resources for one group of virtual machines need to be increased or decreased.

Finally, a failure recovery mechanism is required. This mechanism needs to ensure continued service of user terminals if a virtual machine fails. This can be done in two ways. Firstly, a stand-by copy of each virtual eNB is maintained and ready to be used at any time. Obviously, this would require significant resources and contradicts the idea of improving the utilization efficiency. As an alternative, 3GPP mechanisms could be used to re-connect user terminals to a new virtual machine after a failure.

In the case of virtual eNBs, the following cases of migration need to be considered:

1. Reassigning UEs between virtual eNodeBs at the same RANaaS entity
2. Reassigning UEs between virtual eNodeBs at different RANaaS entities
3. Reassigning iSCs between different veNBs at the same RANaaS entity
4. Reassigning iSCs between different veNBs at different RANaaS entities
5. Moving complete veNBs between different RANaaS entities

These cases will be investigated and explained in the upcoming deliverable D5.2.

4.1.6 Implementation requirements

iJOIN does not envision a monolithic porting of the 3GPP LTE stack into the veNB; on the contrary, ittargets a modular and even dynamic environment, where only the best suited part of the stack functions is executed into the veNB hosted in the RANaaS platform.

Generally speaking, IaaS platforms provide virtual machines as a natural mechanism for implementing modularity and, in this perspective, programs implementing CTs functionality can be integrated inside proper virtual machine images, so that, whenever we need to activate a new instance of the CT, it is only a matter of creating and activating a new virtual machine configured with the right software.

The actual feasibility or effectiveness of the implementation depends upon some key parameters. Such parameters are different for different candidate technologies, being tied to the characteristics of each algorithm in terms of distribution, computational intensity and timing.

In addition, when considering the porting of CTs into an IaaS platform, it is fundamental to consider some limitations imposed by the very nature of the target environment.

Processing power can be a critical parameter for CPU bound algorithms, since general purpose CPUs like the ones powering industry standard servers can’t generally reach the top processing performance rates which a DSP (or even more an ASIC or a FPGA) can achieve. The limitation is both in the CPU own computational power, and in the fact that industry standard servers don’t execute microcode but software programs whose interaction with the processor is mediated by an operating system, and are written in non-machine languages which poses a performance penalty.

This aspect could also be worsened when using virtualization, a foundation technology of cloud computing. As already mentioned in Section 4.1.2, virtualization implements virtual hardware resources by using the physical underlying resources (e.g., RAM, CPUs, disks, etc.). For example, with server virtualization, it is possible to run multiple virtual servers on a single physical computer and the amount of CPUs required for running all the virtual servers can be higher than the actual CPUs actually available. In such a case, virtualization transparently shares the actual CPUs among the virtual servers with mechanisms similar to

time sharing computing paradigm. This can raise issues when running real-time applications because a critical computation allocated to a VM could be periodically interrupted by the hypervisor for permitting other concurrent VMs to proceed. Cloud computing can partially address this issue by 'regulating' the actual number of VMs running on a single physical server and ensuring that the total amount of resources required for running does not exceed the actual amount of the available resources (e.g. number of CPUs).

However, as reported by [9], hypervisors introduce significant overhead on interrupt management that results in higher latency in respect to situations where the processing is executed on 'bare metal' computing environment. This happens because when an interrupt occurs, the hypervisor must dispatch the event to the 'right' VM, the VM that was originally waiting for it. For example, when a network packet is received on a network interface, it must be dispatched to the VM that was waiting for it.

Results in [9] show that the typical latency to interrupt on a virtualized environment with Kernel-based Virtual Machine (KVM) hypervisor [43] ranges from 300 to 700 µsec against a typical latency of 20 µsec on non-virtualized environments (Intel® Romley Server with 2 Intel® Xeon® CPU E5-2697 v2 processors,70GHz 12 cores x86_64 architecture). This is not necessarily a problem as it strictly depends on the type of the applications running on the virtualized environment. The potentially more serious challenge is the non-real-time behaviour of commodity hardware while usually applied FPGA and DSP architectures are real-time systems. Furthermore, commodity may require more processing time which potentially exceeds the achieved processing times on FPGAs and DSPs. The problems, higher computational latency and jitter, need to be solved in order to implement a 3GPP RAN system on commodity hardware.

To summarize, with the current state of art (which is going to deeply change in the forthcoming years) the most CPU bound and/or real-time candidate technologies could have issues to be centralized in the RANaaS.

4.2 Implementation constraints of 3GPP LTE

The previous section has addressed the capability of a cloud made of general purpose CPUs to perform RAN functionalities, with the PHY layer being the most extensive computational task. If nothing prevents one functional split to be applied in theory with any kind of backhaul, there are still strong timing constraints to consider if the 3GPP LTE compliancy is targeted. The lower we perform the functional split in the layer, the more bandwidth is required to support the forwarding of the data between the iSC and the RANaaS platform. In addition, the split point within the PHY processing chain itself (see Figure 4-6) can also impact greatly the required bandwidth as shown in [39] and IR2.2 [10], [39].

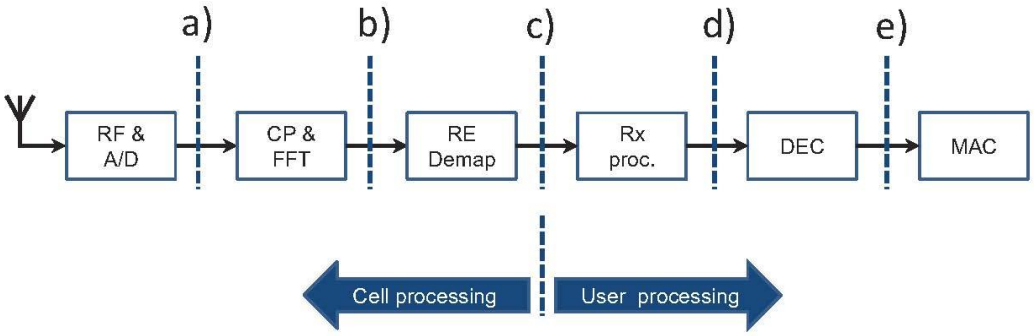


Figure 4-6: Functional split options for the PHY layer [39]

More important than the bandwidth requirements, the timing requirements must be considered in the functional split decision. Indeed, 3GPP has defined many timers for each of the layer (from MAC to RRC) which dictate the behaviour of the complete LTE system. They may impose some serious constraints on the feasibility of one specific functional split within a legacy 3GPP LTE ecosystem. In IR3.2 [11], [40], those timers have been all gathered. Table 4-1 only presents the ones we identified which possibly impact a functional split decision. Many of the higher layer timers are configurable with a specified range definition large enough to allow for a setting adapted to the backhaul and processing time required.

Table 4-1: 3GPP timing requirements [40]

	Timer	Short description	Max Value
PHY	Subframe	Physical subframe length	1 ms (fix)
	Frame	Physical frame length	10 ms (fix)
MAC	HARQ RTT Timer	When an HARQ process is available	8 ms (fix)
RLC	t-PollRetransmit	For AM RLC, poll for retransmission @TX side	500 ms
	t-Reordering	For UM/AM RLC, RLC PDU loss detection @RX side	200 ms
	t-StatusProhibit	Prohibit generation of a status report @RX side	500 ms
PDCP	discardTimer	Discard PDCP SDU / PDU if expiration or successful transmission	Infinite
RRC	TimeToTrigger	Time to trigger of a measurement report	5.12 s
	T300	RRCConnectionRequest	2 s
	T301	RRCConnectionReestablishmentRequest	2 s
	T304	RRCConnectionReconfiguration	2 s or 8 s
	T310	Detection of physical problem (successive out-of-sync from lower layers)	2 s
	T311	RRC connection reestablishment (E-UTRA or another RAT).	30 s

4.3 Preferred functional splits

The objective of this section is to provide an initial assessment of the different implementation options in order to support the flexible functional split proposed by iJOIN. For these purposes, a three step process is required.

The first step in this process is to identify those functionalities that are best implemented on specialised hardware such as ASICs, FPGAs or DSPs, from those that may benefit (or not be significantly impaired) from an implementation on general purpose processors (GPPs). In the end, whether a functionality falls into one category or the other would depend on whether it is composed of repetitive tasks that can be accelerated if they are implemented in a specific hardware solution or would benefit from the possibility of supporting larger algorithmic complexity that is provided by a software based solution.

In the following, an identification of the LTE radio interface functionalities in terms of the optimal implementation option is carried out. In general, it can be assumed that splits that are user dependent will potentially provide statistical multiplexing processing gains when the traffic generation is not homogeneous. Furthermore, latency, throughput, and execution jitter need to be considered to decide upon the functional split. In general, 3GPP LTE is a real-time system and therefore hard deadlines must not be violated. However, the RAN protocol stack may be decomposed into a time-critical and a less time-critical part. The time-critical should preferably be implemented on hardware while less time-critical parts may be implemented in software.

Finally, a third step that needs to be performed is the evaluation of the time resilience of the solutions adopted, i.e. how well they can be adapted to the support of new functionalities to be incorporated in the evolution of the networks. Examples may be the support of massive MIMO solutions or full duplex communications as well as the incorporation of new operational frequency bands.

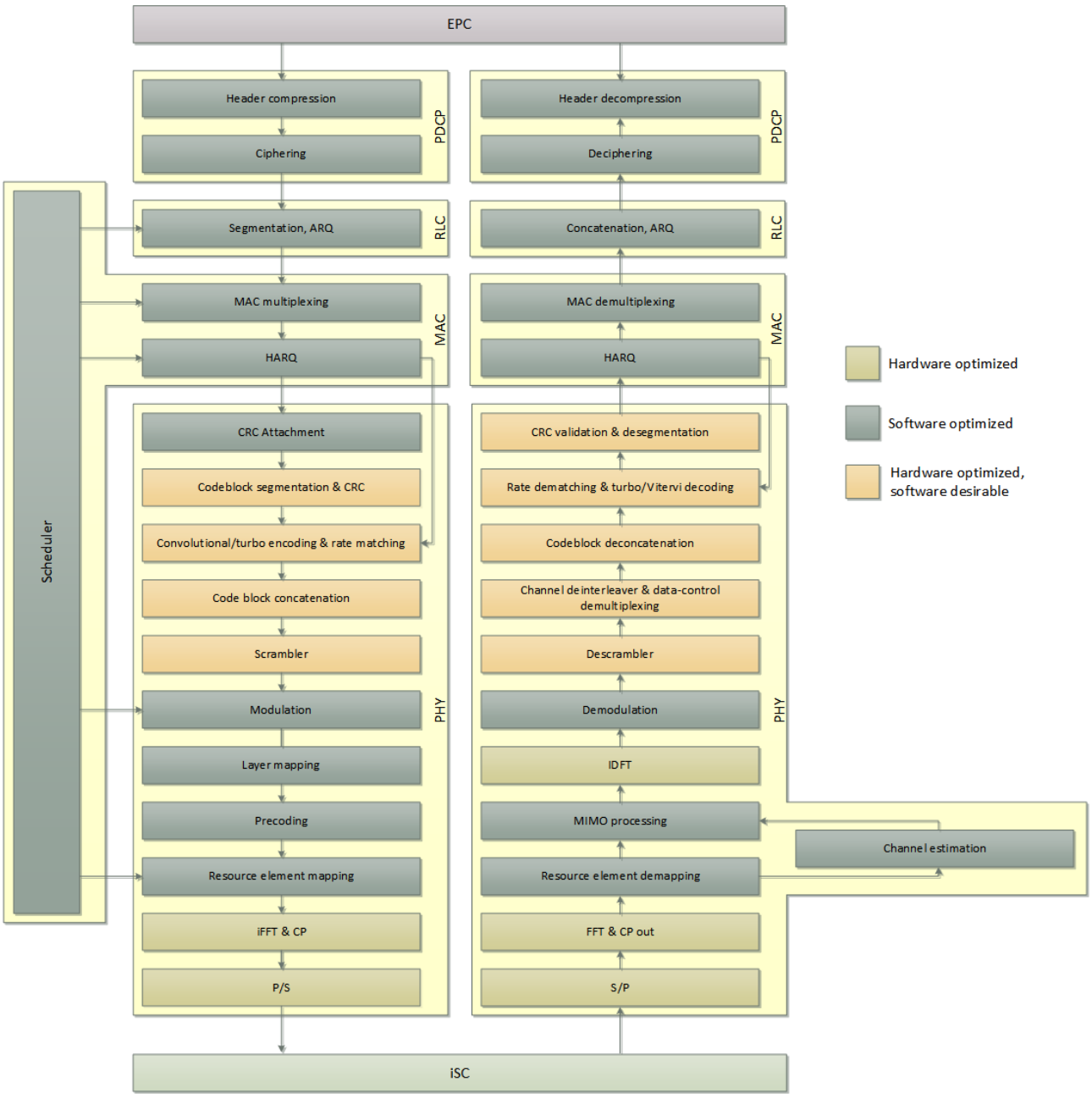


Figure 4-7: Implementation options of 3GPP LTE RAN functionality

Based on the previous discussion and the results in IR2.2 [10] and IR3.2 [11], the following three functional splits are candidates for more detailed investigations (see also Figure 4-8:):

A. Similar to CPRI (Common Public Radio Interface), most of digital processing is centralized. In this case, a very low latency high-capacity backhaul is required. Furthermore, this option does not allow for exploiting multiplexing gains in the backhaul. However, it offers most centralization gains through joint processing. A detailed analysis is provided in IR2.2 [10] as a result of work performed in iJOIN WP2.

B. In this case, user-based functionality is centralized including forward error correction while cell-specific processing such as FFT remains at the iSCs. This allows for exploiting multiplexing gains in the backhaul, computational diversity gains at the central processor, and centralization gains through multi-point algorithms. This split is further considered in iJOIN WP2 and WP3 [10], [11].

C. Finally, one option is to only centralize upper MAC functionality and part of the scheduler (mostly control-plane functionality). In this case, time-critical processing at the central processor is avoided while advanced coordination algorithms can be executed. This split is further discussed in iJOIN WP3 [11].

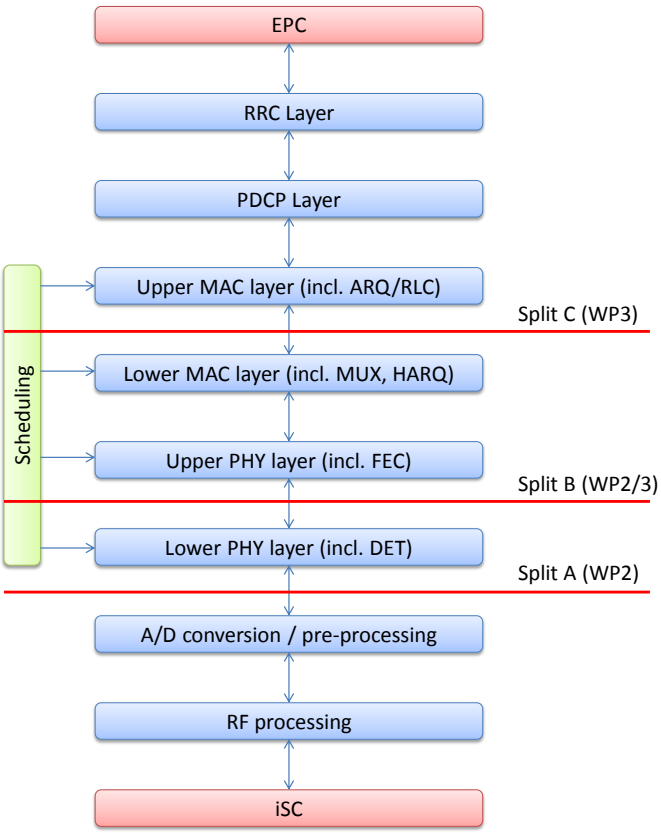


Figure 4-8: Preferred functional splits considered in iJOIN

4.4 Flexible functional split assignment

The objective of this section is to analyse, from a practical viewpoint, the actual flexibility that can be achieved in terms of functional split. Ideally, a fully virtualized environment where each function could be moved at any place would be feasible. In practice, there will be different small-cell implementations which may or may not support different functional splits, or different co-processors which can be turned on/off. In addition, the assignment of the RANaaS data centre cannot be assumed to be changed on the fly.

In practice this means that the set of supported functional splits will be a reduced set. If only one functional split would be supported, it will be the one that provides the best ratio of potential benefits associated to the centralization degree with respect to potential cost variations. The benefits that can be obtained from centralization are mainly associated to increased spectral and energy efficiencies. This ratio depends not only on technical factors but also on other factors, e.g. the traffic demand, user distribution, the possibility of reusing deployed infrastructure, etc.

For example, the centralization of the baseband processing may allow for an implementation of cooperation mechanisms that may help to improve the spectral efficiency, reducing the need for new deployments in high traffic demand areas and making it a sensible option from a techno-economic viewpoint. But if demand is relatively low (or other solutions, like using additional carriers, are available), then it may be that the opposite is reached.

On top of this, different technical criteria should also be taken into account.

- Cell based vs. user based processing
  One of the criteria to be used is that cell based processing should be distributed as far as possible, as it should reduce the transport requirements and does not exhibit potential processing multiplexing gains. On top of this, this processing is better implemented using hardware solutions.
- Software based processing vs. hardware based processing
  As has been indicated in previous sections, some functionality is more efficiently implemented by means of hardware based solutions, while others benefit from a software based implementation.

- Latency requirements
  Some processing functionalities are more sensible to latency than others. Obviously, this factor determines whether their potential centralization in the RANaaS infrastructure is feasible or not.

Based on this description and the previous section, it may feasible that the implementation at the iSCs split into two parts: a hardware implementation and a software implementation. Then, based on the actual functional split individual modules would be turned on and off. Each module may be composed of the functionality shown in Figure 4-8. Some of these modules may be implemented in hardware and some in software based on the recommendation in Figure 4-7. In a practical setup, an iSC may support only two, at most three, functional splits:

- A preferred functional split where functionality at the iSC is executed on hardware and all remaining functionality is executed in software at the RANaaS entity. The individual modules at the iSC may be implemented on different co-processors in order to allow for flexible functional split configurations based on a single hardware platform.
- A fall-back solution where all upper layers are executed in software at the iSC and no functionality is centralized.
- A centralization where the iSC executes part of the functionality in software while a smaller set of functionality is centralized.

The second option may be useful in the case that RANaaS entities fails or is overloaded, or if there is no need for centralized processing, e.g. in the case of low traffic. The third option may be used to reduce the load of RANaaS entity as well as and may only leverage from inter-cell coordination algorithms. Furthermore, the third option may allow for adaptation to the backhaul network and offer a way to different functional splits within one deployment.

4.5 Preliminary Results

4.5.1 Opportunistic HARQ

As early introduced, 3GPP LTE underlies tight timing constraints. One of the discussed split would centralize the decoding process while Radio Frequency (RF) processing is still performed at the iSC. This split of RAN functionality allows for implementing advanced decoding algorithms at the central processor and would centralize a major part of computational complexity.

However, in this case stringent latency requirements must be fulfilled, e. g., HARQ requires that all uplink processing must be finished within 3 ms after receiving a subframe (in 3GPP LTE frequency division duplexing (FDD)). These 3 ms include both the round-trip delay between central processor and iSC, as well as the actual decoding operation. Non-ideal backhaul may imply significantly higher latencies. In addition, deploying general purpose hardware at the central processor will lead to computational jitter violating real-time constraints.

