Coordinated Control and Spectrum Management for 5G Heterogeneous Radio Access Networks

Grant Agreement No. : 671639
Call: H2020-ICT-2014-2

Deliverable D5.2
Final Specification and Implementation of the Algorithm from Programmable Radio Access Networks

Version:	1.0
Due date:	31.1.2018
Delivered date:	7.3.2018
Dissemination level:	PU

The project is co-funded by

Authors

Roberto Riggio (CREATE-NET, Editor), Dejan Kostic (SICS), Shaoteng Liu (SICS), Xianfu Chen (VTT), Tao Chen (VTT), Olav Tirkkonen (AALTO), Junquan Deng (AALTO), Sergio Lembo (AALTO), Navid Nikaein (EURECOM), Dimitri Marandin (CMA), George Agapiu (OTE), Kostas Pentikousis (TP), Antonio Cipriano (TCS)

Coordinator

Dr. Tao Chen

VTT Technical Research Centre of Finland Ltd

Tietotie 3

02150, Espoo

Finland

Email: tao.chen@vtt.fi

Disclaimer

Acknowledgement

This report is funded under the EC H2020 5G-PPP project COHERENT, Grant Agreement No. 671639.

Executive summary

In this deliverable we report on the final Algorithms for programmable Radio Access Networks. The scope of the WP5 encompasses RAN sharing, traffic engineering, and network resiliency. The results presented in this document build upon the radio and MAC abstractions devised in WP3 and WP4, and on the COHERENT C3 and RTC implementations provided by WP2.

More specifically we report on:

- Implication of the CAP (Consistency, Availability, and Partition tolerance) theorem on the control plane of a software-defined mobile network (Objective 4).
- Network graph-based mobility management and traffic steering in dense small cell networks (Objective 5).
- RAN sharing aspects, including performance isolation and revenue maximization in multi operators (Objective 6).

The outcomes presented in this document show promising results as far as the benefits expected from the COHERENT control and coordination abstractions and the SDK are concerned. For each technical contribution we specify if such contribution has been implemented using either the C3 or the RTC SDK and if the contribution is RAT-agnostic or RAT-specific.

Moreover, from the scientific dissemination standpoint the results of this last reporting period are encouraging. In particular, at the time of writing the output of WP5 has been published in 15 papers. In particular, 10 papers have been published in top conferences (IEEE CoNEXT and IEEE NOMS among others), 3 papers have been published in top journals and magazines (IEEE Communication Magazine and IEEE Transactions on Network and Service Management), and 2 papers have been published in workshops and as posters. Moreover, at the time of writing three additional papers are under review at various conferences/journals.

The next steps for the validation of the COHERENT abstractions and of the SDK call for further experimentation and evaluation to be conducted in the framework of WP6. Note that not all the novel solutions presented in this document will be integrated in the final COHERENT demonstrator. On the contrary, we plan to prioritize the solutions to be implemented and validated on the testbed according to both their significance in terms of market impact and their suitability to be supported in an effective and efficient manner by the SDK and by the available prototypes. This is done in order to maximize the technical impact of the project given the amount of resources available

Table of contents

List of abbreviations

2G	2nd Generation Mobile Networks
3D	Three-Dimensional
3GPP	Third Generation Partnership Project
4G	4th Generation Mobile Networks
5G	5th Generation Mobile Networks
5GNR	5G New Radio
5G PPP	5G Infrastructure Public Private Partnership
AA	Simulated annealing algorithm
API	Application Programming Interface
ARPU	Average Revenue Per User
BBU	Baseband Unit
BS	Base Station
C3	Centralized Controller and Coordinator
CAPEX	Capital Expenditure
CAP	Consistency, Availability and Partition tolerance
CN	Core Network
CNV	Centralized Network View
CoMP	Coordinated Multipoint
C-RAN	Cloud Radio Access Network
CQI	Channel Quality Indicator
D2D	Device-to-Device
D-RAN	Distributed Radio Access Network
eICIC	enhanced ICIC
eNB	evolved Node B
eV2X	enhanced Vehicle-to-X Communications
EE	Exhaustive search algorithm
EPC	Evolved Packet Core
EU	European Union
FS	Fault tolerant algorithm
GA	Genetic Algorithm
GPP	General Purpose Processors
GSM	Global System for Mobile Communications
GW	GateWay
GWCN	Gateway Core Network
HMN	Heterogeneous Mobile Network
IaaS	Infrastructure as a Service
ICIC	Inter-Cell Interference Coordination
IMT	International Mobile Telecommunications
IoT	Internet of Things
IP	Internet Protocol
ISD	Inter-Site Distance
ITU	International Telecommunication Union
ITU-R	ITU – Radiocommunication Sector
KPI	Key Performance Indicator
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced
M2M	Machine-to-Machine

MAC	Media Access Control
MEC	Mobile Edge Computing
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
MOCN	Multi-operator Core Network
MTC	Machine-Type Communication
MTD	Machine-Type Device
mmWave	Millimeter Wave
MNO	Mobile Network Operator
MVNO	Mobile Virtual Network Operator
NaaS	Network as a Service
NFV	Network Function Virtualization
NGMN	Next Generation Mobile Networks Alliance
OPEX	Operational Expenditure
PCCS	Primary Component Carrier Selection
PCI	Physical Cell ID
PHY	Physical Layer
PNF	Physical Network Function
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RF	Radio Frequency
RRM	Radio Resource Management
RT	Radio Transceiver
SA	Service and System Aspects
SC	Small Cells
SDK	Software Development Kit
SDN	Software Defined Network
SD-RAN	Software-Defined Radio Access Network
SLA	Service Level Agreement
SON	Self Organizing Network
TDD	Time-Division Duplex
TDM	Time Division Multiplexing
TN	Transport Node
UE	User Equipment
V2X	Vehicle-to-X Communications
vBBU	virtual Baseband Unit
vBSC	virtual Base Station Controller
VNF	Virtual Network Function
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WP	Work Package, Working Party

1. Introduction

Fifth generation mobile networks are expected to support vastly different applications and services. Examples include high–definition video delivery, machine–to–machine communications, e-health, device-to-device (including vehicle-to-vehicle) communications, and public safety. Each of these applications imposes different requirements on the network in terms of bandwidth, latency and resiliency. Starting from the assumption that it will not be economically viable to support each use case with a custom mobile network, it becomes of capital importance to be able to tailor the behavior of 5G networks according to requirements of each vertical.

In this deliverable, we report on the specification and evaluation of several algorithms that can be implemented on top of the programmable, heterogeneous RAN enabled by the COHERENT platform. Notice how, not all the algorithms and prototypes described in this document will be actually integrated and tested in the final demonstrator, on the contrary we have prioritized solutions that will be implemented and validated on the integrated testbed according to *both* their significance in terms of market impact and their suitability to be supported in an effective and efficient manner by the SDK and by the available prototypes. The selected solutions are then passed as software packages to WP6 for integration and testing in the final COHERENT demonstrator.

1.1 Outcomes

The work achieved in the framework of WP5 during the final phase of the project contemplates several results covering different individual research areas. The current outcomes show promising results as far as the benefits expected from the COHERENT control and coordination abstractions and the SDK are concerned.