Hence, we need a solution which is applicable to non-ideal backhaul (latency in the order of milliseconds), which allows for centralizing the computationally intense part of the PHY layer, which meets the stringent timing requirements imposed by HARQ, and which does not impose significant performance penalties. If we were applying currently available technology, only the backhaul round-trip delay would already exceed the HARQ timing requirements and therefore lead to RAN protocol errors. The solution must be standards compliant as any change particularly to mobile terminals should be avoided. Preferably, the solution only applies to deployed radio access points and is transparent to mobile terminals.

In the following, we provide numerical results for an opportunistic HARQ approach where the iSC estimates the probability of decoding success based on the received SNR. Using this estimate, the iSC sends HARQ feedback to the mobile terminal, and forwards the received packets as well as information on the HARQ feedback to the central processor. If this approach is applied, the central processor could then combine the received packets, taking into account the HARQ feedback provided by the iSC. The iSC needs not to decode any packet, and only deploys one mapping curve using an effective SNR based on the channel state information from all transmissions. Due to the fact, that only one mapping curve is used, the approach is independent of the number of HARQ retransmissions.

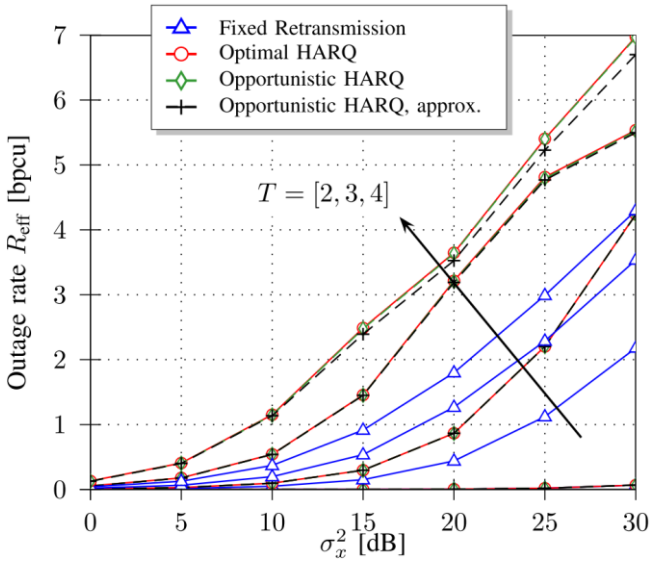


Figure 4-9: Achievable outage rate depending on the SNR for an outage probability of 0.1%

Figure 4-9 shows numerical results for the considered opportunistic HARQ approach. It shows the results for

- Fixed retransmission: For each codeword, T transmissions are independently encoded, transmitted and combined,
- Optimal HARQ: HARQ feedback is provided based on the actual decoding result,
- Opportunistic HARQ: HARQ feedback is provided based on the exact outage probability expression,
- Opportunistic HARQ, approx.: HARQ feedback is provided based on a single mapping curve which employs an effective SNR computed over all transmitted codeword. The effective SNR is given by

$$\gamma_{eff} = \max\left(\sum_{t=0}^{T-1}\gamma_t, (e/4)^T \prod_{t=0}^{T-1}\gamma_t\right). \tag{4.2}$$

The results shown were acquired for identical and independent block Rayleigh fading, 360 information bits, effective outage probability $\varepsilon = 10^{-4}$, and T=[1, 2, 3, 4] transmission rounds. The figure shows the effective spectral efficiency under the given outage probability constraint and considering the actual number of transmissions.

For T = 1, the effective rate of all four approaches coincide below 0.1bpcu and is therefore only recognizable at the bottom of the figure. The results show that opportunistic HARQ is able to maintain the benefits of HARQ and offers the same diversity gain. The benefits compared to a fixed number of transmissions would increase with decreasing outage probability. We can further observe that the effective SNR in (4.2) implies only a minor performance loss for T = 3 and T = 4 compared to optimal HARQ and opportunistic HARQ.

In currently deployed centralized RAN, a very high capacity and very low latency connection between RRH and central processor is required in order to provide HARQ feedback within the required time, i.e. 3ms in the case of 3GPP LTE FDD. Our approach divides the HARQ process into a time-critical part and computationally intense part. The time-critical part, i.e., determining HARQ feedback, is implemented at the RRH based on the channel state information and without the need to decode the received codeword. The computationally intense part, i.e., decoding the received codeword, is moved towards the central processor where advanced and computationally intense algorithms may be implemented. In addition, this implies that non-realtime commodity hardware may be deployed at the central processor, which would imply computational jitter.

Since the time-critical part has been removed from the central processor (at least the part of particular relevance to PHY and MAC), it is possible to relax realtime constraints and deploy general purpose processors. It further allows for using non-ideal backhaul in a centralized RAN architecture which is critical in areas of high deployment costs, e.g., small-cell deployments where trenching optical fibre would constitute a major part of the capital expenditures.

4.5.2 Computational Outage

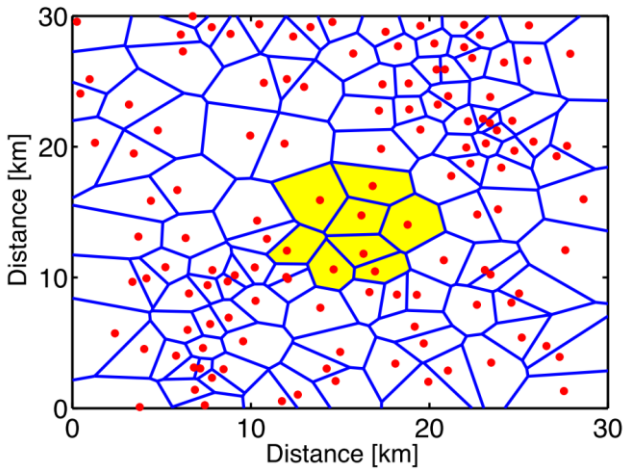
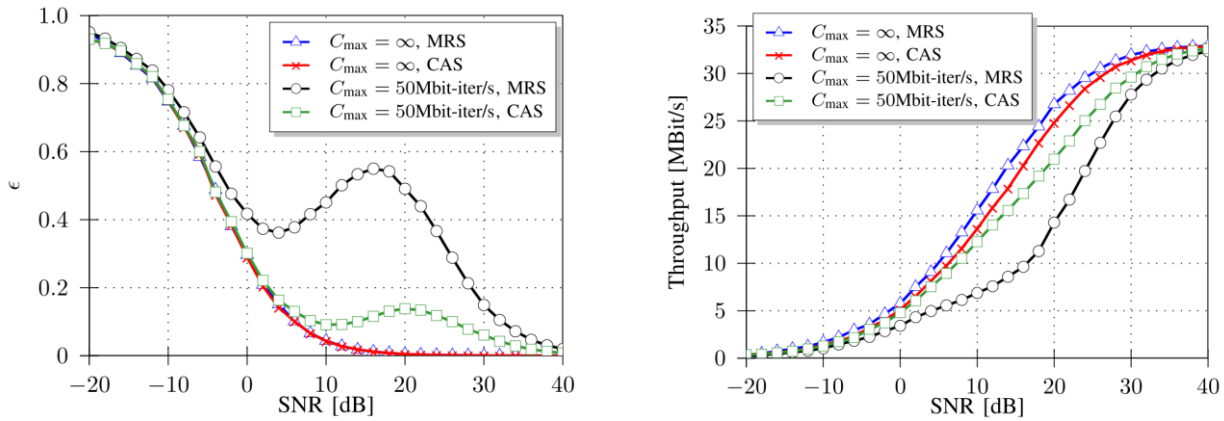


Figure 4-10: Considered network deployment

In the following, we provide numerical results for the impact of computational outage on the system. This analysis uses the network as illustrated in Figure 4-10 (originating from an actual deployment of a UK telecommunications operator). We further assume block Rayleigh fading, a simplified distance-dependent path-loss model, and that each base-station serves exactly one user on the whole bandwidth within one sub-frame. In future reports, we will extend this network model to be aligned with the evaluation assumptions of iJOIN.

Based on this network model, Figure 4-11 shows performance results for a single link ignoring any inter-cell interference. For these results, we limited the complexity to 50 Mbit-iterations/s which corresponds to about 6-12 processor cores. The left side shows the outage probability depending on the average channel SNR (corresponding with large-scale fading). We can observe a sudden increase of outage between 10-20dB. The reason for this increase is that the likelihood of higher modulation schemes increases where more computational power is required (see Figure 4-5). In the case of CAS, the outage is much lower than in the case of MRS. The right hand-side of Figure 4-11 shows the effective throughput of both approaches. We can see that in the case of limited computational resources, CAS provides higher effective throughput than MRS although the latter uses more spectral efficient modulation and coding schemes. There reason is the outage probability in the practically relevant region of 10-20dB.



(a) Overall outage probability **(b) Effective throughput**

Figure 4-11: Results for single-cell under a computational complexity constraint

In Figure 4-12, we show the expected sum-throughput as a function of the maximum normalized per-RAP computational complexity and for eight centralized iSCs. We can observe that in the case strong computational limitations, the centralized CAS (CAS, CP) outperforms all other approaches, e.g. if local processing and MRS is applied, the maximum performance is only achieved with more than Mbit-iter/s while CAS with local processing as well as MRS with central processing achieve their maximum performance with 60 Mbit-iter/s, and CAS with central processing requires only 25-30 Mbit-iter/s.

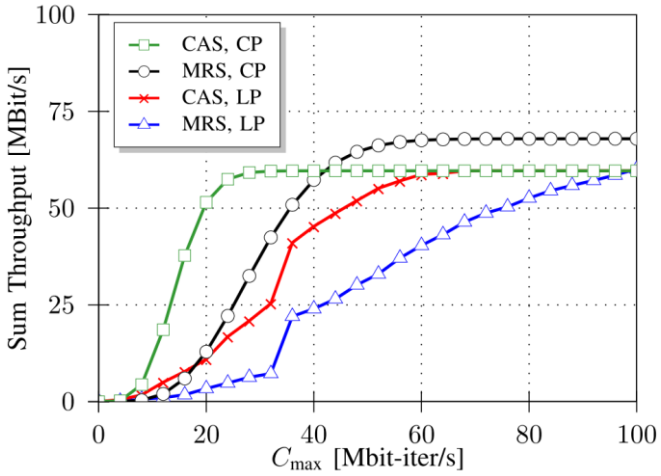


Figure 4-12: Results for multi-cell network for different computational complexity constraints

These results show that a RAN implementation on cloud-platforms requires an intelligent design of channel scheduler and resource scheduling in the virtualized infrastructure. It is necessary that both are aware of each other in order to optimize the throughput performance and resource usage. In the next subsection, we further elaborate on the resource usage by showing the scaling behaviour of computational diversity gain.

4.5.3 Computational Diversity

In Section 4.1.3, we introduced the idea of computational outage and computational diversity $c(N)$. In order to evaluate both quantitatively, we applied and extended the complexity model which was introduced in [14]. In Figure 4-13, we show both the measured curves using system level evaluations (blue line) as well as the theoretical complexity (black line). We use as a measure of complexity again bit-iterations, here, normalized to a channel use.

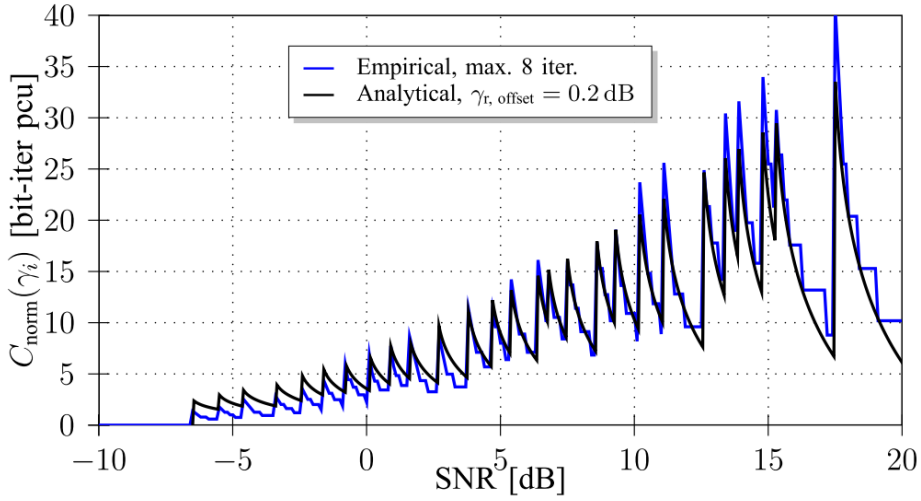


Figure 4-13: Numerical and analytical complexity model for 3GPP LTE uplink

Based on this model, we derived the expected computational complexity at a single base-station as a function of the decoder quality and the number of MCS schemes. The decoder quality is represented by a constant SNR offset between theoretical Shannon AWGN capacity, $C = \log(1+\gamma)$, and the actual link-adaptation curve. Therefore, smaller rate-offset will be closer to Shannon capacity but also requires more computational complexity. Furthermore, the number of MCS schemes impacts both complexity and achievable rate, i.e., the more MCS schemes are employed, the closer we operate to Shannon's capacity which drives complexity but also improves spectral efficiency. In 3GPP LTE, we would apply the MCS schemes and, in this model, a rate-offset of 0.2dB provides results close to our numerical results.

Figure 4-14 shows the complexity scaling in the number of schemes as well as the rate-offset. Apparently, as we apply more MCS schemes, also the complexity increases. However, the more significant complexity increase is observed when the rate-offset is reduced. In this case, the complexity increases significantly.

Furthermore, the results compare numerical results (solid line) and results of an approximating analytical framework (markers).

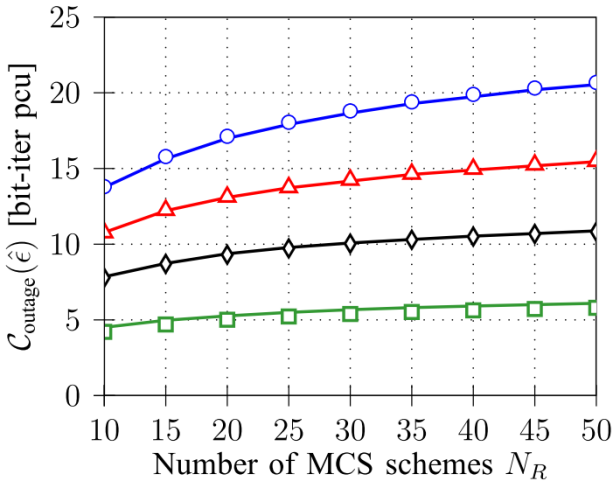
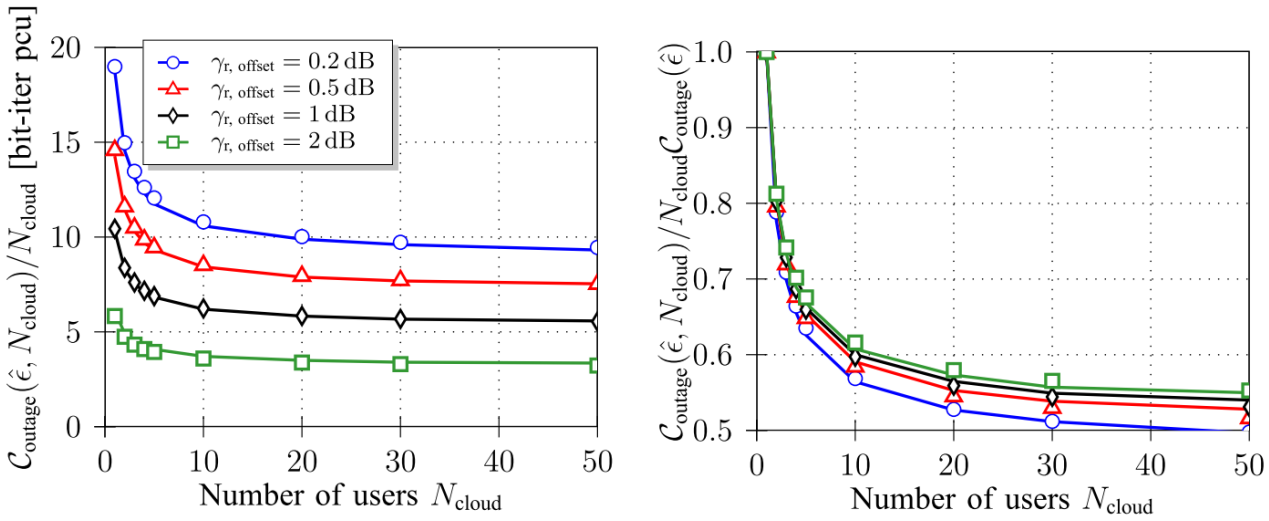


Figure 4-14: Expected computational complexity for one cell

This model is further extended to a network of cells. We assume that each cell serves exactly one user, each user experiences block Rayleigh fading at mean SNR of 10dB. Furthermore, we normalize the per-cell computational outage as described in Section 4.1.3. In Figure 4-15(a), we can see the absolute normalized complexity $C_{outage,N}(\varepsilon)/N$ for varying decoder quality and different number of cells (or users, respectively). We can see that already for a small number of cells, the full computational complexity is exploited, i.e. for $N > 10$ the computational complexity decreases slowly with the number of users while for $N < 10$ the complexity is reduced significantly. This is again illustrated in Figure 4-15(b) which shows $1/c(N)$ as defined in Section 4.1.3. By contrast to the absolute complexity, we can see that the relative gain is rather independent of the decoder quality. Again a saturation is seen at $N = 10$ to 20 where about 50-60% of the computational resources of a distributed system is required.

Note that the presented results are subject to limitations and assumptions which will be addressed in the forthcoming report:

- Assumption of fully loaded cells; so far, we assume that each cell is fully loaded and we do not include the inherent multiplexing gain of Cloud-RAN. This gain will further improve the computation diversity gain.
- Static mean SNR for block Rayleigh fading process; we will extend this model to more realistic channel fading processes where also the mean SNR varies (path-loss and shadow fading).



(a) Absolute normalized complexity in bit-iterations per channel use

(b) Relative normalized per-cell complexity

Figure 4-15: Scaling of computational complexity as a function of number of users/cells

5 Joint Radio Access and Backhaul Network Support

5.1 Required interfaces and interaction

In this section we describe the interfaces required for a joint RAN/BH operation. In particular, we focus on the interaction required between iTN, iSC, iNC and RANaaS. We will give a special emphasis on the timescale on which information is exchanged. The timescale is a key parameter for the integration of the candidate technologies. D5.2 will report a detailed description about the integration of candidate technologies by timescales and bandwidth requirements.

CT2.1 In-Network Processing

- Each iSC obtains network information (nodes, available links, etc.) and RRM information per UE and RRM information per BH link from iveC (located in RANaaS).
- iSCs iteratively exchange messages with each other over J2 link for distributed MUD (~1ms).
- iSCs deliver estimated user messages over J1 link to RANaaS (~1ms).
- iSCs deliver UL channel information over J1 link to iveC located in RANaaS (~1ms).

CT2.2 Multipoint Turbo Detection

- iSC-RANaaS J1 interface is required (~1ms for MPTD, otherwise SPTD will be used).
- iSC-iSC link is required for SPTD only (<1ms, either X2 or J2 interface).
- Other network entities are beyond the scope of CT2.2 (see CT3.7). Only iveC may be added once its status/place will be defined.