Table 1 – Summary of outcomes

Outcome	C3/RTC	RAT(s)	Code	**Use Cases**
Algorithms for placement and routing of distributed control plane functions [Shaoteng2018]	C3/RTC	Agnostic	Yes	UC3.MN
Algorithms for integer-valued radio resource management (Beam ID [Ahmed2016]	LTE	LTE	No	UC1.RS
Algorithms for End-to-End Wi-Fi RAN Slicing [Coronado2018c]	C3	Wi-Fi	Yes	UC1.RS UC4.GD UC4.SD
Algorithms for Channel–Aware User Association in Software–Defined Wi–Fi Network [Coronado2018a, Coronado2018b]	C3	Wi-Fi	Yes	UC6.MR UC4.GD UC4.SD
Algorithm for traffic–Aware User Association in Heterogeneous RANs [Harutyunyan2018]	C3	Wi-Fi/LTE	Yes	UC6.MR UC4.GD UC4.SD
Algorithms for LTE RAN Slicing [Foukas2016, Ksentini2017]	RTC	LTE	Yes	UC1.RS UC4.GD UC4.SD
Algorithms for Delay-Tolerant Resource Scheduling for VNO [Chen2017]	C3	LTE	No	UC1.RS UC4.SD
Algorithms for Traffic Steering and Resource Allocation [Deng2017mag, Deng2017ICC, Lembo2017]	C3	LTE	No	UC1.SR UC3.CE

Algorithms for flexible functional split in 5G Networks [Harutyunyan18mmw, Harutyunyan18mig, Harutyunyan18flex, Harutyunyan16, Riggio16, Harutyunyan17]	C3	LTE	No	UC1.RS UC1.CO

As it can be seen from the table above, at the time of writing the output of WP5 has been published in 15 papers. In particular, 10 papers have been published in top conferences (IEEE CoNEXT and IEEE NOMS among others), 3 papers have been published in top journals and magazines (IEEE Communication Magazine and IEEE Transactions on Network and Service Management), and 2 papers have been published in workshops and as posters. Moreover, at the time of writing three additional papers are under review at various conferences/journals.

1.2 Organization of this deliverable

This deliverable is organized as follows.

- In Section 2 we tackle the control plane resiliency problem by studying the implication of the Consistency, Availability, and Partition tolerance (CAP) theorem on centrally controlled heterogeneous wireless networks. We also introduce a system architecture for resilient network nodes and we report on some preliminary results leveraging on a real-world prototype (Objective 4).
- In Section 3 the application of centralized algorithms in problems where network-wide optimization of the use of discrete radio resource management is studied. The problems studied in this section are Primary Component Carrier Selection (PCCS), Physical Cell Identifier (PCI) and Beam Identifier assignment, as well as TDD switching point management (Objective 4).
- In Section 4 we present a programmable end–to–end network slicing framework for enterprise WLANs. The proposed framework builds on top of the COHERENT C3 in order to allow programmability, isolation, and customization of end-to-end network slices (Objective 6).
- In Section 5 we present Wi–Balance, a joint channel selection and user association scheme for Wi–Fi–based WLANs. A constraint programming algorithm is designed to isolate possible collision domains among the APs. Then a user association scheme capable of rescheduling traffic across the available APs is proposed and implemented (Objective 5)
- In Section 6 we formulate a traffic–aware user association problem aiming at optimizing resource utilization in a heterogeneous Wi-Fi/LTE RAN. Then we implement and test a heuristic in real–world conditions (Objective 5).
- In Section 7 we present a RAN slicing framework for LTE networks, built on top of the COHERENT RTC, in order to allow programmability, isolation, and customization of end-to-end network slices (Objective 6)
- In Section 8 we proposed a solution allowing virtual network operators to lease resources from a large-scale RAN with a particular focus on delay tolerant applications (Objective 4).
- In Section 9 we present a hierarchical control framework to address network management problems related to traffic steering, offloading & D2D management in a multi-RAT situation with mmWave & sub-6GHz multi-connectivity (Objectives 4 and 5).
- In Section 10 we present a set of algorithms for dynamically choosing the optimal functional split in a virtualized 5G network. The proposed approach allows to select the optimal demarcation point between vRP and R-TP according to the actual network load and to the available frontal resources (Objective 6)

2. Algorithms for placement and routing of distributed control plane functions

2.1 Overview

Logically centralizing the control and coordination of the programmable RAN has advantages in flexibility, programmability and can provide a global view over resource allocation. In COHERENT, the centralized control and coordination task resides in the C3 entity [CoherentD2.2]. It collects the Centralized Network View (CNV) from the network nodes and exposes it to the network control applications. In its most general form, the C3 entity should be composed of several physically distributed C3 control instances with each of them in charge of a portion of the RAN. This is due to that in a programmable RAN, a single control instance might not be enough to serve all the requests in the network. It would become a bottleneck in terms of processing power, memory, or input/output bandwidth, latency and so on. Furthermore, if the control instance fails, the whole control service may become unavailable. Thus, for the purpose of improving reliability and scalability, deploying multiple control instances to implement a physically distributed but logically centralized control plane is almost an inevitable choice.

However, as the number of deployed control instances increases, there is a significant risk that the consequent inter-controller traffic grows into an unacceptable overhead. Regardless of the consistency level (strong vs. eventual), updating shared state at one of the n controllers intuitively requires a one to many style communication to update the $n-1$ remaining instances. Observations and investigations in [Koponen2010][Berde2014] also suggest this dramatic increase in the communication overhead for maintaining shared state among controllers.

In existing controller placement approaches, the importance of considering the control plane traffic as part of the solution is usually overlooked. Dealing with the traffic associated with a certain controller placement is typically ignored, although control plane traffic flows have to be forwarded timely and reliably through a network infrastructure with varying link capacities, availability, and other networking properties. Control traffic congestion, for example, is especially destructive since it may degrade control service performance, or worse, cause availability issues - the latter cannot be tolerated in e.g., services critical to human safety.

To this end, we propose a new optimization process to deal with the control plane deployment problem. We advance the current state of the art by 1) proposing a novel formalization of the problem, enabling 2) tuning of reliability and bandwidth requirements. The proposed approach is also contained in an accepted paper which will be published in [Shaoteng2018]. Essentially, by analyzing the challenges and complexity of the controller placement and traffic routability problem, we introduce a generic optimization process. We specify each step of the process along with guiding implementation examples. Unlike existing approaches, our optimization process adds the extra steps needed for quantifying the consequences of deploying a control plane solution that fulfills the specified reliability and bandwidth requirements. As a powerful prediction tool, the approach can be used by service providers and operators to fine-tune control plane deployment policies.

2.2 Relationship with the previous deliverable

In a previous deliverable [CoherentD5.1] (Section 2), we have accounted for a RAN configuration where the C3 entity is only composed of a single C3 Instance. To deal with the C3 instance failure, we propose a leader election method that will elect a new leader node from the candidate nodes, in order to launch a new C3 instance to repair the control service.

In this deliverable, we address the more general RAN configuration where the C3 entity is distributed. We focus on how to effectively deploy multiple distributed C3 instances over a certain RAN topology to tolerate failures and enhance the overall reliability of the control system, while still guaranteeing the routability of the control traffic.