CT2.3 Joint Network-Channel Coding

- The network-coding messages are sent from the relaying SC to the destination SC through the J2 interfaces, or X2 in case the destination is represented by the eNB. The latency depends on the HARQ ACK/NACK constraint (<6ms).
- If the centralized decoding at RANaaS is used, then the destination SC or eNB send to the RANaaS the decoded streams through interfaces J1 and X1. In this case the latency due to J2 plus two-way J1 latency depends on the HARQ ACK/NACK constraint (<6ms).
- J2 interface is used also for sharing CSI (~1ms)

CT2.4 Sum-Rate and Energy-Efficiency Metrics of DL COMP with backhaul constraints

- Each cooperating iSC requires global CSI which could be obtained by exchanging local CSI with each other over J2 interface, operating on the same timescale (i.e. ~1 ms, depending on channel coherence time).

CT2.5 Partially Centralized Inter-Cell Interference Coordination

- It is required to have CSI feedback from the UE and to exchange this CSI with low delay (few ms) between iSCs (J2) and between an iSC and the RANaaS (J1).
- RANaaS delivers user data through the J1 interface to the iSCs.

CT2.6 Data Compression over RoF

- RANaaS delivers user messages over J1 interface to iSCs for DL transmission (~1ms).
- iSCs delivers estimated user messages over J1 interface to RANaaS for UL transmissions (~1ms).
- iveC function in the RANaaS may command changes in PHY functional split between RANaaS and iSC as a function of e.g. radio interface load, available RANaaS computation capability and available backhaul capacity on a medium/long terms basis (e.g. minutes to hours).

CT2.7 Millimetre wave backhauling

- The uncoded BH scheme makes use of the adaptive modulation and coding scheme already in place in LTE and operates on the same timescale (~1ms). Depending on the functional split, sampled I/Q data or pre-processed user data is forwarded from the iSC to the RANaaS.

- Additionally, channel state information of the mmWave channel, e.g. in the form of an SNR, is required. Since a directive wireless BH link faces neither varying multipath nor Doppler effects, the BH's SNR can most probably be updated on a lower timescale (~1s).
- The channel state can be measured at the mmWave receiver and is required at the joint RAN/BH decoder. Depending on whether both functionalities are co-located or not, an exchange of this CSI via a separate interface, e.g., an interface between an iTN and the RANaaS, is required.

CT3.1 Backhaul Link Scheduling and QoS-aware Flow Forwarding

- J1: iSC delivers information to RANaaS (~1s).
- J1: RANaas delivers information to iSCs (~1s).
- J3: RANaas delivers information to iNC (~1s).
- J3: iNC delivers information to RANaaS (~1s).

CT3.2 Partly decentralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments

- J1: iSC delivers information to RANaaS (~1s).
- J1: RANaas delivers information to iSCs (~1s).
- J3: RANaas delivers information to iNC (~1s).
- J3: iNC delivers information to RANaaS (~1s).

CT3.3 Energy-Efficient MAC/RRM at Access and Backhaul

- J1: iSC delivers information to RANaaS (~100ms).
- J1: RANaas delivers information to iSCs (~100ms).

CT3.4 Computational Complexity and Semi-Deterministic Scheduling

- J1: iSC delivers information to RANaaS (~20ms)
- J1: RANaas delivers information to iSCs (~20ms)

CT3.5 Cooperative RRM for Inter-Cell Interference Coordination in RANaaS

- iSC delivers information to RANaaS (~1ms).
- RANaas delivers information to iSCs (~1ms).

CT3.6 Utilization and Energy Efficiency

- Not available for this CT since it does not require any interaction.

CT3.7 Radio Resource Management for Scalable Multi-Point Turbo Detection

- J1: iSC delivers information to RANaaS (ideally ~1ms, otherwise up to ~50ms).
- J2: iSC delivers information to other iSCs (ideally <1ms).
- RANaas delivers information to iveC (~1ms).
- J3: iveC delivers information to iNC (ideally ~1ms, otherwise up to ~1s).
- J3: iNC delivers information to iveC (ideally ~1ms, otherwise up to ~1s).

CT3.8 Radio Resource Management for In-Network-Processing

- J1: iSC delivers information to RANaaS (~1ms).
- J1: RANaas delivers information to iSCs (~1ms).
- J3: iNC delivers information to RANaaS (~1ms).

CT3.9 Hybrid local-cloud-based user scheduling for interference control

- J2: iSC delivers information to other iSCs (ideally <1ms).
- optionally J1: iSC delivers information to RANaaS (~1ms).
- optionally J1: RANaas delivers information to iSCs (~1ms).

CT4.1 Distributed IP Anchoring and Mobility Management

- iSC notifies to the iNC the event of an attachment of a UE (whenever an UE attaches to an iSC).

CT4.2 Network Energy Optimization

- iNC delivers information to iSCs (whenever the iNC decides to switch off/on some iSC nodes).
- iNC delivers information to iTNs (whenever the iNC decides to switch off/on some iTN nodes).
- iSC delivers information to iNC (~1min).
- iTN delivers information to iNC (~1min).

CT4.3 Joint Path Management and Topology Control

- RANaaS delivers information to iSCs/eNBs (whenever a re-association between iSC and RANaaS is required).
- iNC delivers information to the RANaaS (~1h).
- iNC delivers information to iTNs (~1h).

CT4.4 Routing and Congestion Control Mechanisms

- iTN delivers information to the iNC (the communication is based on events related to the thresholds implemented in the CT).
- iNC delivers information to the RANaaS (the communication occurs when changes of the functional split implemented are required to reduce congestion).
- iNC delicers information to the iSC (the communication occurs when changes of the functional split implemented are required to reduce congestion).

CT4.5 Load Balancing and Scheduling

- iNC delivers information to RANaaS (the communication occurs when changes in the current paths are required to increase utilization efficiency).
- iNC delivers information to iTNs (the communication occurs when changes in the current paths are required to increase utilization efficiency).

CT4.6 Backhaul Analysis based on Viable Metrics and "Cost" Functions using Stochastic Geometry

- The interaction between network's nodes is not available for this CT since it does not require any interaction. Although there is no interaction, the deployment cost analysis is typically done during new network roll-out or during network expansion. So, the timeframe would be months or years.

5.2 Limitations in 3GPP LTE

5.2.1 3GPP interfaces and requirements

3GPP considers the transport network underlying the mobile network as out of scope of its standardization focus. Consequently, 3GPP specifications are in general agnostic to transport network technologies and, in particular, 3GPP assumes that underlying transport networks are not contended. They are therefore assumed to satisfy the requirements for network operation.

In the mobile backhaul, the following traffic types based on 3GPP interface definitions can be differentiated:

- S1-U traffic destined for the S-GW; note that S1-U traffic can be further differentiated according to the assigned QCI value;
- S1-C traffic destined for the MME;
- X2-U and X2-C traffic destined for other eNodeBs;
- OSS (operations support system) traffic destined for core applications that provide fault, configuration, and performance management;
- Network synchronisation traffic.

All these traffic types have different requirements regarding QoS, where it can be generally stated that control plane traffic (e.g. S1-C, X2-C, and synchronization traffic) have higher requirements in terms of latency and reliability, but have a lower demand on bandwidth compared to user-plane traffic (S1-U and X2-U).

In today's networks, traffic differentiation for 3GPP traffic types is implemented via traffic type (e.g. control plane/user plane) and traffic class (e.g. based on QCI) mapping on transport network traffic differentiation techniques, which depend on the employed transport network technology. For example, legacy ATM defines four different traffic classes which describe bandwidth requirement characteristics such as constant bit rate or variable bit rate. However, no delay requirements are specified. In LTE-Advanced, all-IP networks with layer 3 routing/VPN technologies (e.g. MPLS) or QoS and IP-aware layer 2 switching technologies (e.g. based on 802.1q/p) are expected to play a larger role due to the availability of Ethernet-capable eNodeBs in the access network and corresponding cost benefits.

While 3GPP defines a set of standardized QCI values [24], there is no standardized guideline available on how mobile network traffic is mapped to service classes on the transport layer. The problem is amplified by differences in the implementation between different vendors.

To illustrate the challenge, Table 5-1 shows the standardized QCI values in 3GPP for different service classes. The quantitative parameters include the packet delay budget and the packet error loss rate, both referring to the overall connection from access to core or vice versa. In Table 5-2, IEEE 802.1Q Priority Code Point (PCP) recommendations are shown (note that there are no standardized parameter sets), which are often applied to Ethernet or similar link technologies in the backhaul network. The challenge is now to map QCI traffic to according PCP values, which additionally need to be parameterized appropriately.

Table 5-1: 3GPP standardized QCI values

QCI	Resource Type	Priority	Packet Delay Budget	Packet Error Loss Rate	Example Services
1	GBR	2	100 ms	10^{-2}	Conversational Voice
2		4	150 ms	10^{-3}	Conversational Video (Live Streaming)
3		3	50 ms	10^{-3}	Real Time Gaming
4		5	300 ms	10^{-6}	Non-Conversational Video (Buffered Streaming)
5	Non-GBR	1	100 ms	10^{-6}	IMS Signalling
6		6	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
7		7	100 ms	10^{-3}	Voice, Video (Live Streaming) Interactive Gaming
8		8	300 ms	10^{-6}	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
9		9			

Table 5-2: IEEE 802.1Q Priority Code Point recommendations [25]

PCP	Priority	Acronym	Traffic Types
1	0 (lowest)	BK	Background
0	1	BE	Best Effort
2	2	EE	Excellent Effort
3	3	CA	Critical Applications
4	4	VI	Video, < 100 ms latency and jitter
5	5	VO	Voice, < 10 ms latency and jitter
6	6	IC	Internetwork Control
7	7 (highest)	NC	Network Control

It can be concluded that neither 3GPP nor other standardization bodies offer a standardized methodology on how to map interface and protocol requirements of the mobile network to the backhaul network. Configuration is thus a case-by-case issue which needs fine-tuning for each deployment and equipment scenario.

5.2.2 Impact of centralization and coordination

As indicated in Sections 4.2 and 5.1, centralization in RANaaS changes the requirements on backhaul characteristics both for latency as well as for bandwidth KPIs. As a rule of thumb, both the bandwidth and latency requirements become stronger (i.e. lower average latencies, lower jitter, higher bandwidth) if the functional split moves down in the protocol stack. A pivotal point is the split between MAC and PHY layer, where the requirements on latency become realtime requirements (that is, a deterministic deadline must be fulfilled) of at least 1 ms. On MAC and higher layer, the lowest latency requirement from a protocol point of view is determined by the HARQ processing, which is around 4 ms in LTE.

Nevertheless, some CTs on MAC/RRM layer also require a fast transmission of control or feedback data between the centralized RAN function in RANaaS and the iSCs. Fine-granular scheduling and coordination schemes (e.g. in CT 3.5) need resource allocation to be completed within a subframe time of one ms. However, missing deadlines may be less disruptive to the system operation than on PHY layer, since the frame construction could still be performed although potentially without resource allocation.

In summary, the requirements with centralization tighten. Quantitative values depend on the functional split configuration and on the set of employed CTs in the system.

5.2.3 Recommendations

It can be concluded that for 3GPP, the ongoing discussion on the impact of virtualization both in core network and RAN is an opportunity to also discuss the necessity of standardized interfaces or mechanisms on how to deal with varying backhaul characteristics.

Within iJOIN, a coordination of backhaul characteristics and CT functions is foreseen in the interplay between the SDN-based iNC and the iveC, which is potentially deployed in the RANaaS platform. The coordination has to take into account the functional split, the CT requirements and the backhaul characteristics.

5.3 Preliminary results

5.3.1 Joint RAN/BH Coding

In current mobile networks, RAN and BH links are often perceived as separate links in terms of coding. The RAN link is encoded to fit the channel quality of the radio access channel and BH link is encoded according to the BH channel quality. While a BH channel code is not required in the fibre-based CPRI backhaul, it is important when using outdoor E-band links as they face varying channel conditions due to e.g. rain. In CT2.7, iJOIN investigates the possibility of jointly en- and decoding both links in the uplink. If the decoding of the RAN link is offloaded to the RANaaS, then the data on the BH link is already protected by the RAN

channel code. By adapting the code rate of the RAN channel code not only to the RAN channel's quality but to that of the BH link as well, a second en-/decoding is unnecessary. Further details on this can be found in IR2.2 [10]. This not only reduces the required hardware in the iSCs but also reduces the latency. Every additional processing performed on the BH increases the overall latency between the UE and the RANaaS. If the decoding is offloaded to the RANaaS, all timing constraints discussed in Section 4.2, especially the tight constraint for the HARQ acknowledgement, have to be met by the combined RAN/BH transmission, which is why it is important to keep the latency added by the BH to a minimum.

The joint encoding integrates very simply into the current standard. The RAN code rate is decided by the (v)eNB by taking the current frame error rate into account and communicating the decision in form of DCI (Downlink Control Information) information during the UL grant. A low-quality BH link increases the frame error rate, which will be noticed by the veNB and adjusts the code rate accordingly, which is depicted in Figure 5-1.

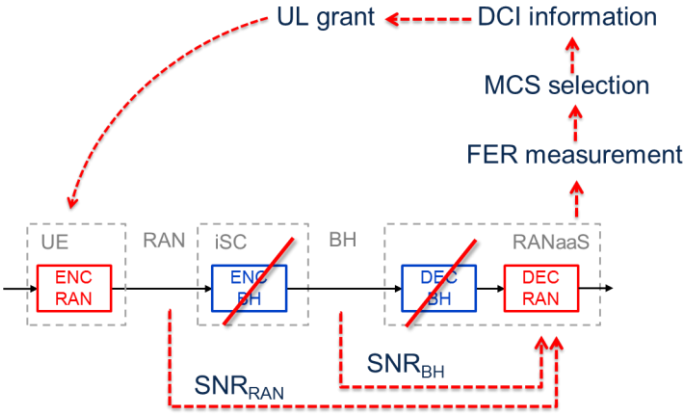


Figure 5-1: Code rate adaption and channel quality measurements required for joint RAN/BH en-/decoding

The joint encoding/uncoded BH scheme slightly decreases the throughput compared to using a separate BH code [10], which can be seen from Figure 5-2 when comparing the dashed to the solid lines. However, the additional BH transmission should not decrease the end-to-end throughput of the system. To mitigate the lower performance of joint decoding, a soft-input/soft-output dequantizer (SISODQ) can be employed, which enables forwarding soft information between the demodulation modules of the RAN and the BH link [23]. This increases the throughput even beyond the value of a separately coded BH, which can be seen from the dotted lines in Figure 5-2. The increased throughput can also be traded off for energy efficiency by using lower transmit power on the BH As investigated in, the SISODQ is faster than an additional BH en-/decoder under certain circumstances, thereby reducing the latency as required. However, the exact latency depends very much on the actual implementation.

In conclusion, CT2.7's approach on joint RAN/BH coding integrates easily into the RANaaS architecture, enables the low latency required for centralized processing, and reduces the complexity of the iSCs, while at the same time increasing throughput or energy efficiency of the network.

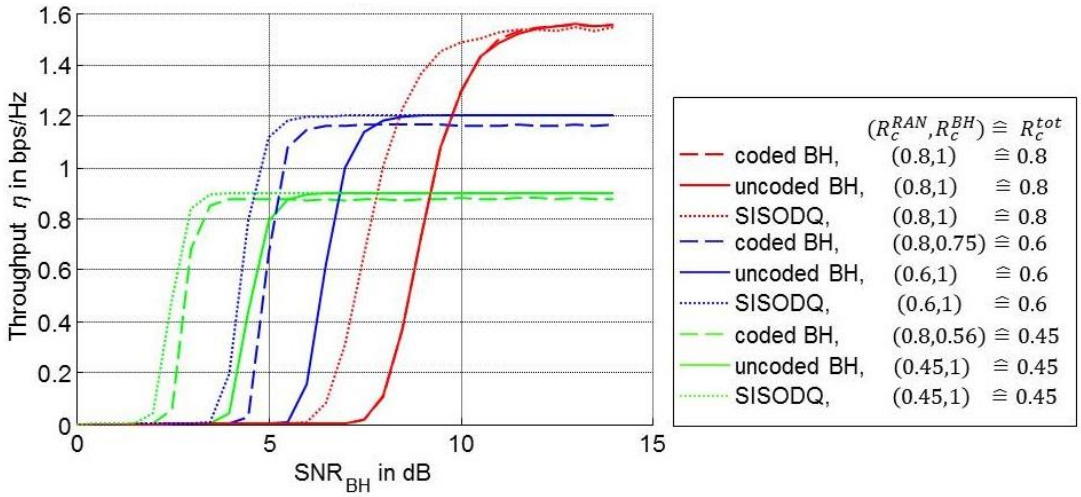


Figure 5-2: Throughput when using encoded BH (dashed lines) as compared to an uncoded BH (solid lines) and when employing a SISODQ (dotted lines)

5.3.2 Distributed IP Anchoring and Mobility Management

This CT4.1, iJOIN investigates an SDN-based "DistributedDistributed IP Anchoring and Mobility Management". As reported in IR4.2 [12], the AMM (Anchoring and Mobility Management) is the module defined in the functional architecture in charge of managing the mobility of the UEs attached to the network. When a UE moves from a point of attachment to another, AMM runs different algorithms to select the optimal anchor and configure the new path in the network. The AMM module follows the SDN-paradigm and runs on the iNC. The key point here is that the decision occurs in the iNC based on information gathered from the other nodes in the network, such as iTNs and iSCs.

At the time of writing this document, an initial implementation of the AMM module is available. The AMM runs on the iNC of the SDN Testbed as Ryu application. Ryu is the iOpenFlow[12] controller running on the iNC and hence AMM has been implemented directly upon Ryu APIs. Since this is the first draft not all the features are implemented yet, but it already provides the following functionality:

- UE attachment detection: iSCs detect the attachment and inform directly the AMM module. At the moment no communication occurs with the MME.
- Upon the UE attachment, the AMM selects the anchor statically. This means that the anchor selection algorithm is not implemented yet.
- Once an anchor is selected, the AMM configures properly anchor rules. The configuration is performed for new assigned anchor and also for old anchors.

In order to evaluate the AMM module, a basic Traffic Engineering Enforcement Module (TEEM) version has been implemented. The TEEM module provides the functionalities required by AMM. The AMM module requests the TEEM module to compute the best path, and setup the new best path for UE or traffic flowFigure 5-3 shows the software architecture implemented on the SDN-Testbed running on the iNC. The communication between the modules occurs as internal Ryu event following an event-driven communication paradigm. Each module exports the required events making available the subscription to other modules that are interested in such events.