2.3 Background and motivation

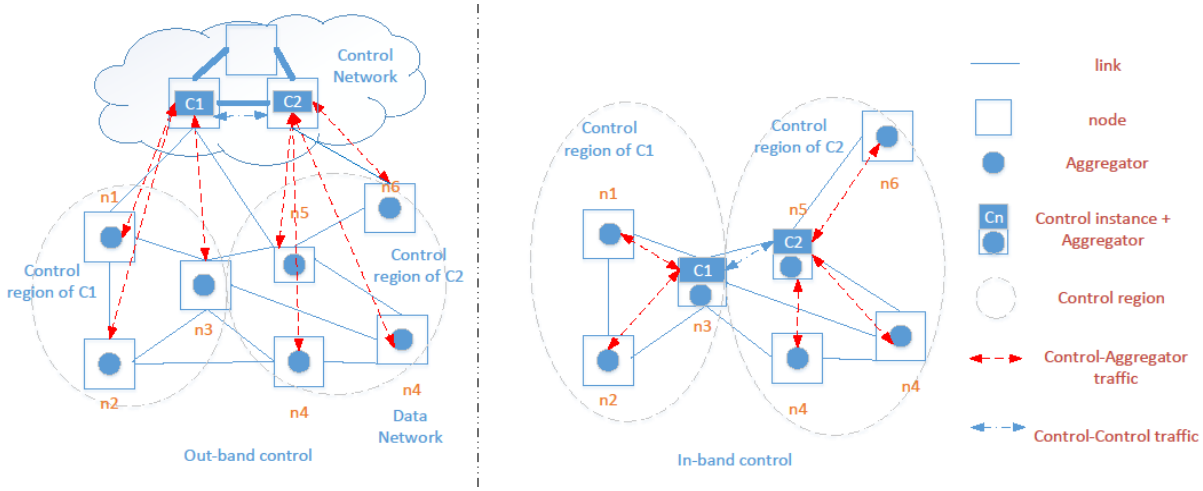


Figure 1 The schemes of a distributed control plane

Figure 1 illustrates two typical cases of the distributed control plane of a programmable RAN. Notice that with the term ***aggregator*** here we address any programmable network element in the COHERENT data plane. It can refer to either Transport Node (TN) or Radio Transceiver (RT) [CoherentD2.2]. In any case, the aggregator acts as a programmable data forwarding device. The term ***controller*** here refers to a distributed control instance C3, which mainly deals with non latency-sensitive control tasks, according to [CoherentD2.2]. The *out-of-band* control setting is shown on the left of Figure 1. In this case, all the controllers are connected with dedicated networks and running in dedicated nodes. The *in-band* control case, where both control and data traffic share the same network, is illustrated on the right of Figure 1. A control instance can in this case be co-located with an aggregator in one node. In both cases, the control of the aggregators is distributed over two controllers, C1 and C2 (Figure 1).

Coordinating distributed controllers, appearing as one single logical control entity, requires that system events and network state information are shared between the controllers with a certain level of consistency. In general, the behavior of such inter-controller traffic depends on the control application and varies with the size and intensity of information transactions, number of controllers, as well as the consistency model. Hence, inter-controller traffic can become potentially expensive in terms of communication overhead in addition to control messages. For example, as observed in the evaluations of [Koponen2010], a single update of shared information may generate 4*n* transactions in the control plane, where *n* is the number of the controllers.

In this deliverable, our work focusses on addressing the controller deployment problem. Control plane deployment here refers to the planning of the controller placement and associated control traffic in the distributed control plane. There are two major challenges: first, the control instances must be placed in a way to optimize the gain related to e.g. reliability and scalability. This includes decisions on how many control instances to use, where to place them in the topology, how to define their control regions. The controller placement problem in general is NP-hard. Second, we must verify whether the control traffic introduced by a placement solution can be scheduled on the underlying network without overloading any link. It can be modelled as a multi-commodity flow problem [Agarwal2013]. Depending on the underlying routing mechanisms of the infrastructure, if flows can be splittable [Kandula2007], then the problem can be formulated as a linear programming (LP) problem; otherwise, it is a Mixed Integer Linear Programming (MILP) problem [Zhang2010], which is known to be NP-hard. Moreover, the number of decision variables inside the problem increases exponentially with the topology size. Thus, even if it is a LP, it is still challenging to solve it in polynomial time.

We propose a generic optimization process to deal with the controller deployment problem. Although in this deliverable the controller deployment solution targets at optimizing reliability, the proposed generic optimization process can also be extended to deal with other optimization objectives, such as

link bandwidth requirements, control latency, load balancing and so on, which are marked as our future works.

2.4 Problem formulation

According to [Rupe2003], the reliability of a system is defined as the probability that the system operates without failure in the interval [0, t], given that the system was performing correctly at time 0. In this section, we describe our optimization approach that targets at the *control system reliability*. The control system reliability here refers that the minimum reliability among all the aggregators (noted as *Rmin*). Here, the reliability of an aggregator corresponds to the probability that an operational aggregator is connected to at least one operational controller during the interval.

Let $G(V = N \cup M, E)$ be a graph representing a network topology, where V and E denote nodes and links, respectively. Moreover, let N denote the set of nodes holding aggregators and M a candidate set of nodes eligible for hosting control instances. Further, each aggregator $n \in N$ and each control instance have a given probability of being operational, denoted by p_n and p_c, respectively. Analogously, links $(u, v) \in E$ are operational with probability $p_{u,v}$. We assume different i.i.d. operational probabilities for links, aggregators, and controller instances. Note that, this probability can be set based on expert knowledge or inferred by learning from the system monitoring data.

We use binary variables y_i, where $y_i = 1$, if node $i \in M$ hosts a control instance, and $y_i = 0$ otherwise. Let $C = \{i | y\_i = 1, i \in M\}$ denote the set of deployed control instances. Let binary variable $a_{ij} = 1$ if aggregator $j \in N$ is controlled by the controller in $i \in C$, otherwise $a_{ij} = 0$. Although each aggregator j can only be controlled by one controller at a time, it can have multiple backup controllers. The reliability of aggregator j is represented as $R(G, j, C)$ (among $|C|$ control instances), capturing the probability of aggregator j connecting with at least one of the operational control instances. The control system reliability equals to $Rmin = min(R(G, j, C), \forall j \in N)$. Here we define a reliability threshold β and require that $R(G, j, C) \geq \beta, \forall j \in N$. Thus, the problem of improving $Rmin$ is equivalent to maximize β.

For the traffic routing problem in programmable networks, we can formulate it as a multi-commodity flow problem [Agarwal2013] by taking flow splitting into account [Kandula2007]. Let u_e be the reserved bandwidth capacity on each link $e \in E$ for control plane traffic. Suppose (s_f, t_f) being the $(source, sink)$ of control traffic flow f. Let d_f denote the demand (throughput) of f. Let $F = \{f = (s_f, t_f, d_f)\}$ be the set of all the control flows. Let $F_c \subset F$ be the inter-controller traffic that $F_c = \{f = (s_f, t_f, d_f) | s_f \in C; t_f \in C\}$. Let κ_f denote all the non-loop paths for $f \in F$, and let $\kappa = \cup_f \kappa_f$. Let variable $X(K)$ denote the amount of flow sent along path $K, \forall K \in \kappa$. Then, the reliable controller deployment problem can be formulated as follows:

$$
\begin{aligned}
&maximize && \beta \\
&s.t: && \sum_{i \in C} a_{ij} = 1, \quad \forall j \in E && (1) \\
& && \sum_{i \in M} y_i > 1 && (2) \\
& && R(G, j, C) \geq \beta, \forall j \in N && (3) \\
& && \sum_{K \in \kappa_f} X(K) \geq d_f, \forall f \in F && (4) \\
& && \sum_{K: e \ni K} X(K) \leq \mu_e, \forall e \in E && (5) \\
& && y_i, a_{ij} \in \{0,1\} && (6) \\
& && X(K) \geq 0, \forall K \in \kappa && (7)
\end{aligned}
$$

Note that the above formulation of the control plane management problem is general, and covers both in-band and out-of-band control cases. For example, $M \subseteq N$, corresponds to an in-band control

problem formulation, whereas $N \cap M = \emptyset$, corresponds to the out-of-band case. Although the out-of-band case may additionally require that the paths for the inter-controller traffic should be limited within the control network, this requirement has already been implicitly included in the definition of the set κ_f. The κ_f is defined as the set of all the possible paths for a certain flow f. A possible path for flow $f \in F_c$ in the out-of-band case can only be selected among links belonging to the control network.

The difference between this formulation and the traditional reliable controller placement problem [Ros2016] is that we take into account the constraints (4), (5), (7) on control traffic, which, to our knowledge, has not been addressed previously.

This problem is hard in terms of computational complexity for the following reasons. First, constraints (1), (2), (3), (6) constitute a fault tolerant facility location problem. Second, constraints (4), (5), (7) form a multi-commodity flow problem. Third, the reliability $R(G, j, C)$ is a non-linear constraint - computing $R(G, j, C), \forall\, j \in N$ based on links and nodes with non-equal probabilities of operation can be an NP-hard problem by itself [Ros2016]. Fourth, the number of variables $X(K)$ can be exponential in the number of nodes and edges.

2.5 The proposed approach

The optimization process for finding a reliable and routable solution to the optimization problem described in the previous section is illustrated in Figure 2. The process starts with an initial value for $\beta \in (0, 1)$. It checks whether a feasible deployment solution exists under the β. Based on the checking results, the process optimizes the value of β with a binary search method.

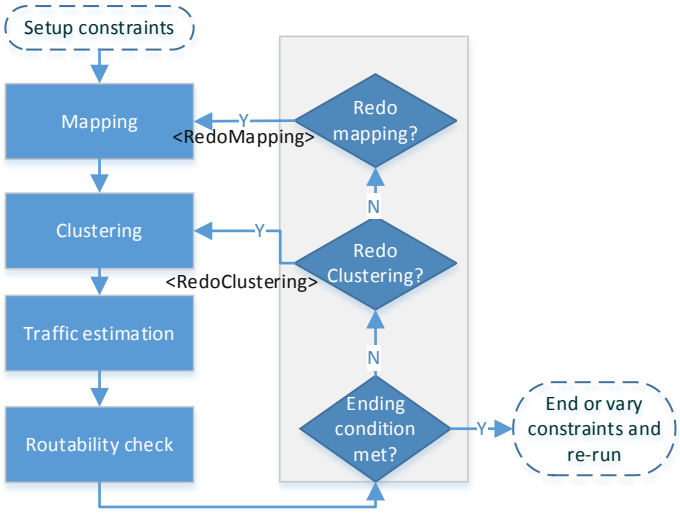


Figure 2 The general steps in the approach

Next, we outline each step of the process.

2.5.1 Mapping

The placement of the control instances given a network topology is provided in this step. The input here includes (but is not limited to) network topology properties including link bandwidths and failure probabilities. The output is a controller location map which describes which nodes in the graph are selected to host controller instances.

Several different algorithms can be applied for this step. Here, we show a simulated annealing algorithm for mapping (SAM) outlined in Algorithm 1. In short, the algorithm will after initialization (lines 1-4) generate a new mapping in each round when receiving the redoMapping signal, carried out as follows:

1) The cost cost_{new} (i.e. a user-defined cost function of the last mapping solution C) is used to decide whether the last mapping plan can replace the current mapping, based on a *Transition Probability* Function $P = exp\frac{cost\_\{new\}-cost\_\{old\}}{T}$ (line 5-10).
2) A new mapping is generated based on the current mapping by randomly adding or removing a control instance (line 11).
3) The Rmin of the new mapping is computed (line 12).
4) The temperature T used in simulated annealing is decreased by a factor of α (line 13).

The simulated annealing algorithm for mapping
Input signal: *RedoMapping* with inputs $\mathbf{C}, \mathbf{cost}_{new}$
 # Initialization:
1: Choosing $\|\mathbf{C}\|$ nodes from $\mathbf{V}$ as the initial set of controllers $\mathbf{C}$
2: Calculate Rmin = $\mathbf{Min}\ (\overleftarrow{\mathbf{R}(\mathbf{G},\mathbf{j},\mathbf{C})}, \forall\ \mathbf{j} \in\ \mathbf{V})$
3: $\mathbf{P_{current}} = \mathbf{C}, \mathbf{T} = \mathbf{T_{Initial}}$
4: **Output** Rmin, C
5: **Upon signal <RedoMapping| C, cost_new>**
6: **If** $\mathbf{T} == \mathbf{T}_{Initial}$
7: $\mathbf{cost_{old}} = \mathbf{cost_{new}}$
8: **Else If** $\mathbf{P}(\mathbf{cost}_{old}, \mathbf{cost}_{new}, \mathbf{T}) \geq \mathbf{random}(\mathbf{0},\mathbf{1})$
9: $\mathbf{P_{current}} = \mathbf{C}$
10: **End If**
11: $\mathbf{C} = \mathbf{getNextSolution}(\mathbf{P_{current}})$,
12: Calculate Rmin = Min $(\overleftarrow{\mathbf{R}(\mathbf{G},\mathbf{j},\mathbf{C})}, \forall\ \mathbf{j} \in\ \mathbf{V})$
13: $\mathbf{T} = \boldsymbol{\alpha} * \mathbf{T}$
14: Output Rmin, $\mathbf{C}$
15: **End Upon**

Algorithm 1 The simulated annealing algorithm for mapping

The mapping tries to maximize the following cost function:

$$cost = min\left(0, -log_{10}\left(\frac{1-R_{min}}{1-\beta}\right),\ \lambda - 1\right) \qquad (8)$$

The λ is calculated in the routability checking step. It is an indicator on whether control traffic is routable ($\lambda \geq 1$) or not ($\lambda < 1$).

Since directly computing the reliability of each aggregator is NP-hard [Ros2016], the approximation method proposed in [Ros2016] is applied. In general, the method in [Ros2016] computes a lower bound $\overleftarrow{R(G,j,C)}$ of $R(G,j,C)$ by considering all the disjoint paths between aggregator j and the $|C|$ controllers.

2.5.2 Clustering

The control region of a controller is defined by clustering aggregators. The input of this step is the controller location map. The output is an aggregator clustering plan, which defines the aggregator-control instance association.

The algorithm implements simulated annealing for clustering (SAC) and is similar to Algorithm 1. The two main differences relate to the cost function $cost = min\ (0, \lambda - 1)$ and the implementation of the $getNextSolution()$ function that randomly clusters aggregators and controllers towards obtaining a satisfying solution.

2.5.3 Traffic estimation

In this step, the control traffic matrix is estimated according to the placement and clustering results generated by the mapping and clustering steps. Let $f = (s_f, t_f, d_f)$ denote each control traffic flow, where s_f, t_f, d_f represents the source, sink and demand, respectively. The objective of this step is to estimate d_f while s_f and t_f are known from the mapping and clustering steps.

In principle, since the optimization process treats the control traffic model as an input variable, any traffic model can be applied for estimating each d_f. For example, we can model either average or worst-case demands, with either simple linear modelling method or advanced machine learning techniques.