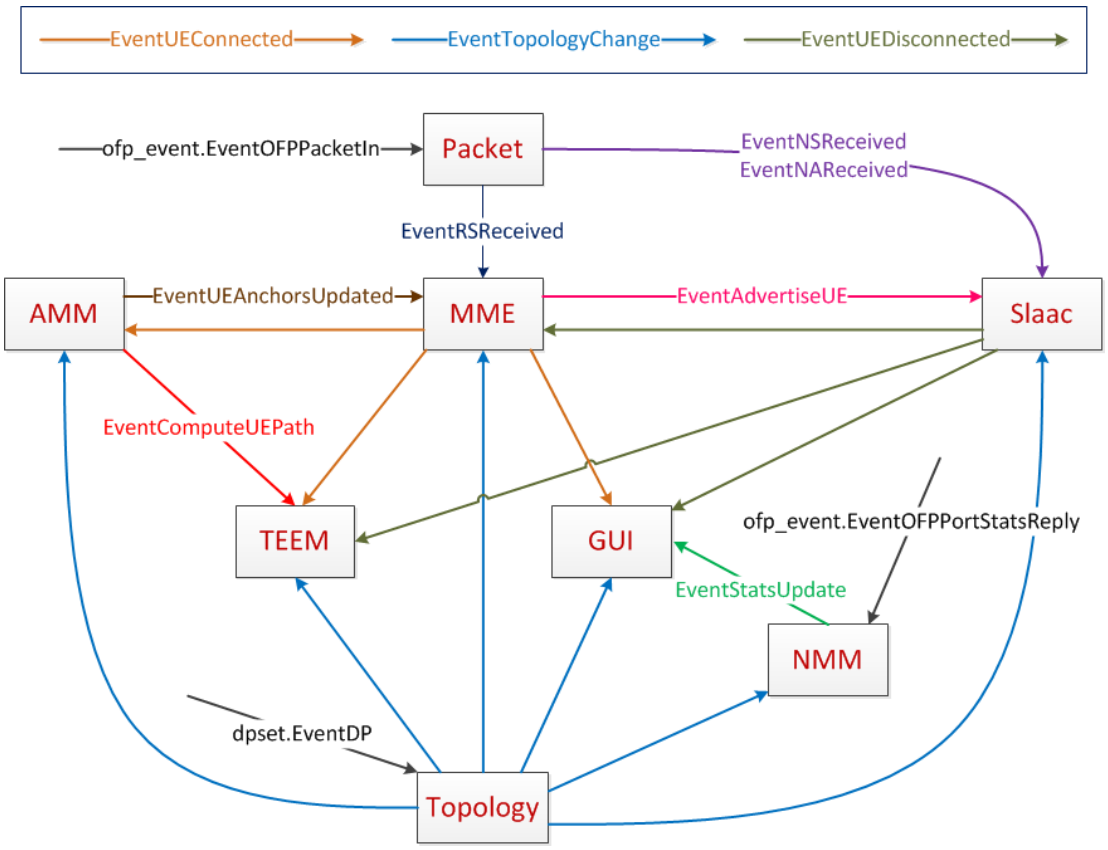


Figure 5-3: Partial functional architecture implemented on SDN-Testbed

We have obtained some preliminary results for the inter-anchor mobility scenario using the following measurement methodology:

- One node external to the SDN Testbed starts pinging the UE by sending ICMPv6 echo request packets every 2 ms.
- During this pinging procedure we detach the UE from the current iSC and we attach to a new iSC. The attachment triggers the AMM procedure.
- By measuring the ICMPv6 sequence number gap we can roughly estimate the overall handover time with a granularity of 2 ms.

In Figure 5-4 are reported the Cumulative Distribution Functions (CDFs) for the total handover time considering separately layer 2, layer 3 and ping disconnectivity, in case of 3 anchors assigned to a single UE.

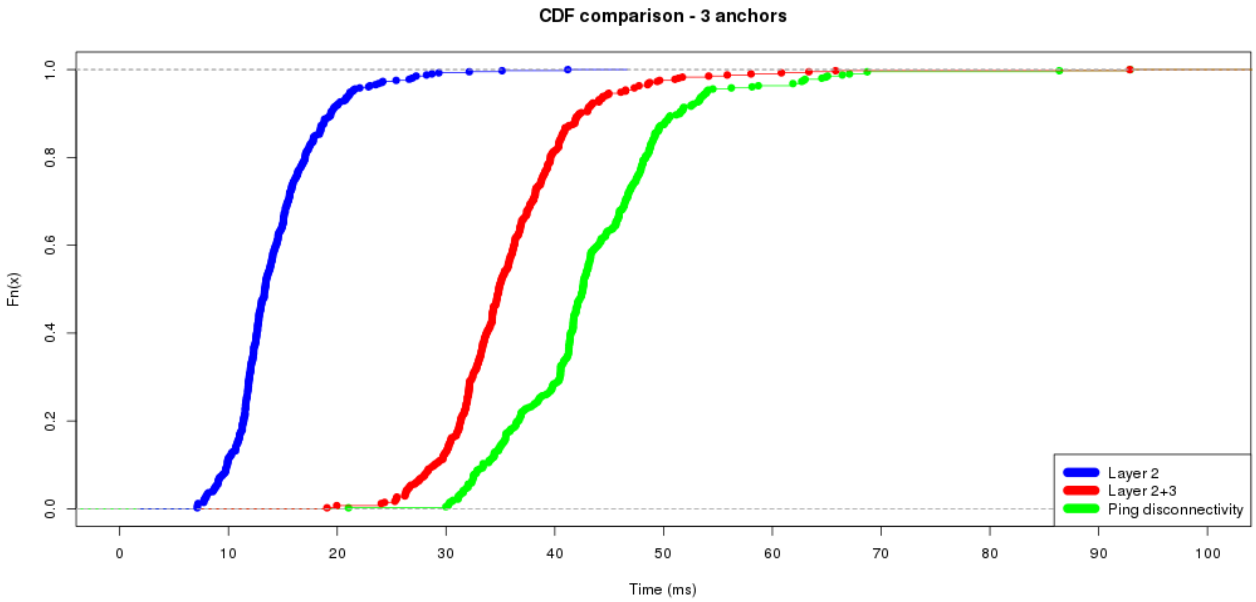


Figure 5-4: Handover time CDF

Using this methodology, the total handover time in terms of 95% percentile is:

- 21 ms, for layer 2 handover;
- 46 ms, for layer 3 handover;
- 54 ms, for ping disconnectivity.

The selection of the anchor is actually performed statically, therefore we believe that the total handover time will be slightly higher. Currently the main contribution to the total handover time is given by the configuration of OpenFlow rules on the anchors as depicted in Figure 5-5. In the SDN-Testbed, TEEM takes 1 ms for sending one OpenFlow configuration packet to the anchors. This time is also highly dependent on the distance between the controller and the anchors, further measurement will be made in order to evaluate this impact. The second main contribution is given by the creation and the delivery of Router Advertisement packets by AMM to iSCs. In our SDN Testbed this procedure takes 1 ms.

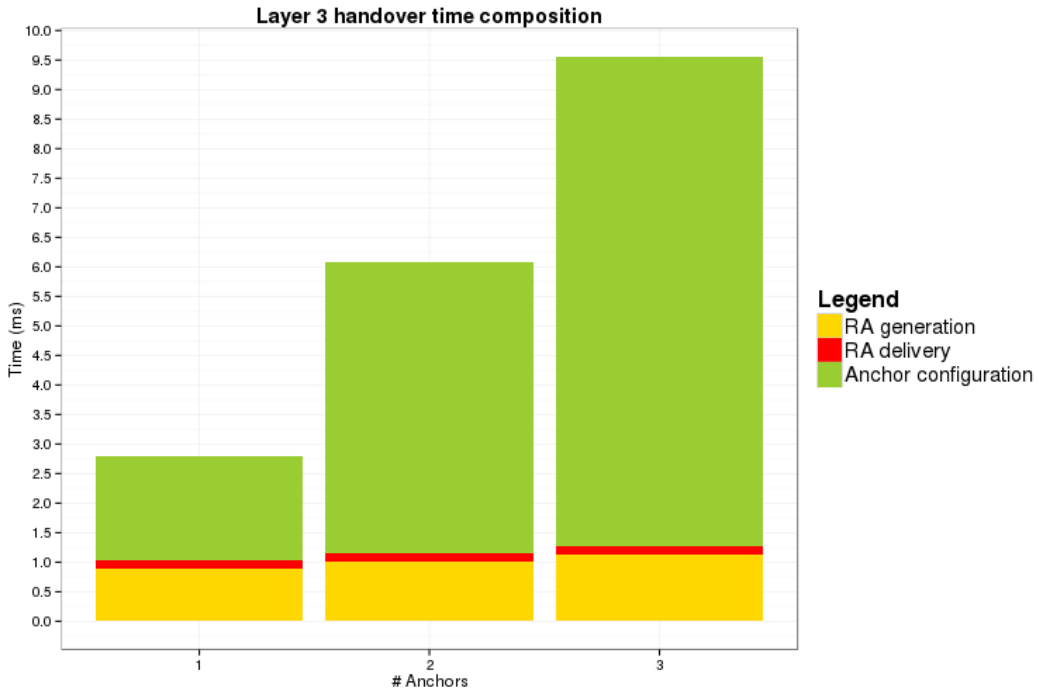


Figure 5-5: Total handover processing time

5.3.3 Network Wide Energy Optimisation

This CT4.2, iJOIN investigates an SDN-based "Network Wide Energy Optimisation". As reported in IR4.2 [12], the Network Energy Optimizer (NEO) is the module defined in the functional architecture in charge of managing the energy savings that the cellular network can achieve. Thus, NEO runs an algorithm that tries to decrease the energy consumption of the cellular network.

In general, dealing with energy consumption issues becomes more challenging. Significantly more opportunities arise for switching off iSCs in smaller time scales due to (a) coverage overlaps stemming from heterogeneous deployment of cells, (b) larger spatial-temporal load variations due to smaller number of users associated to each iSC and (c) power-proportional and load-dependent iSC. Thus, NEO not only guarantees the user Quality of Experience (QoE) while switching-off an iSC, but also tries to consider the achievable energy savings even for short time-scales. The key point here is that the decision occurs in the iNC based on information gathered from the other nodes in the network, such as iTNs and iSCs.

At the time of writing this document, an initial implementation of the NEO module is available. Thus, we have already considered the iSCs (access network), and we present some initial results about the energy savings. We are going to include the iTN (backhaul network) and switching-off schemes for these as well in future work.

While NEO tries to switch-off a cellular node, it must guarantee some desired levels for the user QoE[1][44] . Specifically, it should consider:

- **Network coverage**, i.e. the probability that a random user experiences poor signal quality when he/she needs to use the network (e.g. making a call, or sending a web request), defined as failure probability. While switching-off an iSC, then some users are going to be attached to further iSCs, so the average failure probability of a random user increases. We denote as $p_{failure}$ the maximum tolerant failure probability, defined from the operator.
- **Admission control and "blocking" probabilities** *(for flows that require a "dedicated" amount of bandwidth)*; these probabilities are not only related to user admission but also admission of flows that require a certain amount of dedicated bandwidth. While switching-off an iSC, then some users are going to be attached to further iSCs, so some iSCs will have to deal with more flows that require a certain amount of bandwidth, thus the blocking probability of such a flow due to lack of resources increases. We denote as p_{block} the maximum tolerant probability that is defined from the operator.
- **Service delay** *(for "best-effort" flows);* and the probability of delay exceeding some desired upper bound. While switching-off an iSC, then some users are going to be attached to further iSCs, so some iSCs will have to deal with more best effort flows, thus the ongoing delay of these flows increases. We denote as D_{max} the maximum delay threshold that is defined from the operator.

Thus, NEO should decide to switch-off a iSC, depending on the parameters defined above. It should check one, two or the three of them simultaneously, and be as strict as needed. Obviously, the larger the QoE thresholds ($p_{failure}$, p_{block}, D_{max}) defined from the operator, the less strict we are with the switching-off criteria,so the more iSC we can switch-off and the more energy we can save.

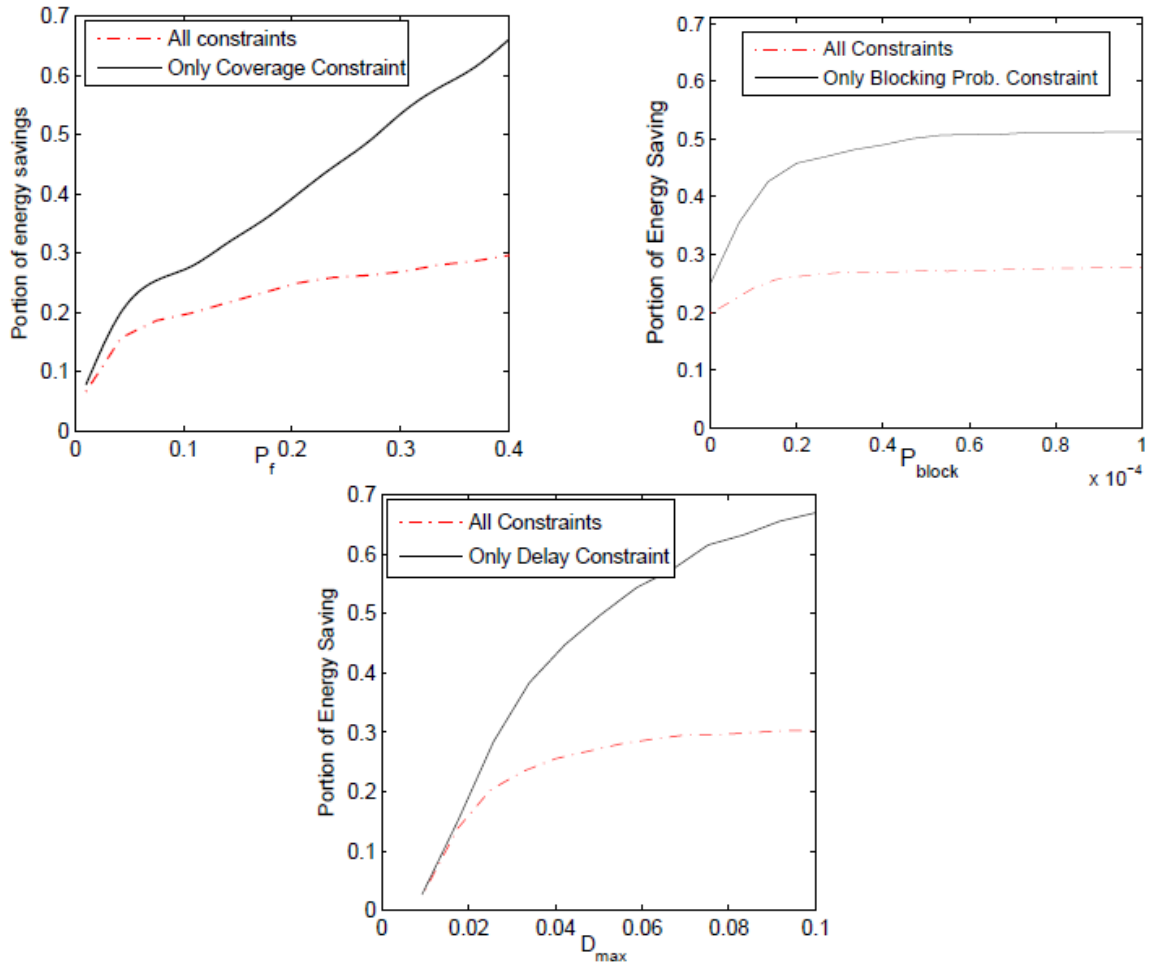


Figure 5-6 Achievable Energy Savings Vs. Thresholds

Figure 5-6 illustrates the achievable energy savings for different values of the "guaranteed" thresholds for the user QoE in a scenario of 120 iSCs and 2 macrocells. For example, in the up-right picture, the top curve corresponds to the portion of energy saved when we consider only the first constraint active, if the switching-off duration is supposed to last 10 min. On the x-axis we increase the constraint threshold and plot the respective energy savings. As can be seen there, increasing the threshold (i.e. making the constraint less strict) increases savings, as it allows for more iSCs to be switched off. For example, we can save up to 68% of the total energy consumption of our cellular network, for $p_{failure}$ = 0.4. The bottom curve also shows the energy savings, but now with the other two constraints active as well: the blocking threshold is fixed at 10−3 and the delay threshold at D_{max}=50 ms. As can be seen there, savings increase again, but less sharply, as the other two constraints can become the "bottleneck" for a switch-off decision, especially as $p_{failure}$ increases. For example, now, with $p_{failure}$ = 0.4 and the other two thresholds fixed, the portion of energy saving can be up to 30%. Similar behaviour is noticed in the other two pictures of the figure, for the other two constraints.

Figure 5-7 depicts the portion of energy saved for different values of the switching-off period (X). As can be seen there, energy savings are maximum when X is relatively small, but start decreasing and eventually flatten out, as X increases. The reason is that, for small X, one needs to only consider the impact of active users when evaluating the constraint and the impact of hand overs to neighbouring iSCs. However, as X increases, there is a higher chance connected and potential disconnected users will add to the total transferred load and thus a bigger impact on existing and remote users, which might prevent us from switching off an iSC. Finally, the plot for each respective constraint is not always linear, as some additional phenomena, such as convergence to stationarity for the stochastic systems we use in constraints 2 and 3, also affect systems' behaviour.

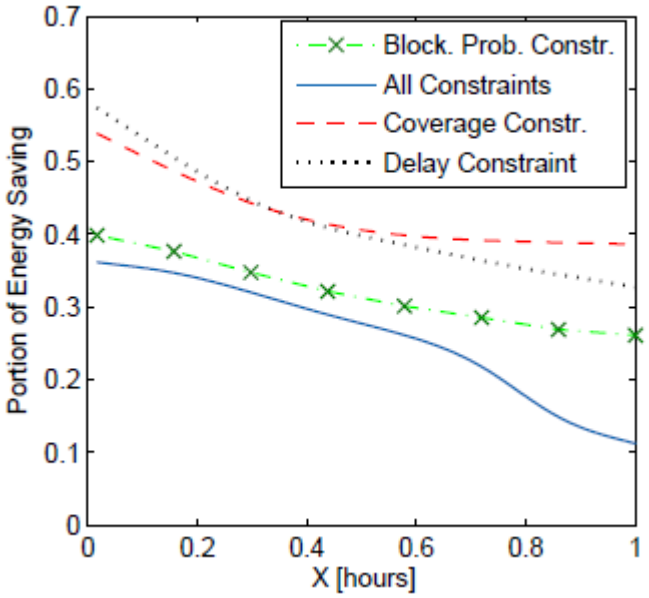


Figure 5-7 Achievable Energy Savings Vs. Switching-off period

6 System Performance Evaluation

6.1 Relevant metrics

6.1.1 Area Throughput

6.1.1.1 Objective

Past few years have seen a tremendous growth in internet data carried over by mobile cellular networks. Furthermore, the growth of mobile data traffic is expected to continue in the years to come, e.g. the mobile traffic demand in year 2020 will be at least 1000 times more than the capacity of current cellular networks [13]. Since today's mobile cellular networks cannot cope with this growth in traffic, both academia and industrial research have initiated research studies on how to evolve current 4G systems (e.g. LTE-Advanced) or design new ones to cope with the expected rise in mobile traffic. A key element of this effort is to increase the spectral efficiency of current and future cellular networks.

In this context, iJOIN targets to increase the system throughput within the same spectrum by a factor of 50-100 as a result of:

- High density of small cells, re-use of spectrum, and PHY / RRM improvements enabled by RANaaS to adequately address interference (≥10x)
- Shorter distances and increased LOS probability (5-10x)

6.1.1.2 Definition

Throughput is expressed in terms of *bits/sec/area* also referred to as area throughput. Area throughput measures the utilization of the radio spectrum over a given geographic area and also represents the capacity which a mobile operator offers to its subscribers.