However, as the scope of the current work mainly concerns the optimization process, we just employ a simple traffic estimation model here as an example. Suppose that each aggregator request and the corresponding controller response cost T_{req} and T_{res} bytes, respectively. Furthermore, after dealing with a request, the controller instance sends messages of size T_{state} bytes to each of the other $|C| - 1$ control instances notifying about the network state changes. The message sizes can be set arbitrarily. With these parameters, given the request rate $r_j, j \in N$ of each aggregator, we can simply estimate the traffic between aggregator j and its associated controller is $r_j * T_{req}$ and $r_j * T_{res}$, for aggregator-controller direction and controller-aggregator direction, respectively. The outgoing inter-controller traffic from controller i to another controller is estimated $T_{res} * \sum_{j \in N} a_{ij} * r_j$.

2.5.4 Routability check

In this step, the input is the network topology and all control traffic from the previous step. The output is a decision variable, which indicates whether the control traffic throughput can be scheduled, or not.

The routability check is a multi-commodity flow feasibility LP problem. Solving this problem means dealing with an undesired exponential number of variables, as indicated by the constraints (4), (5), (7). This issue can be circumvented by formulating a maximum concurrent flow problem [Naveen2007] (as (9), (10), (11), (12) suggest), which is easier to solve and equivalent to the multi-commodity flow problem. The formal constraints (10), (11), (12) indicate that if $\lambda > 1$, then all the traffic flows are routable. Now, the dual of the maximum concurrent flow problem has a linear number of variables and an exponential number of constraints. This elegantly allows for solving the problem to a desired level of accuracy using a primal-dual algorithm. For the dual formulation, please refer to [Agarwal2013] [Naveen2007] [George2008].

$$maximize \quad \lambda \qquad (9)$$

$$s.t: \quad \sum_{K:\, e \ni K} X(K) \le \mu_e, \quad \forall\, e \in E \qquad (10)$$

$$\sum_{K \in \kappa_f} X(K) > \lambda d_f, \forall f \in E \qquad (11)$$

$$X(K) \ge 0, \forall K \qquad (12)$$

We use the algorithm designed by Karakostas [George2008] based on fully polynomial time approximation schemes (FPTAS) to solve the problem. By using Karakostas's algorithm [George2008], we can get the near-optimal λ, which is guaranteed within the $1 + \epsilon$ factor of the optimal, within the time complexity of $O(\epsilon^2 |E|^2 \, log^{O(1)} |E|)$, where $|E|$ denotes the number of edges. For detailed theory and proof of the algorithm, please refer to [George2008].

2.5.5 Notes about the proposed approach

Note, that the process is generic and can be extended to include other (single or multiple) requirements related e.g. to load balancing and response delays, by adding proper constraints in the mapping and association step along with a cost function and end conditions. In principle, each step can be viewed as a black-box implementation matching the input and output at each step as previously defined.

2.6 Use case study

Our proposed approach can deal with arbitrary network topology, link bandwidth settings, and traffic parameters. In this section, we exemplify our proposed approach with a usage case study, in which we select a network topology called "Internetmci" from internet topology Zoo (ITZ)[1] [Topologyzoo2011]. For traffic parameters, we set $T_{req} = T_{res} = 128$ bytes, and $T_{state} = 500\ bytes$. We also set the request rate of each aggregator as 500 requests/s [Levin2011] and the reliability of each aggregator, link and controller as 0.9999. We set the bandwidth constraint u_e of each link as 24 Mbits/s.

The case exemplifies maximization of control system reliability *Rmin* using the aforementioned optimization process. The maximum reliability achievable in this case is 0.99989. The corresponding mapping and clustering solution is shown in the figure. Red node denotes locations of controllers. Node number (outside the parenthesis) and corresponding controller association (inside the parenthesis) is labelled on each node.

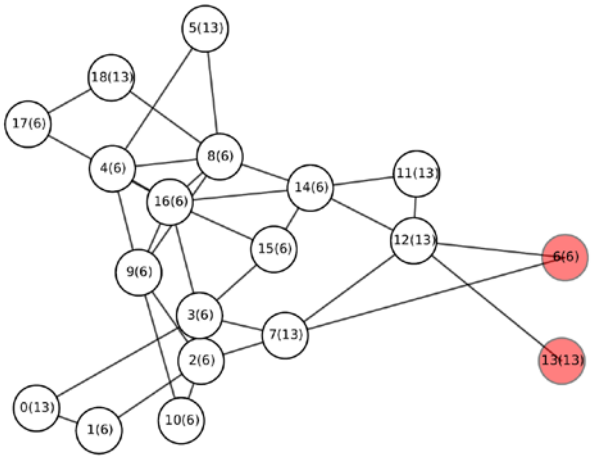


Figure 3 The corresponding deployment plan of controller instances (red), the system reliability achieved is 0.99989, given the bandwidth constraint as 24 MBits/s per link.

2.7 Evaluation

In this section, we evaluate the performance of different implementations of the optimization process, followed by a scaling test on the bandwidth and reliability. The parameters used in the experiment are set as the followings: the request rate of each aggregator varies randomly within [250reqs/s, 750req/s] by the use of a truncated normal distribution, where $\mu = 500$, $\sigma = 500$. Further, the reliability of each link and node is randomly drawn from a Weibull-distribution, with $\lambda = 0.9999$ and $k = 40000$ by considering the long tails in the downtime distribution of WAN links with four nines of mean reliability [Ros2016] [Turner2010]. For traffic parameters, we set $T_{req} = T_{res} = 128$ bytes, and $T_{state} = 500\ bytes$. To effectively display the achieved control system reliability *Rmin* in the figures, we plot the failure probability (i.e., 1-*Rmin*) instead, since it is more suitable for plotting in log-scale.

The main purpose of the evaluation is to illustrate the capabilities and shortcomings of our proposed optimization process. However, our optimization process is in general applicable to other more complicated control services with different traffic parameters.

2.7.1 Evaluation of implementations

Implementation comparisons of the mapping and clustering steps (while holding remaining steps fixed) encompass the following combinations:

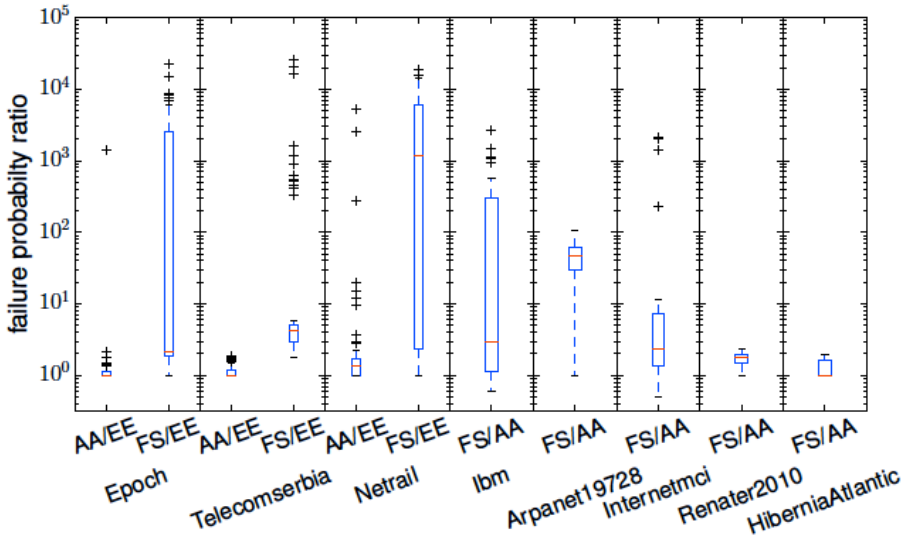
[1] We think that in certain cases, the C3 instances can be deployed in geo-distributed cloud environment, and communicate with each other by using public networks.

1. Exhaustive search algorithm for the mapping step and an exhaustive clustering algorithm for the clustering step (referred as EE). EE can find the theoretically optimum placement.
2. Simulated annealing algorithm for the mapping (SAM) and simulated annealing algorithm for the clustering (SAC) step (referred as AA).
3. Heuristic fault tolerant mapping [Ros2016] and simple closest aggregator-controller clustering (referred as FS).

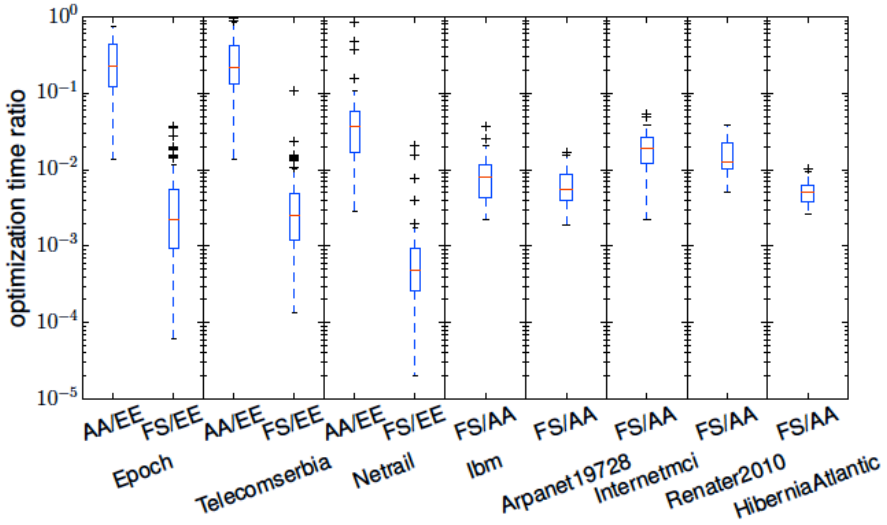
We compare the performance in terms of failure probability and the optimization time for each implementation using EE as a baseline. In all cases the link capacity $\mu_e = u$ varies within $[u/2, u * 3/2]$, randomly drawn using truncated normal distribution with $(u, \sigma = 500)$. Three small, three medium and two large topologies [Topologyzoo2011] are used as test cases. We set u as 8 Mbits/, 24 Mbits/s and 48Mbits for small, medium and large topologies, respectively. This setting is based on the considerations that bandwidth constraint should be sufficient to satisfy at least 3-nine reliability, but not overly large to make it too easy for routing. All results are based on 100 repetitions of the experiments.

In Figure 4 (a), the performance of AA and FS for the small topologies is shown as a ratio relative to the baseline implementation EE in terms of the failure probability achieved. For the medium and large topologies, we only plot the performance ratio between FS and AA, as EE is too slow for getting any result. In Figure 4 (b) the optimization time is shown as a ratio over each other, as the x-axis suggests.

Overall, the results in Figure 4 demonstrate that the outlined optimization process in combination with suitable implementations of the mapping and clustering steps can provide a tunable control plane management solution close to optimum. The choice of methods is a trade-off between the ability to produce close to optimal solutions for different topology sizes and optimization time.



(a) The failure probability ratio



(b) Optimization time ratio

Figure 4 (a) the failure probability ratio; (b) the optimization time ratio for various topologies and implementations.

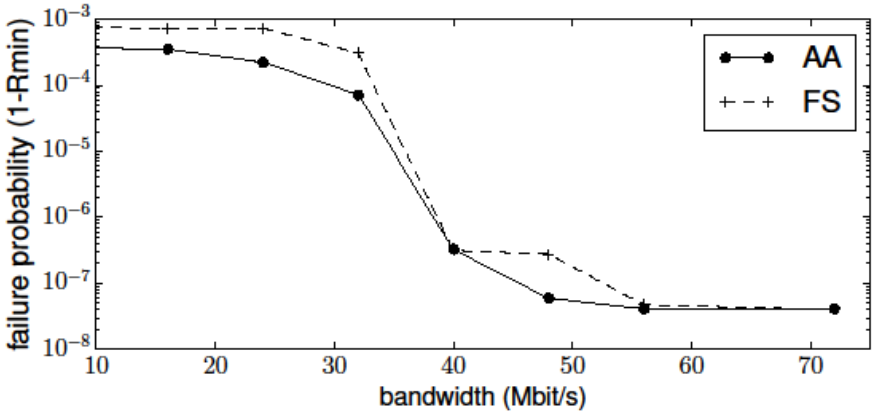


Figure 5 Failure probability versus link bandwidth - the graph can be used to determine the optimal trade-off between required reliability and associated bandwidth demands.

2.7.2 Link bandwidth scaling test

We also systematically quantify the influence on the failure probability relative to an increasing link bandwidth constraint. In general, when scaling up the link capacities in a topology, the failure probability decreases, and hence *Rmin* increases. Figure 5 illustrates this effect for the "Internetmci" topology. As the mean of the link bandwidth distribution increases from 8 Mbit/s to 72 Mbit/s, the median of failure probability with increasing bandwidth decreases from 7×10^{-3} to 4×10^{-8}. Note however, that the decrease does not scale linearly with the bandwidth. By analyzing Figure 5, we are able to quantify the reliability gain relative to a certain bandwidth limit, at around 40Mbit/s. Beyond this point, increasing the bandwidth will only lead to an insignificant increase in reliability.

These experiments demonstrate that service providers and operators can use the proposed optimization process as a practical tool for quantifying the trade-off between bandwidth and reliability gains, enabling development of flexible and fine-tuned controller deployment policies.

2.8 Conclusion

2.8.1 Technical/Scientific Impact

In this deliverable, we tackle a difficult and previously unresolved issue in distributed control plane placement. Simply put, it is necessary to have multiple controllers in a large deployment but keeping

them synchronized incurs significant bandwidth demand. This communication overhead cannot be avoided as all these controller instances have to be updated at some point, either synchronously or asynchronously. We are the first ones to consider the bandwidth overhead and its implication on the controller placement problem.

We take a holistic view in that we also include reliability into account. Our approach can take as input the distribution of reliabilities and determine what bandwidth is needed to ensure correct operation with high probability. Alternatively, our approach can quantify the reliability that can be achieved with a bandwidth limit. This is important as the bandwidth used for synchronization is the cost to pay and is expected to be limited.

2.8.2 Feedback toward development

Our approach is flexible in that it can take an arbitrary communication pattern among the controller instances that occurs during synchronization. One particular feedback toward development would be finding a way to automatically collect the communication pattern and intensity over time. This can be directly fed into our algorithm. Discovering the reliability model over time would also be useful, and this is something else that needs to be logged. For example, node and link uptimes can be collected and processed.

2.8.3 Expected business impact

Our approach is taking into account all practical aspects of deployment, including reliability concerns and bandwidth overheads. All this is something that network operators have to pay attention to. The expected business impact will be large. For example, network operators can use our approach in two different ways. They can feed the reliability model and targets, as well as the communication model, and obtain the lowest the bandwidth overhead that we can obtain while achieving the reliability target. If it turns out to be too high, they can revise the communication model (i.e., different consistency level might result in a lower overhead). Alternatively, they can cap the bandwidth overheads to an amount they can justify and check the kind of reliability that will be achieved. If it is insufficient they can go through an iterative process whereby they can increase the bandwidth until they reach acceptable reliability levels. We make the code for our approach available on (https://github.com/nigsics/dcpmtool).