Observing the network over some time period *T*, one can measure the traffic flowing through the network and also the network power usage. Denoting by $r_i(t)$ the rate by which bits are correctly delivered at (from) the UE *i*, the total information (number of bits) delivered, within the time period *T*, in a network comprising *N* UEs is calculated as:

$$I = \sum_{i=1}^{N} \int_{0}^{T} r_i(t)dt \qquad \text{[bit]} \tag{6.1}$$

The average rate *R* in the network is then simply *I/T*. It may often be helpful to normalise the rate *R* by either the number of cells or the network area. To make the normalised measures independent of the deployment, we choose here to work with rate per area unit expressed in square kilometres. Area throughput is defined within iJOIN as the average rate per area unit R_A. It is then calculated as:

$$R_A = \frac{R}{A} = \frac{I}{A \cdot T} = \frac{1}{A \cdot T} \cdot \sum_{i=1}^{N} \int_{0}^{T} r_i(t)dt \qquad \text{[bps/km}^2\text{]} \tag{6.2}$$

A shortcoming of this definition is that it provides only an average value and does not reflect the distribution of throughput in a given area. Capacity distribution in a given area greatly impacts Quality of Service (QoS) to a mobile user, for example, in terms of session dropping / blocking and data rate requirements.

In this direction, simulations are often used to produce not only average values but also the CDF of the cell or user throughput, in order to give more complete information about the system behaviour. In particular, an important metric that must be taken into account is the cell edge user throughput: a good system design should take into account also this statistic, so that also minimum radio performance is guaranteed in the covered area.

6.1.2 Energy Efficiency

Energy efficiency (EE) evaluation of the iJOIN system (and in particular of each CT) is strictly related to the proposed logical architecture. In fact, the power consumption of the veNB should take into account the iSCs,

RANaaS data centre and also the backhaul network including iTNs. At a first glance, it is not yet clear whether an iJOIN system may imply a higher energy consumption than today's networks, and whether the consumed per delivered bit increases or not. Moreover, the iJOIN architecture is potentially enabling advanced and convenient RAN sharing scenarios that may significantly improve energy performance and long term sustainability also in the view of future 5G systems. Energy efficiency evaluations are traditionally performed [4], [15] by considering (at network level) mainly two kind of EE metrics: energy per information bit (expressed in [J/bit] or equivalently [W/bps]) and power per area unit (expressed in [W/m^2]). Thus, given a specific evaluation scenario (CS), it is possible to compare a certain EE metric of a classical flat architecture compared to the iJOIN architecture, considering both RAN and backhaul parameters.

The main metric used for energy efficiency used in iJOIN is consumed energy per information bit (see deliverable D5.1 [4] for further details). In any case, all EE metrics, in order to be evaluated (at network level), need the computation of energy consumption of the assessed network (given by the contribution of of all network elements). While the traditional architecture considers several sophisticated small cells, iJOIN architecture is composed by several iSCs and a RANaaS platform where pooling of complex (e.g. baseband) processing can be performed. In order to investigate the convenience (from an energy performance perspective) of this proposed architecture compared to the traditional one we need to introduce a generalized holistic power model. In fact, power consumption at system level should be evaluated by considering the sum of all contributions in the network. This will help us to at least perform a quantitative analysis on the RANaaS system power consumption and discuss the potential benefits in terms of energy efficiency, especially when varying the load in the RANaaS.

The system energy consumption is directly related to the power usage of all network elements over a time period. Considering the system architecture as introduced in Section 3 a holistic power model for a RANaaS system comprising N_{iSC} iSCs can be given by:

$$P_{\text{Total}} = P_{\text{RANaaS}} + P_{\text{Bh}} + \sum_{n=1}^{N_{\text{iSC}}} P_{\text{iSC-}n} \tag{6.3}$$

where P_{RANaaS}, P_{Bh} and $P_{\text{iSC-n}}$ stand for the power consumed at the RANaaS platform, the power needs for backhaul and the power usage at any iSC *n*, respectively. It should be noted that the amount of power consumed by iSCs and RANaaS depends also on the particular functional split considered by the CT in iJOIN. In some cases, CTs dealing with flexible functional split should consider in their evaluations (during the time period considered, e.g. 24 hours) a variable power consumption, according to the functional split switching applied by the CT in that period). As a side note, the power consumption of the backhaul network (based on wireless links) is mainly due to the presence of iTNs (for the time being TNs are not considered). Further details of the power model (also described in [27]) are given in Annex A.

After the calculation of network energy consumption, Energy Efficiency is thereby measured as an ECI, (Energy Consumption Index, as defined in D5.1 [4]), and finally baseline and frontline perfomances are compared, by calculating the relative gain in terms of ECI values.

6.1.3 Utilisation Efficiency

Utilization efficiency is defined as a metric which expresses how well the utilized resources are used for a given performance metric. Therefore, high utilization efficiency means the following:

- The system (such as a network) is highly utilized, and therefore not over-provisioned.
- The system is capable to exploit utilized resources efficiently to provide the desired output, such as cell throughput or other metrics.

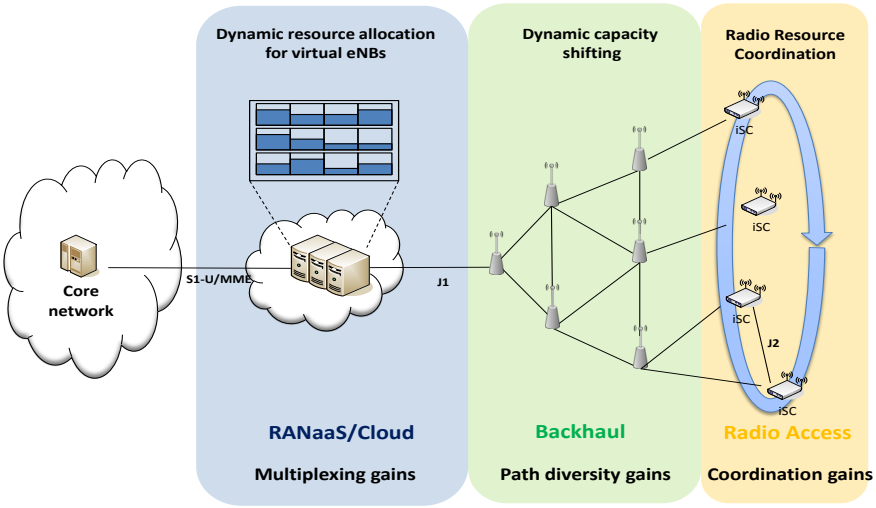


Figure 6-1: Utilization gains in different network domains

Figure 6-1 shows an example of how different resource allocation techniques in different iJOIN network domains can lead to different types of gains (e.g. multiplexing, diversity and coordination gains). It also illustrates a fundamental problem of defining a network-wide metric for utilization efficiency: different network domains (i.e. RANaaS, backhaul, radio access) utilize different types of resources (e.g. CPU cycles, link bandwidth, radio spectrum), such that a simple summation of domain-specific metrics is in general not possible. We define the total utilization efficiency of a system as following:

$$\eta_U = \frac{\sum_{d \in D} \alpha_d u_d}{|D|} \tag{6.4}$$

where α_d is a scaling factor s.t. $\sum \alpha_d = 1$, and u_d is the *domain utilization* for the considered domain, with D as the set of network domains (e.g. RANaaS, backhaul, RAN).

The definition of the domain utilization u_d depends on the resource of interest. As described in [26], different network domains have in many cases different resources. However on a more abstract level, resource normalization can be applied across network domains. We identified the following resource classes which will be investigated in more detail:

- Bandwidth/capacity resources. The domain utilization is defined as

$$u_d^B(X) = \frac{B_{mean,d}(X)}{B_{cap,d}(X)}, \tag{6.5}$$

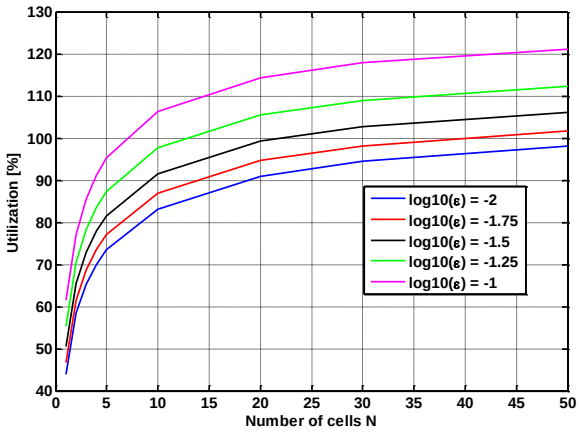
  where $B_{mean,d}(X)$ is the average measured data rate and $B_{cap,d}(X)$ is the corresponding outage or theoretical maximum capacity of the system. The parameter X depends on the investigated network scenario and can be the number of cells, user arrival rate, etc.

- Computational resources. Here, the domain utilization is defined by

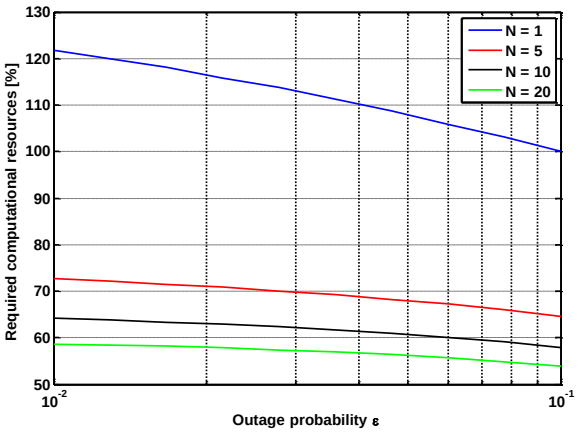
$$u_d^C(X) = \frac{C_{mean,d}(X)}{C_{outage,d}(X)}, \tag{6.6}$$

  where $u_d^C(X)$ is the ratio of expected computational demand and provided computational resources, depending on the number of cells in the scenario, X. The latter is the outage complexity which is defined as the amount of computational resources to make sure that a per-cell computational outage ε is not exceeded. Both are defined through an analytical framework which has been described partly in Section 4.1.3. This framework resembles the characteristics of computational load of a 3GPP LTE uplink decoder.

Preliminary evaluation of computation utilization efficiency:



(a) Utilization as a function of number of cells



(b) Required outage complexity as a function of the per-cell outage probability

Figure 6-2: Computational utilization efficiency

Based on this framework, the expected utilization of a centralized processor for different number of cells and depending on the outage is shown in Figure 6-2. For these results typical LTE parameters including actual SNR link-adaptation thresholds have been used. Furthermore, a Rayleigh fading process is assumed with SNR of 10dB. In the next report, this investigation is extended to more complete fading processes including path-loss and power control. However, the results will differ quantitatively but not qualitatively.

From Figure 6-2(a) we can see that for a large number of centralized base stations, an expected utilization of more than 100% is achieved. This implies that less computational resources than the expected overall computational demand are provided. This is due to the fact that the system is optimized such that a per-cell outage probability is not exceeded. We can observe that this effect depends strongly on the chosen outage probability, e.g. for a computational outage of 10% already 7 centralized base stations would exceed the provided resources while for a computational outage of 1% more than 50 base stations need to be centralized. This utilization performance curve will be helpful to dimension the centralized resources accordingly and to design the resource scheduler. Based on the actual communication resource demand (throughput) also the computational resource demand (processing) can be scheduled, and vice versa.

In the next report, these results will be further detailed to include more practical constraints and characteristics, e.g. multiplexing gain and more complex channel models.

6.1.4 Cost Efficiency

The cost efficiency of iJOIN will be investigated by combining the large-scale analytical results based on stochastic geometry obtained within CT4.6 with the analysis of computational complexity and diversity as illustrated in Section 4. The users, base stations, backhaul nodes, and data centres are modelled using independent homogeneous Poisson point processes as illustrated in Figure 6-3.

A particular equipment cost for each device is assumed. Capacity and infrastructure cost are assumed to have given "base" cost for connecting two different network components and this base cost is assumed to increase with the distance between the network components, i.e., the base cost is multiplied by a function of the distance which is in power law form (e.g., for a distance 'r' the function 'f(r)' takes the form Ar^{β}, where A is the base cost and $\beta > 0$ is the rate at which the cost increases). Utilising this method of modelling, we can obtain an expression for the average cost of deploying a backhaul node. From which, we obtain the total cost of the network.

Furthermore, we use the results of the computational complexity analysis to derive the expected computational diversity gain, i.e. a linear function which defines the required computational resources depending on the number of centralized users and as multiple of the required computational resources for a single base-station. The required computational resources depend on the service quality, i.e. if the LTE system operates at its maximum achievable rate more computational resources are required while at slightly reduced achievable rate fewer resources are necessary. These dependencies are taken into account by scaling the required computational resources with the offered achievable rate per base station while increasing the base-station density accordingly. We further take into account that a maximum computational outage must

not be exceeded. The model will incorporate expected costs of data-centres as a function of the data-centre size.

Using this model,

- we will compare the cost performance against a traditional scenario in which there is no RAN functionality executed in the cloud;
- we will leverage on analysis of the computational resources required in the cloud vs. resources required in the eNB, size of the area served by the cloud, etc;
- we will determine the deployment cost for the iJOIN network and for a traditional one as a function of the cost of the individual parameters (such as the cost of a processing unit, the cost of a bandwidth unit in the backhaul, etc.).

Based on the above, we plan to evaluate cost efficiency based on ranges of the costs of the various components (e.g., we may conclude that iJOIN is cost efficient as long as the cost of a processing unit is not too high as compared to that of a bandwidth unit).

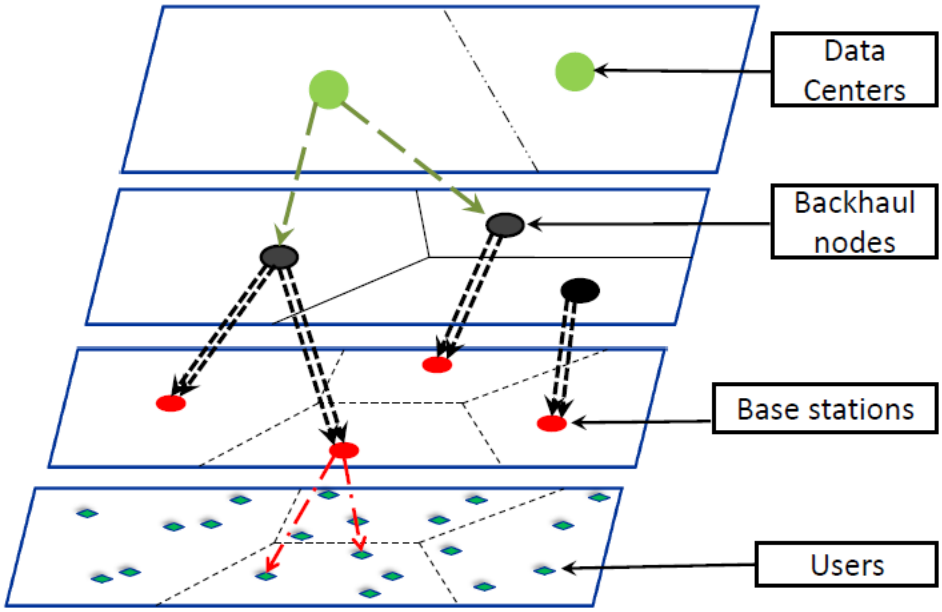


Figure 6-3: Network model for cost-efficiency analysis

6.2 Performance evaluation campaigns and parameterization

iJOIN's overall vision is based on the previously introduced four metrics, which are associated to four numerical targets that should be evaluated at project level by the end of the project (M30):

1. Area throughput: R = 100x
2. Energy-efficiency: *J/bit* < 5%
3. Utilisation efficiency: η_U > 75%
4. Cost-efficiency: €/bit < 10%

Each CT (from WP2, WP3, WP4) will be evaluated by considering at least one of the above discussed iJOIN metrics. Moreover, it is likely that a single CT will not be able to achieve the specific project target by itself, e.g. it may happen that a single CT will provide gain of 80x in terms of area throughput, while target at project level is 100x.

As a consequence, in order to provide global evaluations, iJOIN will compare its CTs and combine their respective results, if possible. It should be noted that it is not an objective of the iJOIN project to perform a joint implementation of different CTs, but the intention is to compare and combine individual gains. The

main requirement that has been identified to enable this kind of final project-wide evaluation of CTs is the alignment of evaluation scenarios (i.e. the four iJOIN common scenarios described in D5.1) and related PHY architectures (with high level parameterization).

WP5 established a common workflow in order to provide global evaluations of performances:

1. Definition of four CSs and related PHY architectures (with high level parameters)
   For each scenario and work package, a set of relevant parameters is defined. In addition, value ranges are defined. Each of these parameters needs to be supported directly or indirectly by each CT in order to make sure that evaluation results are aligned.
2. Alignment of common assumptions across all CTs, in order to have comparable results
   Based on the previous input, a project-wide alignment of evaluation parameters is performed. The result of this alignment needs then to be supported by all CTs.
3. Understand (at WP level) the conceptual possibility to have multiple CTs in a system
   Within each WP, a preliminary analysis is performed which determines if and how individual CTs can be combined. As a possible result, CTs may be complementary and can be used at the same time, or they are contradicting and cannot be used at the same time.
4. Evaluate combination of numerical results wrt. the targets
   Using the previous input how CTs can be combined as well as the aligned evaluation results, a project-wide evaluation is performed.

The first step is addressed in this report and will be finalized in D5.2 deliverable. The second step is preliminarily addressed in [10], [11], and [12]. Steps 3 and 4 will be addressed in the final deliverable D5.3.

6.2.1 Radio Access Network Settings

This section resumes common reference link level and system level parameters defined in iJOIN WP2 and WP3, respectively. This work is an indispensable prerequisite to compare the different partners' solutions. Parameters and settings described here are mainly based on the LTE system (3GPP TR 36.872 [3], 36.932 [2], and 36.814 [1]). The described reference deployment assumptions apply to small cells, which can be located in both outdoor and indoor scenarios. Note that here a small cell "cluster" only refers to the characteristics of the small cell deployment and it is not related to the iJOIN logical architecture. In particular, according to the 3GPP terminology, it indicates a number of neighbouring iSCs.

Link level simulation settings are presented in Table 6-1. Table 6-2 describes parameters for system level simulations. Note that system level parameters directly apply on the top of link level ones.