3. Algorithms for integer valued radio resource management

3.1 Overview

When managing a RAN, there are many integer optimization tasks, where physical resources used in individual cells are considered logical ones, labeled by integers. Conventionally, such logical resources are distributed to cells through operation and maintenance, based on off-line optimization, or applying SON, see e.g. [Bandh 2009]. With evolving 5G systems, such problems become more prominent. There is room for dynamic optimization, and a centralized approach assisted by C3 is foreseen. The four problems of PCCS, PCI, BeamID and TDD switching point assignment, considered here, belong to this category. A discrete degree of freedom is allocated to logical network objects that are related to a spatial location and possibly spatial direction. The discrete degree of freedom should be allocated in such a way that collisions are avoided. Here, we first consider the Beam ID assignment problem, relevant for 5GNR. The work in this section work as published in part in [Ahmed2016]. Next, we solve the PCCS and PCI problems in a centralized setting, by adopting a Genetic Algorithm (GA) for graph coloring, following the approach of [Gwee1993]. A network graph for coloring is constructed on an application running on C3 by centralizing information from the network. Then, the GA solves the PCCS and/or PCI for each BS. Finally, we consider TDD switching point optimization.

PCCS is studied in the context of carrier aggregation in 3GPP [3GPP TR 36.912] [Ahmed 2010]. In carrier aggregation, each Base Station (BS) selects a primary carrier, among all the component carriers available. The goal is to minimize mutual interference by selecting different component carriers for neighboring BSs [Garcia 2009]. This problem is essentially a graph-coloring problem. In the PCI assignment problem, each BS needs a physical cell identification to distinguish itself from others [3GPP TR R3-080376] [Bandh 2009] [Ahmed 2010]. A user relies on the PCI to identify different BSs, for example, during hand-over. Furthermore, to avoid confusion to the user, a BS should not have two neighbors with the same PCI. Thus, the PCI assignment problem becomes a graph-coloring problem on the graph of two-hop neighbors. In the literature, this problem has been addressed in a SON setting, and local search algorithms have been used for evaluation.

In addition to CellIDs, Beam IDs become relevant for mmWave based 5G New Radio (5GNR) networks. There, due to the high numbers of antennas, omnidirectional broadcast transmissions are not possible. As a result, broadcast channels have to be transmitted by base stations from a select set of fixed beams. Thus, in addition to conventional cell-search, where UEs should find the best cell, the UEs have to perform *directional cell search*, to identify the best beam that they can decode from the best cell. The sequence of beams that are used for such broadcast transmissions, constitutes a BeamID in a cell.

TDD systems present the possibility to adapt dynamically the ratio of uplink (UL) and downlink (DL) resources according to the existing traffic, by allocating different number of UL/DL timeslots. The boundary between UL and DL timeslots is defined as “switching point”. It is assumed here that timeslots are arranged in a sub-frame, and the sub-frames are synchronized over the network. Thus a switching from DL to UL happens at the same time in all cells, whereas the time of switching from UL to DL is an integer resource.

3.2 System model and simulation scenario for 5GNR Beam ID assignment

We consider the Beam ID allocation problem which is fundamental in future mmWave networks. Due to the extremely small wavelength at mmWave frequencies, single antenna apertures are small, and large antenna arrays are required to provide beamforming power gain for both cell discovery and data transmission. In contrast to traditional low-frequency cellular networks, mmWave BS discovery signals should be transmitted using phased beamforming, otherwise, there may be a mismatch between the range of a cell where the network is discoverable, and the range where an acceptable service is possible, so that the latter is larger. Thus beam-specific discovery transmissions are needed.

We model a network consisting of N base stations. For transmission of discovery signals, each base station has M fixed analog beams covering the served angular domain. In order to cover the whole cell,

time division multiplexing is used for broadcast transmissions from the beams – in a given broadcast channel resource, one beam, or a subset of beams, is used for broadcast transmission of discovery signals for cell search. Here, the beam ID is defined as the index of time slot during which the beam is transmitted. These beams enable user equipment to discover the cells. User equipment is able to discover a given cell only if the beam from the base station of that particular cell is pointing in its direction, and the interference from own cell beams is under control. The spectrum allocation among base stations is based on reuse-1, thus beams may interfere with each other. Accordingly, it is important for BSs to schedule their beams in a way that not only the whole cell is covered for each BS, but the interference from one BS to the beams of other BSs is minimized. For this, a graph multi-coloring formulation of the problem along with different algorithmic approaches can be used. The graph is constructed on the basis of user measurements by considering interference-to-carrier (I/C) ratios between the strongest beam of potential handover candidate cell, and the beams in own-cell. During network operation, a cell collects historical information from handover measurements of its users during operation, and aggregates this to a local description of the graph underlying directional cell search. A user receives multiple beams with varying powers from each base station which depends on the downlink channel conditions. For a user *k* currently served by BS *i*, its largest beam discovery received power from its serving BS is denoted by $p_{i,b}^{k}$, where *b* is the best discovery beam for UE *k* in its own cell *i*. The potential handover beams in neighboring cells depend on a Handover Margin (HOM) parameter Ω which specifies the handover users. For a neighboring BS *j*, if the largest beam discovery received power $p_{j,p}^{k}$ received by user *k* from BS *j* on the beam *p* satisfies $p_{j,p}^{k} > p_{i,b}^{k}/\Omega$, then user *k* is a potential handover user from BS *i* to *j*. The task of beam coloring is to allocate time slots to the discovery beams in each cell so that the SINR performance for handover measurements is optimized for the handover users, based on instantaneous network state or historical information. We assume that there are *M* colors (number of time slots) for beam assignment. The beam assignment in cell *i* is denoted by a vector $\mathbf{c}_i$ of size *M* ×1. For a set of handover users U_i served currently by BS *i*, the I/C vector for handover measurement is calculated as

$$V(\mathbf{c}_i) = [\frac{p_{i,q}^{k}}{p_{j,p}^{k}}]_{k \in U_i},$$

where beam *q* in BS *i* and beam *p* in BS *j* has the same color. Good beam assignments in multiple cells should make sure that maximum of $V(\mathbf{c}_i)$ should be minimized for all cell *i* (*i* =1, 2… *N*).

The control framework for discovery beam assignment is illustrated in Figure 6. The users are involved in measurement and reporting of potential handover beams to the serving base station. The base station creates a local neighborhood graph on the basis of abstractions of the measurements received from the users. Moreover, it forwards the local graph to the Centralized Controller and Coordinator (C3), which then constructs a network graph. The beam coloring algorithm is run by C3 on the created network graph. A feedback loop exists enabling C3 to control the base stations. Radio Resource Control (RRC) measurements are conducted by UEs and BSs and the local aggregated information is reported to the C3. C3 can also ask base stations and users for measurements and abstractions. This would happen over a connection from C3 to RRC running in the base station. The C3 has a picture of the overall resource situation in the whole network by merging the information reported by UEs and BSs.