Table 6-1: iJOIN link level simulation settings

Parameters	Outdoor Model	Indoor Model
System bandwidth per carrier	10 MHz	10MHz
Carrier frequency	2/3.5 GHz	2.6/3.5GHz
Carrier number	1 carrier	1 carrier
Total BS TX power	30 dBm	24dBm
Total UE TX power	23 dBm	N/A
Distance-dependent path loss	ITU Umi (3GPP TR36.814 [1])	ITU InH (3GPP TR36.814 [1])
Antenna configuration	1x1, 2x2	1x1, 1x2, 2x2
Number of small cells per cluster	4/10	5/10
Number of UEs	Varying	Depending on the CS
UE speed	Static UEs (0km/h) or pedestrian (3km/h)	Static UEs (0km/h) or pedestrian (3km/h)
Channel estimation	Perfect	Perfect
Synchronization	Perfect	Perfect
UL Modulation	QPSK,16QAM, 64QAM	QPSK,16QAM, 64QAM
Coding for data channel	LTE Turbo Code, LDPC	LTE Turbo Code

Table 6-2: iJOIN system level simulation settings

Parameters	Outdoor Model	Indoor Model
Layout	Outdoor small cell cluster with/without the macro eNB	Indoor small cells according ITU indoor Hotspot 3GPP TR36.814
Total BS TX power	30 dBm	24dBm
Number of small cells per cluster	Depending on the CS	Depending on the CS
Number of UEs	Depending on the CS	Depending on the CS
UE dropping	Random or Hotspot-like Outdoor	Random (indoor)
Radius for small cell dropping in a cluster	Depending on the CS	2/4 small cells per floor, 1/2 floors
Radius for UE dropping in a cluster	Depending on the CS	n.a.
Minimum distance between nodes	iSC-iSC: 20m iSC-UE: 5m	Small cell-UE: 3m
Traffic model	Full Buffer	Full Buffer
Cell selection criteria	RSRP; CRE not applied	RSRP; CRE not applied
Backhaul model	Depends on CS	Depends on CS
Other Simulation variables	Backhaul Delay; QoS requirements; Power constraint	n.a.
Target metric	Throughput; Energy Efficiency; Utilization Efficiency; Delay vs. offered load	Throughput; Spectral Efficiency

6.2.2 Backhaul Network Settings

This section resumes common reference parameters defined in iJOIN WP4 for the assessment of CTs related to the performance optimization in the transport network.

A generic backhaul scenario considered by iJOIN is shown in Figure 6-4. We consider both wired and wireless links for the interconnection of the small cells with the metro network. Different deployment options are possible, depending on the scenario. For example, iSCs might be connected via wireless links to one iJOIN transport node, and from there with Ethernet to the aggregation network, or even a multi-hop wireless network might be used to provide connectivity to the aggregation or metro network. It is important to consider the different deployment scenarios that are possible, map them to the common scenarios that are addressed by iJOIN, and then characterize their latency and bandwidth. This information would be then used to assess the performance of the different CTs, and also to evaluate under which situations each of the different iJOIN innovations provide a significant improvement.

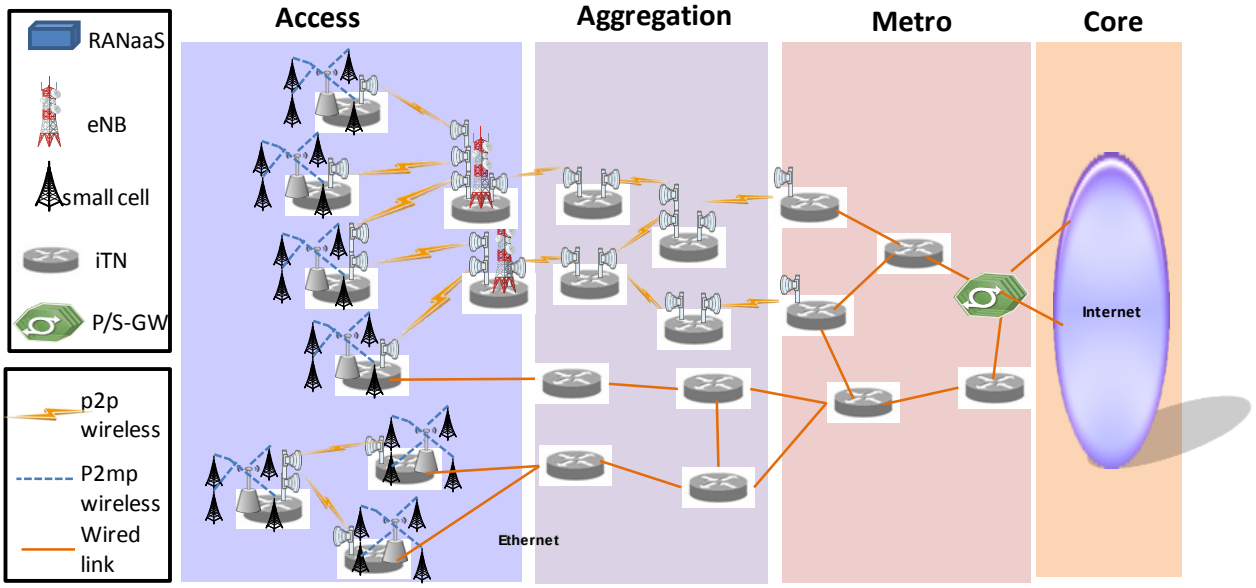


Figure 6-4: iJOIN generic backhaul scenario

Latency requirements need to be fulfilled by the backhaul for reliable operations in RAN as well as to enable different functional split options in the RANaaS. 3GPP defines many timers from the MAC to the RRC layer. These values will ultimately define the maximum latency requirement needed per layer enabling a transparent functional split, i.e. without any specification changes (see Section 4.2).

In LTE, the PHY layer works with 1 ms subframe granularity. At the MAC layer, the HARQ timing is the most critical one. Once a subframe has been sent at subframe *n* for a given HARQ process, an acknowledgement (positive or negative) is expected at subframe *n*+4. Due to the synchronous nature of HARQ in the uplink, any functional split at the base station MAC layer requires the round-time trip time plus the processing to be done in 3 ms, which is a strong constraint (see Section 4.5.1). Having this in mind, it seems that wireless backhauling could only be used in some limited scenarios, as for example involving multiple hops might not be feasible, being wired Ethernet the preferable option. This will be analysed in detail in D5.3.

On the top of these constraints, iJOIN CTs are characterized by further latency requirements to be met in order to successfully operate (see Section 5.2.1).

WP2 CTs are mainly characterized by very tight constraints (from below 1ms to few ms), i.e. to exchange up-to-date CSI for coordinated inter-cell interference management or user messages for centralized or distributed signal reception.

In WP3, iJOIN CTs are characterized by a larger range of latency requirements: CTs focusing on very fast radio resource management/scheduling have latency constraints below 1 ms while coarse grained RRM mechanisms operate on a time scale larger than the LTE time frame (10 ms). Finally, mechanisms that focus on the RRC and BH optimization have light latency constraints (around 1s).

WP4 CTs do not impose critical constraints on the backhaul latency and bandwidth. They can be rather considered as part of the enablers of the functional split concept, aiming at ensuring a certain connectivity characteristics in the backhaul between the iJOIN small cells and the RANaaS.

6.3 Scenario specific parameterization

6.3.1 Common Scenario 1: Stadium

In this section we present the model to assess the iJOIN CTs in CS 1 (the stadium). Performance evaluation in the whole stadium is not feasible; hence we focus on a limited area of the stadium represented by a small cell hotspot. Two layouts are considered, with medium and high user and small cell density, respectively.

The main characteristics of this stadium hotspot are:

- Regular and dense small cell deployment

- Regular and dense user deployment
- Static users
- High capacity, low latency backhaul

Further details are presented in Table 6-3.

Table 6-3: Stadium settings

Parameters	Stadium
Number of small cells per cluster	15 but focus on central 3
Number of UEs	28-high load 24-medium load
Radius for small cell dropping in a cluster	40 m x 80 m Uniform dropping
Radius for UE dropping in a cluster	20 m x 60 m Uniform Dropping
Minimum distance 3GPP TR 36.872 [1]	iSC-iSC 20 m UE-iSC 5 m Macro eNB-iSC cluster center 105 m
Backhaul Capacity	100 Mbps 200 Mbps >200 Mbps
Backhaul Latency	<1, 5, and 10 ms

A typical layout for the stadium scenario is shown in Figure 6-5.

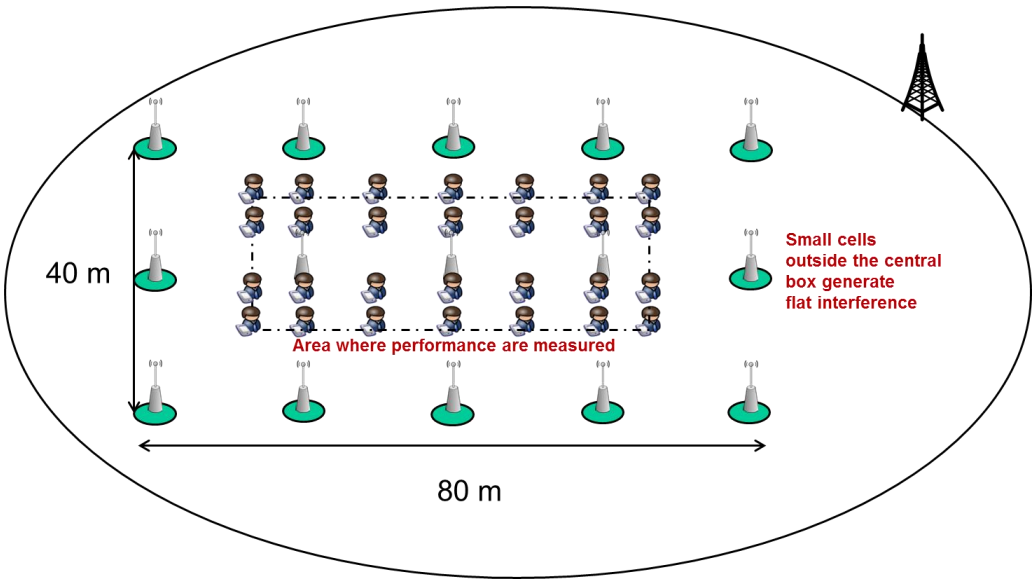


Figure 6-5: Stadium Layout in high load scenarios

6.3.2 Common Scenario 2: Square

In this section we present the model to assess the iJOIN CTs in the CS 2 (the square). The square layout is based on the small cell deployment described by 3GPP in TR 36.872 (A1.1 and A1.2) [1].

The main characteristics of the square hotspot are:

- Random small cell deployment
- Random user deployment
- Static/Nomadic user
- Heterogeneous backhaul

Further details are presented in Table 6-4.

Table 6-4: Square settings

Parameters	Square
Number of small cells per cluster	4-10 (sparse to dense deployment)
Number of UEs	15-30 (lightly to highly loaded scenarios)
Radius for small cell dropping in a cluster	50m (3GPP TR 36.872 [1]) Random Dropping
Radius for UE dropping in a cluster	70m (3GPP TR 36.872 [1]) Random Dropping
Minimum distance 3GPP TR 36.872 [1]	iSC-iSC 20 m UE-iSC 5 m Macro eNB-iSC cluster center 105 m
Backhaul Capacity	~50 Mbps ~100 Mbps >100Mps
Backhaul Latency	<1, 5, and 10 ms

The layout for the square scenario is shown in Figure 6-6.

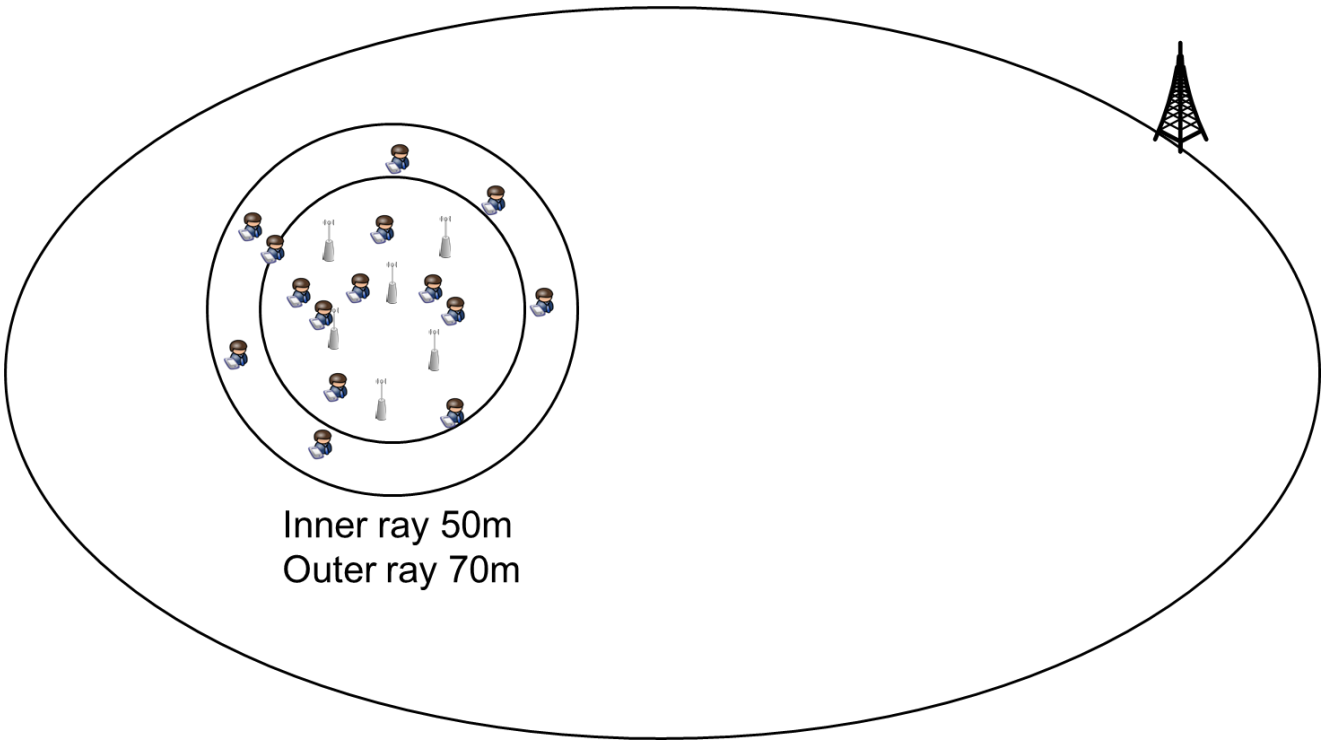


Figure 6-6: Small cell deployment in the square.

6.3.3 Common Scenario 3: Wide Area Coverage

In this section we present the model to assess the iJOIN CTs in the iCS 3 (Wide Area Coverage). The Wide Area Coverage layout is based on regular small cell deployment in a hexagonal grid, which covers 1 Km^2.

The main characteristics of the Wide Area Coverage are:

- Regular small cell deployment
- Random user deployment
- Slow/High mobility
- Heterogeneous backhaul

Further details are presented in Table 6-5.

Table 6-5: Wide Area Coverage settings

Parameters	Wide Area Coverage
Number of small cells	19
Number of UEs	15-30 (lightly to highly loaded scenarios)
Small cell dropping	Regular on Hexagonal Grid
ISD	96/√19 m
UE dropping in a cluster	Random dropping in the seven central small cells
Minimum distance	UE-iSC 5 m
Backhaul Capacity	~50 Mbps ~100 Mbps >100Mps
Backhaul Latency	<1, 5, and 10 ms

The layout for the Wide Area Coverage square scenario is shown in Figure 6-7.

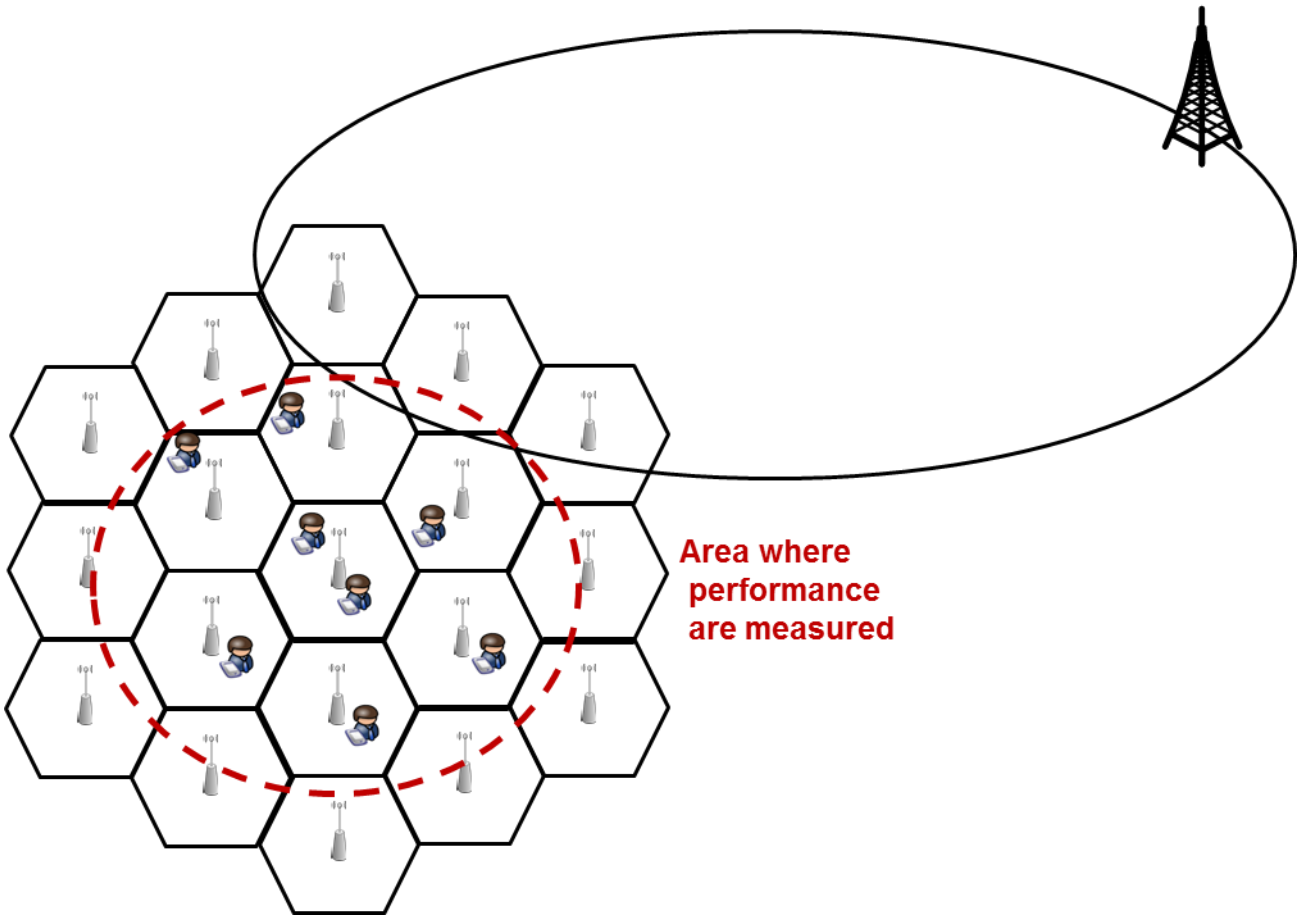


Figure 6-7: Small cell deployment in the Wide Area Coverage.

6.3.4 Common Scenario 4: Shopping Mall / Airport

In this section we present the model to assess the iJOIN CTs in the CS 4 (Shopping Mall/Airport). This layout is based on the ITU indoor small cell deployment described by 3GPP in TR 36.872 (A1.5 and A1.6) [1]. Two layouts are considered, with sparse and dense small cell density, respectively.

The main characteristics of this hotspot are:

- Regular small cell deployment
- Random user deployment
- Nomadic user
- Wireline backhaul (optical fiber and ADSL)

Further details are presented in Table 6-6.

Table 6-6: Shopping Mall / Airport settings

Parameters	Shopping Mall / Airport
Number of rooms/floor	Rooms 16 Floor 1 or 2
Floor height	6 m
Room size	15 m X 15 m
Hall size	120 m X 20 m
Number of small cells per cluster	2 (sparse) 4 (dense) per floor 1 or 2 floor (only in dense deployment)
Small cell dropping	Regular
Number of UEs	10 per small cell (sparse) 5/10 per small cell (dense)
UE dropping	Random
ISD	30 m (dense) 60 m (sparse)
Minimum distance	UE-iSC 3 m
Backhaul Capacity	~100 Mbps >100Mps
Backhaul Latency	<1, 5, and 10 ms

The layouts for the iCS4 scenario are shown in Figure 6-8.

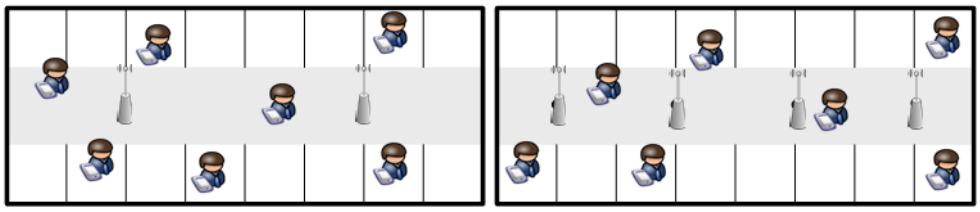


Figure 6-8: Small cell deployment in the Shopping Mall / Airport: Sparse (left) and dense (right) deployment.

7 Summary and Conclusions

This report provided a comprehensive overview of the iJOIN functional architecture, i.e. how novel candidate technologies interact, which objectives they address, and which impact they have on the overall system. This will be required until the end of the project to perform a system-wide evaluation. This report further provides a detailed analysis of the split of RAN functionality. In particular, implementation aspects have been discussed which will lead to a feasible study at the end of the project. In addition, virtualized infrastructure received particular attention as it will lead to new constraints and requirements if RAN functionality is executed on top of it. 3GPP LTE RAN constraints have been identified and discussed. In this report, solutions to the most challenging constraints are discussed and results are provided. Beside the functional split analysis, the joint RAN/BH operation has been further detailed. Finally, this report discussed the evaluation campaign based on harmonized parameters for each common scenario agreed across all partners of the iJOIN project.

Based on this report, the following preliminary conclusions can be drawn

- An implementation of RAN functionality on commodity hardware appears feasible. Only a very limited set of functional splits seem to be useful, i.e. digitized received signals (similar to CPRI) if the required bandwidth and backhaul technology is available, digitized and (soft-) demodulated/modulated signals in order to perform centralized decoding, or only centralized RRC while lower-layer functionality remains with the RAP.
- Due to practical 3GPP LTE RAN constraints, an implementation of a functional split over heterogeneous backhaul network is challenging. Most importantly, latency and throughput constraints of the underlying backhaul technology determine the achievable functional split. The probably most challenging task is to mitigate the latency constraints, e.g. incurred to HARQ and radio resource control for which iJOIN introduced novel technologies which are able to cope with these constraints.
- iJOIN will perform a harmonized evaluation campaign where results will be compared on a relative performance basis, i.e. using a common set of parameters, each CT is compared to the baseline system. Based on this relative performance, CTs are compared and it is shown in which scenarios they are most efficient and how their performance scales in system parameters such as RAP density and user density.
- Basis for the comparison of CTs will the four objectives energy-efficiency, cost-efficiency, utilization-efficiency, and area throughput. All four objectives are defined in this report.

Acknowledgements and Disclaimer

This work was partially funded by the European Commission within the 7th Framework Program in the context of the ICT project iJOIN (Grant Agreement No. 317941). The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the iJOIN project or the European Commission.

References

[1] 3GPP, “TR 25.814 v7.1.0; Physical Layer Aspects for Evolved UTRA”, Sep. 2006.

[2] 3GPP, “TR 36.932 V12.0.0; Scenarios and Requirements for Small Cell Enhancements for E-UTRA and E-UTRAN”, Dec. 2012

[3] 3GPP TR 36.872, “Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects (Release12)”, V12.1.0, Dec. 2012.

[4] iJOIN, Deliverable D5.1 “Revised definition of requirements and preliminary definition of the iJOIN architecture,” November 2013.

[5] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, “Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services”, *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, vol. 8, 2008, pp. 337-350.

[6] B. P. Rimal, E. Choi, Eunmi, and I. Lumb, “A taxonomy and survey of cloud computing systems”, *IEEE 5th International Joint Conference on INC, IMS and IDC*, 2009, pp. 44-51.

[7] A. K. Sidhu, S. Kinger, “Analysis of Load Balancing Techniques in Cloud Computing”, *International Journal of Computers & Technology*, vol. 4, no. 2, 2013, pp. 737-741.

[8] J. W. Smith, “Green Cloud: A literature review of Energy-Aware Computing”, University of St Andrews, http://jamie.host.cs.st-andrews.ac.uk/files/LiteratureReview.pdf, 2010.

[9] J. Di Giglio, D. Ricci, “High Performance, Open Standard Virtualization with NFV and SDN”, http://www.windriver.com/whitepapers/ovp/ovp_whitepaper.pdf

[10] iJOIN, Internal Report IR 2.2 “Preliminary definition of PHY layer approaches that are applicable to RANaaS and a holistic design of backhaul and access network,” May 2014.

[11] iJOIN, Internal Report IR 3.2 “Preliminary set of candidates for joint access/backhaul RRM and novel RRM algorithms for RANaaS scenarios,” May 2014.

[12] iJOIN, Internal Report IR 4.2 “Network-layer algorithms and network operation and management: preliminary set of candidate technologies,” May 2014.

[13] Nokia Siemens Networks, “2020: Beyond 4G, Radio Evolution for the Gigabit Experience”, Aug. 2011

[14] P. Grover, K. A. Woyach, and A. Sahai, “Towards a communication-theoretic understanding of system-level power,” IEEE Journal on Selected Areas in Communications, September 2011

[15] EARTH, Deliverable D2.4 “Most suitable efficiency metrics and utility functions,” December 2011.

[16] 3GPP TR 22.951, Service Aspects and Requirements for Network Sharing, Rel.11, Sep. 2012.

[17] 3GPP TR 22.852, Study on Radio Access Network (RAN) Sharing Enhancements, Rel.12, June 2013.

[18] 3GPP TS 23.251, Network Sharing; Architecture and Functional Description, Rel.12, Dec. 2013.

[19] 3GPP TS 36.413, S1 Application Protocol (S1AP), Rel.12, Mar. 2014.

[20] 3GPP TS 36.423, X2 Application Protocol (X2AP), Rel.12, Mar. 2014.

[21] 3GPP TS 36.331, Radio Resource Control (RRC), Rel. 12, Mar. 2014.

[22] Common Public Radio Interface, [online: http://www.cpri.info/]

[23] J. Bartelt, G. Fettweis, „A Soft-Input/Soft-Output Dequantizer for Cloud-Based Mobile Networks,” *15th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2014)*, Toronto, Canada, June 2014.

[24] 3GPP TS 23.203, Policy and charging control architecture, March 2014.

[25] IEEE 802.1q, Virtual LANs, October 2012.

[26] iJOIN, Deliverable D3.1 “Final report on MAC/RRM state-of-the-art, requirements, scenarios, and interfaces in the iJOIN architecture,” November 2013.

[27] D. Sabella, A. De Domenico, E. Katranaras, M. Imran, M. Di Girolamo, U. Salim, M. Lalam, K. Samdanis, A. Maeder: “Energy Efficiency benefits of RAN-as-a-Service concept for a cloud-based 5G mobile network infrastructure”, submitted to IEEE Networks magazine, 2014.

[28] G. Auer, V. Giannini, I. Godor, P. Skillermark,M. Olsson, M. Imran, D. Sabella, M. Gonzalez, C. Desset, O. Blume, A. Fehske, “How much energy is needed to run a wireless network?,” IEEE Communications Magazine, vol.18, no.5, pp. 40-49, October 2011.

[29] A. De Domenico, E. Calvanese Strinati, A. Capone, “Enabling Green cellular networks: A survey and outlook,” Computer Communications, Volume 37, pp. 5-24, January 2014.

[30] ICT FP7 FIT4GREEN, Project Deliverable D3.3, “Presentation of the full-featured federated energy-consumption models,” March 2012. www.fit4green.eu/sites/default/files/attachments/documents/D3.3_final.pdfwww.fit4green.eu/sites/default/files/attachments/documents/D3.3_final.pdf

[31] T. Werthmann, H, Grob-Lipski, and P. Proebster, “Multiplexing gains achieved in pools of baseband computation units in 4G cellular networks,” Proceedings of the IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), September 2013.

[32] Intel Xeon Processor 5500 Series, [online document, accessed April 2014] http://download.intel.com/support/processors/xeon/sb/xeon_5500.pdf

[33] P. Monti, S. Tombaz, L. Wosinska, and J. Zander, "Mobile backhaul in heterogeneous network deployments: Technology options and power consumption," 14th International Conference on Transparent Optical Networks (ICTON), pp.1,7, 2-5 July 2012.

[34] NGMN Alliance, "Small cell backhaul requirements," White Paper, June 2012.

[35] iJOIN, Deliverable D2.1 “State-of-the-art and promising candidates for PHY layer approaches on access and backhaul network,” November 2013.

[36] iJOIN, Deliverable D4.1 “Report on SotA and requirements for network-layer algorithms and network operation and management,” November 2013.

[37] Open Multi-Processor (OpenMP), [online: http://openmp.org/wp/]

[38] Open Message Passing Interface (OpenMPI), [online: http://www.open-mpi.org/]

[39] D. Wübben, P. Rost, J. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, “Benefits and impactImpact of cloud computingCloud Computing on 5g signal processing”. To appear in the 5G special issue ofSignal Processing”, IEEE Signal Processing Magazine, November 2014.

[40] A. Maeder, M. Lalam, A.D. Domenico, E. Pateromichelakis, D. Wübben, J. Bartelt, R. Fritzsche, P. Rost, “Towards a Flexible Functional Split for Cloud-RAN Networks”, EuCNC'14

[41] Blaise Barney, Lawrence Livermore - National Laboratory, “POSIX Threads Programming”, [online: https://computing.llnl.gov/tutorials/pthreads/]

[42] Kendall Square Research, [online: http://en.wikipedia.org/wiki/Kendall_Square_Research]

[43] Kernel Based Virtual Machine, [online: http://www.linux-kvm.org/page/Main_Page]

[44] N. Sapountzis, S.Sarantidis, T.Spyropoulos, N.Nikaen, U.Salim, “Reducing the Energy Consumption of Small Cell Networks subject to QoE Constraints”, Globecom 2014

[45] EARTH project, “D2.3 - Energy efficiency analysis of the reference systems, areas of improvements and target breakdown,” [online: https://www.ictearth.eu/publications/publications.html]

[46] ASOCS Ltd, [online: http://www.asocstech.com/]

[47] G. Li, S. Zhang, X. Yang, F. Liao, T. Ngai, S. Zhang, and K. Chen, “Architecture of GPP based, scalable, large-scale C-RAN BBU pool,” in IEEE GLOBECOM 2012 Workshops, International

Workshop on Cloud Base-Station and Large-Scale Cooperative Communications, Anaheim, CA, USA, Dec.2012.

Annex A Power consumption models of iJOIN architectural entities

In the following, the power consumption of each individual network element is discussed. Furthermore, some examples of measures are provided to correlate and obtain an idea on the order of magnitude of each element's power consumption, depending on the cells' load (which is interrelated to the cells' RF output power).

A.1 iSC Power Consumption

The FP7 EARTH has investigated how the power consumption of distinct components of several eNBs, such as power amplifier, baseband engine, main supply, and active cooling, depends on the transmission bandwidth, the transmission power, and the number of radio chains/antennas [28]. Furthermore, it was found that a linear function of the transmission power can approximate very well the generalized model.

To adopt the aforementioned model for approximating iSC power consumption, we have taken into account the functional split. Therefore, its power consumption will be bounded by the two extreme cases: 1) RRH and 2) complete small cell, respectively (see Figure A-1a). RRHs are considered as low complexity nodes that solely perform RF operations and rely on self-backhauling (P_{BB} =0). On the other hand, complete small cells perform all the based band (BB) operations (P_{BB} =6.8 W). Table A-1 reports the power model and the associated parameters to estimate the power consumption of iSCs [28], considering two (per-antenna) maximum transmit power, i.e., 24dBm ($P_{\mathrm{Tx},1}$) and 30dBm ($P_{\mathrm{Tx},2}$). It is worth mentioning that for the iSC power model:

1. no active cooling is considered,
2. iSCs may enter a low consumption sleep mode where only the power amplifier (PA) is turned off when no data is received or transmitted (BB engine reductions due to sleep mode are not considered here for simplicity), and
3. PA power consumption is approximated as a linear function of the PA output power (for further details see [45]).

Table A-1: Power consumption model for the iSC and exemplary realistic parameter values

iSC	$P_{\mathrm{iSC}\text{-}n} = \dfrac{N_{\mathrm{ant}} \dfrac{W}{10\ \mathrm{MHz}} \cdot \left(P_{\mathrm{BB}} + P_{RF}\right) + y_n \cdot P_{\mathrm{PA\text{-}max}}}{\left(1-\sigma_{\mathrm{DC}}\right)\cdot\left(1-\sigma_{\mathrm{MC}}\right)}$		
Bandwidth (W)	10 MHz	PA max consumption ($P_{\mathrm{PA\text{-}max}}$)	0.8W if $P_{TX,1}$ 3.2W if $P_{TX,2}$
# antennas per iSC (N_{ant})	2	DC-DC conversion losses (σ_{DC})	6.4 %
BB consumption (P_{BB})	[0 ; 6.8] W	Main Supply losses (σ_{MC})	7.7 %
RF consumption (P_{RF})	0.8W if $P_{TX,1}$ 1.5W if $P_{TX,2}$	Load of cell n (y_n)	0 – 100 %

A.2 RANaaS Platform Power Consumption

To obtain an accurate estimation on the power consumption of the RANaaS platform due to BB processing moved from iSCs, we use of a model from the IT world. Fit4Green has investigated the power consumption for IT resources of data centres [30]. In particular, results for the various computing style servers are provided using a monitoring tool and a generic power consumption prediction model. Considering the measurement results, it can be observed that a linear model approximate well the server power consumption versus its CPU workload.

Considering the RANaaS as an enclosure hosting several identical ISS Blade servers equally sharing the requested workload, the servers' processing capacity will define how many servers are required to process the system BB-related workload. Therefore, the overall power consumption due to BB processing at the RANaaS platform will be the sum of the power consumption at each of the required servers.

Furthermore, Werthman et al. have recently investigated the relation between the CPU workload and the cell load, and they have defined the resource effort required to serve an UE as a function of the number antennas, the modulation bits, the code rate, the number of spatial MIMO-layers, and the allocated frequency resources in DL [31]. Since in the iJOIN architecture some functionality can be moved towards the RANaaS, we extend this work and introduce an average sum to approximate the total average RANaaS workload required to serve all UEs. Therefore, the Giga-Operations-per-Second[1] (GOPS) required at RANaaS will depend on the number of iSCs, their load, the system bandwidth, the number of antennas per iSC, the average number of data bits per symbol per user, and the average number of MIMO layers (see Table A-2). The RANaaS power consumption with respect to the small cell RF output power for different BB shift is shown in Figure A-1b).

Table A-2: Power consumption model for the RANaaS and exemplary realistic parameter values

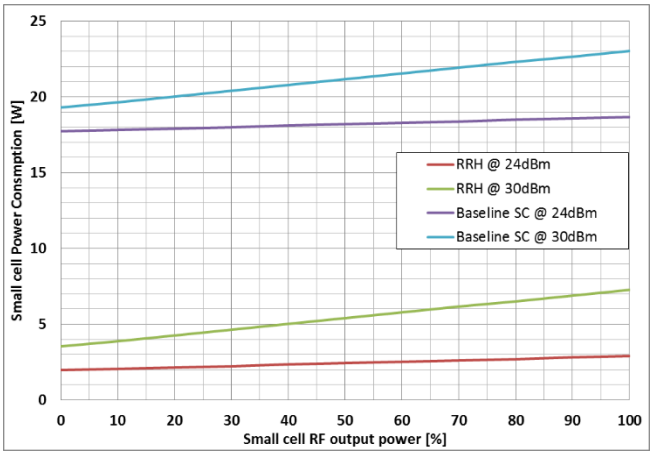
<table>
<tr><td>RANaaS</td><td colspan="3">$P_{\text{RANaaS}} = \left\lceil \frac{X(y)}{X_{\text{Cap}}} \right\rceil \cdot \left(P_0^{\text{srv}} + \sum_{n=1}^{N_{\text{iSC}}} y_n \cdot \frac{\frac{W}{10\ \text{MHz}} \cdot \left(30N_{\text{Tx}} + 10N_{\text{Tx}}^2 + 20\frac{e_{\text{MSC}}}{6} e_{\text{MIMO}} \right)}{X_{\text{Cap}} \left\lceil \frac{X(y)}{X_{\text{Cap}}} \right\rceil} \Delta_p^{\text{srv}} P_{\text{max}}^{\text{srv}} \right)$</td></tr>
<tr><td>BB processing (in GOPS) moved from iSC into RANaaS</td><td colspan="3">$X = \sum_{n=1}^{N_{\text{iSC}}} y_n \cdot \beta_{\text{BB}} \frac{W}{10\ \text{MHz}} \left(30N_{\text{ant}} + 10N_{\text{ant}}^2 + 20\frac{e_{\text{MSC}}}{6} e_{\text{MIMO}} \right)$</td></tr>
<tr><td>Server Capacity (X_{Cap})</td><td>324 GFLOPS</td><td>Consumption at Server Max Workload ($P_{\text{max}}^{\text{srv}}$)</td><td>215 W</td></tr>
<tr><td>Server idle consumption (P_0^{srv})</td><td>120 W</td><td>GOPS/Watt cost factor (c_{BB})</td><td>160</td></tr>
<tr><td>Linear model slope (Δ_p^{srv})</td><td>0.44</td><td># iSCs in veNB (N_{iSC})</td><td>5 - 20</td></tr>
<tr><td>Average # of antennas used to serve a UE (N_{Tx})</td><td>2</td><td>% of BB processing moved into RANaaS from each iSC (β_{BB})</td><td>0 – 100 %</td></tr>
<tr><td>Average # of data bits per symbol per UE (e_{MCS})</td><td>4/3</td><td>Average # of spatial MIMO layers used per UE (e_{MIMO})</td><td>1.1</td></tr>
</table>

[1] It is noted that the processing capacity of the server is expressed in Giga-FLOPS (GFLOPS) [32]; however, it can be converted in GOPS, and in this work we use a 1:1 ratio as a conservative estimation.

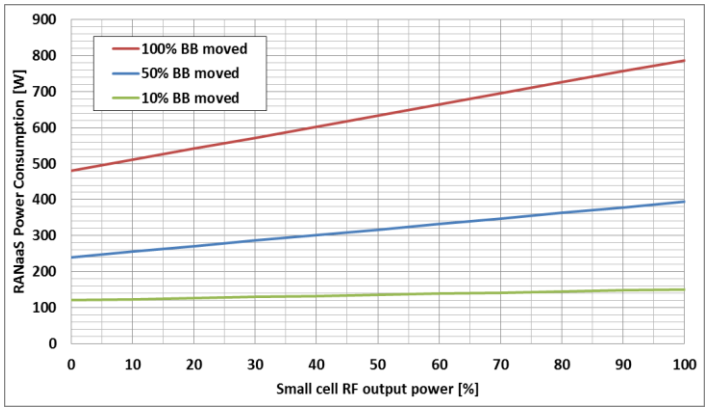
A.3 Backhaul Power Consumption

The last important element that we have modelled is the backhaul network. In general, centralised systems have notable backhaul load; therefore, power consumption due to data transport and switching can become a significant percentage of the total system power consumption [29].

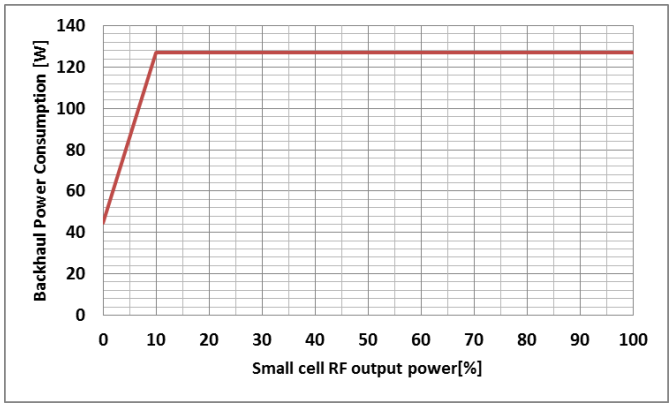
Monti et al. provided some basic power consumption models for data transport through various backhaul technologies and topologies in small cells [33]. Considering iSCs with microwave links and omitting iTNs for simplicity, the backhaul power consumption can be estimated by modifying this model in accordance with the iJOIN architecture; backhaul power consumption shall scale with the power for transmitting and receiving the aggregate backhaul traffic at any iSC, the number of iSCs in the system, the average number of microwave antennas per iSC, and the power consumption of switches at any iSC. Note that the power consumption at any switch will depend on the aggregated traffic at the associated iSC and its maximum capacity. Moreover, the power consumption for transmitting and receiving the aggregate backhaul traffic will generally depend on the traffic conditions. In this work, we consider a two-step function (low/high capacity traffic), where the two capacity regions are distinguished by a single threshold. Our analysis shows that for generic small cells, the backhaul always operates in low capacity region, which results in flat power consumption for medium/high cell RF output power (see Figure A-1c)).



a)



b)



c)

Figure A-1: a) Complete small cell and RRH power consumption with respect to different RF output power and power constraints. b) RANaaS power consumption with respect to the small cell RF output power for different BB shift options; c) Backhaul Power consumption

The question that arises next is how backhaul traffic load can be translated into cell load in current LTE-based RAN. For this, we need to consider the iSC maximum bits-per-second capacity and the non-negligible overheads from X2 U- and C-plane, the transport protocol, and the IPsec [34]. Accordingly, Table A-3 presents the backhaul power model and the relevant parameters with exemplary realistic values. Note that iSC available capacity is evaluated assuming a single carrier with 10MHz bandwidth, 2x2 MIMO, 64QAM,

and 28% control overhead [34]. In addition, we consider that backhaul links can enter in idle mode for energy saving.

Table A-3: Power consumption model for the Backhauling and exemplary realistic parameter values

<table>
<tr><td>Backhaul</td><td colspan="3">$P_{\text{Bh}} = \sum_{n=1}^{N_{\text{iSC}}} P^{n}_{\text{switch}}\left(y_n\right) + N^{n}_{\text{mw-ant}} P^{n}_{\text{link}}\left(y_n\right)$</td></tr>
<tr><td>Switch power consumption</td><td colspan="3">$P^{n}_{\text{switch}} = \begin{cases} 0, & N^{n}_{\text{mw-ant}} = 1 \text{ or } y_n = 0 \\ P_s \cdot \left\lceil \dfrac{y_n \cdot \left[Y_{\max} \cdot f_{\text{cell-Bh}}\% \right]}{C_{\text{switch}}} \right\rceil, & \text{otherwise} \end{cases}$</td></tr>
<tr><td># microwave antennas per iSC ($N_{\text{mw-ant}}$)</td><td>2</td><td>Switch maximum capacity (C_{switch})</td><td>36 Gbps</td></tr>
<tr><td>Switch basic consumption (P_s)</td><td>53 W</td><td>Average cell capacity ($Y_{\max}$)</td><td>86.4 Mbps</td></tr>
<tr><td>% increase from cell load to backhaul traffic ($f_{\text{cell-Bh}}$)</td><td>128 %</td><td></td><td></td></tr>
<tr><td>Backhaul link consumption</td><td colspan="3">$P^{n}_{\text{link}} = \begin{cases} P_{\text{idle}}, & y_n = 0 \\ P_{\text{low-traffic}}, & 0 < y_n \le \dfrac{C_{\text{thr}}}{Y_{\max} \cdot f_{\text{cell-Bh}}\%} \\ P_{\text{high-traffic}}, & y_n \ge \dfrac{C_{\text{thr}}}{Y_{\max} \cdot f_{\text{cell-Bh}}\%} \end{cases}$</td></tr>
<tr><td>Node power region for idle/low/high traffic conditions ($P_{\text{idle/low/high-traffic}}$)</td><td>22.2 / 37 / 92.5 W</td><td>Traffic threshold between low/high power regions (C_{thr})</td><td>500 Mbps</td></tr>
</table>

Annex B CT interactions in WP3

CT 3.1	
Main functional impact	Resource Allocation
Impacted domain/resources	Backhaul/Channel
Main acting entity	RANaaS
Distributed/centralized scheme	Centralized
Specific signalling required	Yes: iSC/iTN to RANaaS for CT3.1 BH channel conditions RANaaS to central-iTN for CT3.1 BH path selection RANaaS to central-iTN for CT3.1 BH channel allocation
Operational time scale	Time scale in terms of seconds (or less)
Functional dependencies	Possible dependency with CT4.4 “Routing and Congestion Control”
Functional split constraints	Requires centralized scheduling at the RANaaS
Additional information	• CT3.1 could operate together with CT3.2 “Partly de-centralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments”. CT3.2 deals with cell selection process which is un-touched during the BH routing and scheduling procedure. However, this might require coordination between CT3.2 as soon as they operate at the same time scale. • CT3.1 is not always compatible with CT3.3 “Energy-Efficient MAC/RRM at Access and Backhaul”. In particular, the discontinued transmission proposed by CT3.3 might have impact on the path selection and link scheduling process, which is mainly decided based on the channel conditions / traffic. • CT3.1 can be implemented with CT3.4 “Computational Complexity and Semi-Deterministic Scheduling”. CT3.4 performs (long-term and short term) user scheduling, whereas CT3.1 operates on top of that by assigning BH links and flows per link. These CTs do not collide; however CT3.1 can impose some additional constraints to CT3.4 for the BH availability. • CT3.1 can be implemented with CT3.5 “Cooperative RRM for Inter-Cell Interference Coordination in RANaaS”, which deals with Inter-cell RRM. These CTs do not collide; however CT3.1 can impose some additional constraints to CT3.5 for the BH availability. • CT3.1 could be implemented together with CT3.7 “Radio resource management for scalable multi-point turbo detection/In-network Processing”, since CT3.1 could be used to route traffic from users not involved in an MPTD processing. • CT3.1 is partially compatible with CT3.8 “Radio Resource Management for In-Network-Processing”, which investigates

	RRM for the uplink; however this might require coordination between the two CTs. • CT3.1 could operate with CT3.9 "Hybrid local-cloud-based user scheduling for interference control" which deals with user scheduling in downlink. These CTs do not collide; however CT3.1 can impose some additional constraints to CT3.9 for the BH availability.

CT 3.2	
Main functional impact	Connection Control
Impacted domain/resources	RAN/cell association
Main acting entity	RANaaS
Distributed/centralized scheme	Centralized
Specific signalling required	yes
Operational time scale	seconds
Functional dependencies	CTs that imply coordinated transmission and reception schemes (CT2.2/2.3/2.5) have an impact on this CT CTs where large scale scheduling is implemented are affected by this CT (CT3.4/3.7) CTs related to BH optimization 3.1 and 4.1-4.5 are affected by CT3.2
Functional split constraints	Yes, centralized connection control at the RANaaS
Additional information	• CT3.2 could operate together with CT3.1, since CT3.1 deals with small cell BH scheduling and routing while CT3.2 deals with cell association. These CTs do not collide; however, CT3.1 has to take into account the changes in cell association due to CT 3.2. • CT3.2 is fully compatible with CT3.3, since CT3.3 deals with RF transmission while CT3.2 deals with cell association. • CT3.2 could operate together with CT3.4, since CT3.1 deals with RRM. These CTs do not collide; however, long term scheduling in CT3.4 has to take into account the changes in cell association due to CT 3.2. • CT3.2 is fully compatible with CT3.5, since CT3.5 deals with short term RRM while CT3.2 deals with cell association. • CT3.2 is fully compatible with CT3.6, since CT3.6 deals modelling iJOIN network characteristics. • CT3.2 is fully compatible with CT3.7, since CT3.7 deals with short term RRM while CT3.2 deals with cell association. • CT3.2 is fully compatible with CT3.8, since CT3.8 deals with short term RRM while CT3.2 deals with cell association. • CT3.2 is fully compatible with CT3.9, since CT3.9 deals with short term RRM while CT3.2 deals with cell association.

CT 3.3	
Main functional impact	RAN RF transmission
Impacted domain/resources	RAN/RF transmission
Main acting entity	RANaaS
Distributed/centralized scheme	Centralized
Specific signalling required	yes
Operational time scale	milliseconds
Functional dependencies	CT3.2 has a tight dependency with CTs that focus on radio resource management (CT 3.4, 3.5, 3.7, 3.8, and 3.9). Cell activation and deactivation can be seen as a long term scheduling. Moreover, cooperative short term scheduling will require earlier small cell activation to enable signalling exchange. CT3.3 also affect CT3.1 since BH links can be set idle when a small cell is de-activated.
Functional split constraints	Yes, centralized connection control at the RANaaS
Additional information	• CT3.3 is fully compatible with CT3.1, since CT3.1 deals with small cell BH scheduling and routing while CT3.3 deals with RF transmission. • CT3.3 is fully compatible with CT3.2, since CT3.2 deals with cell association while CT3.3 deals with RF transmission. • CT3.3 is fully compatible with CT3.3, since CT3.2 deals with cell association while CT3.3 deals with RF transmission. • CT3.3 is compatible with CT3.4, since CT3.4 deals with RRM while CT3.3 deals with RF transmission. However, these functionalities are coupled and have to be jointly designed (coordination and signalling exchange are required) • CT3.3 is compatible with CT3.5, since CT3.5 deals with RRM/ICIC while CT3.3 deals with RF transmission. However, these functionalities are coupled and have to be jointly designed (coordination and signalling exchange are required) • CT3.2 is fully compatible with CT3.6, since CT3.6 deals modelling iJOIN network characteristics. • CT3.3 is compatible with CT3.7, since CT3.7 deals with RRM while CT3.3 deals with RF transmission. However, these functionalities are coupled and have to be jointly designed (coordination and signalling exchange are required) • CT3.3 is compatible with CT3.8, since CT3.8 deals with RRM while CT3.3 deals with RF transmission. However, these functionalities are coupled and have to be jointly designed (coordination and signalling exchange are required) • CT3.3 is compatible with CT3.9, since CT3.9 deals with RRM while CT3.3 deals with RF transmission. However, these functionalities are coupled and have to be jointly designed (coordination and signalling exchange are required)

CT 3.4	
Main functional impact	Resource Allocation
Impacted domain/resources	Backhaul/RAN/RF transmission
Main acting entity	RANaaS (and iSCs)
Distributed/centralized scheme	Centralized (partially distributed)
Specific signalling required	iSC to RANaaS: CSI (of variable granularity), pre-selection of RB allocation RANaaS to iSC: RB allocation decisions
Operational time scale	milliseconds
Functional dependencies	CT3.2, CT3.3, CT3.5, CT3.7
Functional split constraints	Scheduling entity at the RANaaS
Additional information	n/a

CT 3.5	
Main functional impact	Resource Allocation
Impacted domain/resources	RAN/Downlink radio resources
Main acting entity	RANaaS
Distributed/centralized scheme	Centralized
Specific signalling required	Yes: iSC –to-RANaaS for CT3.5 Channel State Information RANaaS-to-iSC for CT3.5 RB allocation decisions
Operational time scale	Time scale of milliseconds
Functional dependencies	No
Functional split constraints	Requires Inter-cell RRM at the RANaaS
Additional information	• CT3.5 could operate together with CT3.1 “BH Link Scheduling and QOS aware flow forwarding”, since CT3.1 deals with small cell BH scheduling and routing. These CTs do not collide; however CT3.1 can impose some additional constraints to CT3.5 for the BH availability, which might affect the Inter-cell RRM. • CT3.5 could operate together with CT3.2 “Partly de-centralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments”. CT3.2 deals with cell selection process which is un-touched during the proposed ICIC. • CT3.5 is not always compatible with CT3.3 “Energy-Efficient MAC/RRM at Access and Backhaul”, since it deals also with RRM for small cells from different perspective (having different objective). • CT3.5 cannot be implemented with CT3.4 “Computational Complexity and Semi-Deterministic Scheduling”. CT3.4 performs (long-term and short term) user scheduling and this might collide with CT3.5, which provides a multi-cell user

	scheduling solution in downlink. • CT3.5 could not be implemented together with CT3.7 "Radio resource management for scalable multi-point turbo detection/In-network Processing", since CT3.5 performs RRM in a systematic manner for all the users in a cluster of small cells (needs discussion). • CT3.5 is compatible with CT3.8 "Radio Resource Management for In-Network-Processing", which investigates RRM for the uplink. • CT3.5 could not operate with CT3.9 "Hybrid local-cloud-based user scheduling for interference control" which deals also with the user scheduling in downlink as CT3.5.

CT 3.7	
Main functional impact	Resource Allocation (large scale scheduling)
Impacted domain/resources	RAN/Uplink radio resources
Main acting entity	RANaaS (and iSCs)
Distributed/centralized scheme	Centralised scheme
Specific signalling required	iSC to RANaaS for CT3.7 activation request RANaaS to iSC for CT3.7 information request iSC to RANaaS for CT3.7 information response RANaaS to iSC for CT3.7 activation response RANaaS to iSC for CT3.7 parameters (resource allocation) iSC to RANaaS for CT3.7 deactivation request (tentative) RANaaS to iSC for CT3.7 deactivation confirmation (tentative)
Operational time scale	General framework update could be done every second (less is better through)
Functional dependencies	CT2.2
Functional split constraints	No. However, if CT2.2 processing is done in RANaaS, then functional split at PHY layer after iFFT is preferred
Additional information	• CT3.7 could be implemented together with CT3.1 "Backhaul Link Scheduling and QoS-aware Flow Forwarding", since CT3.1 deals with backhaul routing to the core network essentially. CT3.1 would be used to route traffic from users not involved in an MPTD processing (no side effect so far) • CT3.7 could be implemented with CT3.2 "Partly decentralized mechanisms for joint RAN and backhaul optimization in dense small cell deployments", since CT3.2 deals with cell (re) selection mechanism. CT3.7 assumes the selection is done, while CT3.2 will act on the selection before CT3.7 has to be applied (no side effect so far). • CT3.7 may not be compatible with CT3.3 "Energy-Efficient MAC/RRM at Access and Backhaul" which deals with discontinuous transmission of iSCs in the downlink. CT3.7 requires that the identified iSCs stay up (discussion is needed).

	• CT3.7 may not be implemented with CT3.4 "Computational Complexity and Semi-Deterministic Scheduling", which deals with RRM in a centralised way: long term scheduling done by the RANaaS, while short term scheduling operated at each iSC. CT3.7 is also a centralised RRM CT and is a "concurrent" of CT 3.4. Ideally if CT3.4 only deals with UEs not involved in MPTD, while CT3.7 operates on those specific UEs, then CT3.7 could be implemented together. (discussion is needed) • CT3.7 could be implemented with CT3.5 "Cooperative RRM for Inter-Cell Interference Coordination in RANaaS" which deals with downlink RRM (no side effect so far). • CT3.7 could be implemented with CT3.6 "Utilization and Energy Efficiency" which evaluates those metrics with the iJOIN context (no side effect so far). • CT3.7 may not be compatible with CT3.8 "Radio Resource Management for In-Network-Processing", which is a concurrent uplink RRM method (discussion is needed). • CT3.7 could be implemented with CT3.9 "Hybrid local-cloud-based user scheduling for interference control" which deals with scheduling in the downlink, while CT3.7 operates in the uplink (no side effect so far).

CT 3.8	
Main functional impact	resource allocation
Impacted domain/resources	RAN/Uplink radio resources
Main acting entity	RANaaS, iveC
Distributed/centralized scheme	centralized
Specific signalling required	yes
Operational time scale	typical scheduling time scale
Functional dependencies	CT2.1
Functional split constraints	split within PHY between detection and decoding or between PHY and MAC
Additional information	• CT3.8 may be compatible with CT3.1, because the jointly detected user data symbols or bits need to be forwarded to the RANaaS over the backhaul network. Nevertheless side effects need to be investigated and in general, coordination is required • CT3.8 can be combined with CT3.2, but in addition to a primary cell association (for control channels), also an additional assignment of jointly detecting small cells is performed by CT3.8, which needs to be coordinated • CT3.8 can be combined with CT3.3, since DTX can be considered as a RRM technique • CT3.8 is not compatible with CT3.4, since CT3.8 assumes centralized RRM

	• CT3.8 can be combined with CT3.5, since CT3.8 considers the uplink only, while CT3.5 considers only downlink • CT3.8 is not compatible with CT3.7 since it relies on CT2.1, which is an alternative to CT2.2 (on which CT3.7 relies) • CT3.8 is compatible with CT3.9, since it operates on uplink only, while CT3.9 considers downlink only

CT 3.9	
Main functional impact	Resource Allocation
Impacted domain/resources	Backhaul/RAN/RF transmission
Main acting entity	iSCs (potential extension with RANaaS)
Distributed/centralized scheme	Distributed (potential extension to partially centralized)
Specific signalling required	iSC-iSC, iSC-RANaaS: CSI short terms (when possible), specific signalling (when possible), long term CSI
Operational time scale	scheduling time, specific signalling
Functional dependencies	CT3.1, CT3.2, CT3.3,
Functional split constraints	Scheduling entity at the iSCs
Additional information	• CT3.1 (Backhaul link scheduling and QoS-aware flow forwarding) : • CT3.2 (Partly de-centralized mechanisms for joint RAN and backhaul optimization in dense small cell deployment) : This CT deals with cell selection and can operate with CT3.5 as it works on a different level. • CT3.3 (Energy-Efficient MAC/RRM at Access and Backhaul) optimizes the activation and deactivation of cells and can operate with CT3.5 as it works on a different level. • CT3.4 (Computation complexity and semi-deterministic scheduling). It is impossible to apply both CTs because both offer alternative solutions for different settings and are operating on the same resources. • CT3.5 (Cooperative RRM for Inter-Cell Interference Coordination in RANaaS) It is impossible because both offer alternative solutions for different settings and are operating on the same resources. • CT3.7 (Radio resource management for scalable multi-point turbo detection/In-network Processing). This CT operates on the uplink and is compatible with CT3.9 which operates on the downlink. • CT3.8 (Radio Resource Management for In-Network-Processing) This CT operates on the uplink and is compatible with CT3.9 which operates on the downlink.