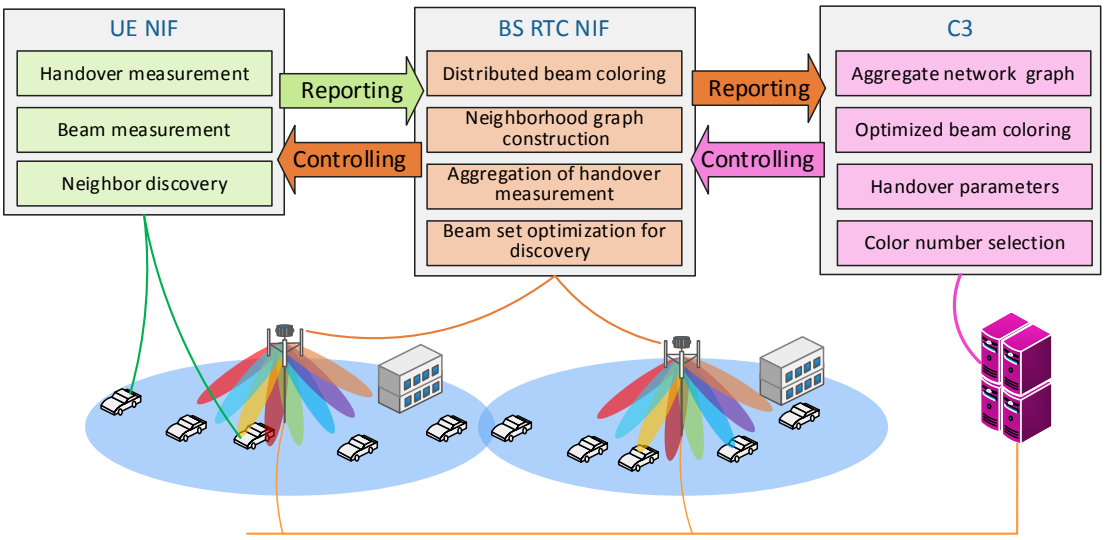


Figure 6 Control framework for discovery beam assignment in multi-cell mmWave network.

The beam assignment problem is a multi-coloring problem. A color combination with T colors has to be chosen for each cell. The task is to schedule the beams in time domain such that the interfering beams are assigned to different time slots, so that I/C of the handover users is reduced. Alternatively, broadcast transmissions to 2 or 4 beams simultaneously can be considered. Here, we use an algorithm based on local search in configuration space, given in Algorithm 2. The algorithm is executed in a centralized manner by the central coordinator. However, at a cost of increased signaling and latency, it is straightforward to formulate its distributed variant, which would be run at the level of base stations.

Algorithm 2: Beam Assignment

Step 1. Cell i using a valid beam schedule $\mathbf{c}$, selects a new schedule $\mathbf{c}_x$ = RandPerm{$\mathbf{c}$} for cell i. Keep the beam schedules for other cells fixed.

Step 2. Find the set of handover UEs U_i which are associated with cell i. For U_i, calculate the I/C vector $V(\mathbf{c}_x)$ and $V(\mathbf{c})$. Compute Δ = max ($V(\mathbf{c})$) - max ($V(\mathbf{c}_x)$).

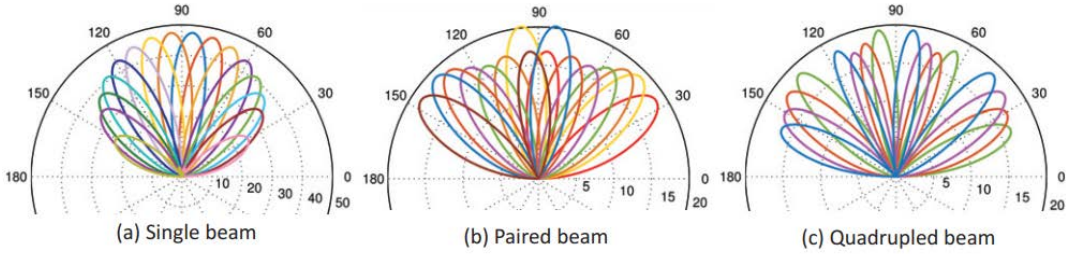
Step 3. If $\Delta > 0$ then $\mathbf{c} \leftarrow \mathbf{c}_x$, else go back to step 1.

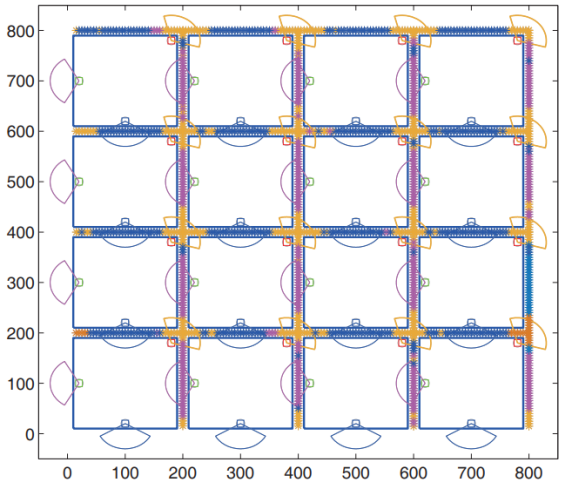
Table 2 - Simulation parameters.

Scenario	Manhattan grid, 800 × 800m
Minimum inter-site distance	100 m
Number of BSs	48
Number of UE positions	Number of UE positions
mmW carrier frequency	28 GHz
LOS pathloss model	61.4 + 20 log10(d)
NLOS pathloss model	72.0 + 30 log10(d)
Shadowing factor standard deviation	8 dB
BS TX power	24 dBm
mmW antenna for BS	8×8 uniform planar array
Beamforming	Analog beamforming, 16 beams

3.3 Results for 5GNR Beam ID assignment

Simulations are carried out using a Manhattan model shown in Figure 7, where the number of base stations N = 48. A summary of the parameters used in simulations is given in Table 2. The historical collection of experiences of users in the network is collected by a uniform sampling of 2560 user positions. Each base station serves a 120 degree wide cell with M = 16 beams. A planar array model is used for beamforming broadcast. Cell discovery broadcast beams are transmitted during a number of time slots assigned by the central coordinator. The number of time slots (beam colors) is assumed to be 16, 8, or 4. A smaller number of time slots for discovery purpose results in less overhead for neighbor cell search. When there are less colors, discovery signals are transmitted simultaneously to multiple directions. The idea is that most of the broadcast channel information, common to the cell, may be transmitted to multiple cells simultaneously. Only a small part of broadcast information would be beam specific. We have designed different kinds of beam patterns for this planar array, as depicted in Figure 7. For a setting of 16 colors, single beams are transmitted during one slot. Paired beams and quadrupled beams are used for settings with 8 colors and 4 colors. The beam assignment algorithm given as Algorithm in Algorithm 2 is used to improve the Signal-to-Interference-plus-Noise (SINR) performance for handover users. Simulation results are shown in Figure 8. The distribution of SINRs of the handover measurements are shown, for users that are in the handover regions, i.e. the received power from a neighboring cell beam is within Ω = 10 dB of the received power of the best serving cell beam. The initial point of each base station is a fixed permutation of beams over time slots. At an update instant, a base station creates a new random permutation as a potential local move, and tests it for the maximum I/C values for handover users. As expected, using 16 colors with 16 directional single beams results in best handover discovery SINR performance. Using less colors results in less overhead in neighbor cell search, but SINR performance will degrade due to smaller beamforming gain and increasing interference by using paired beams or quadrupled beams. It can be seen from the results that in this setting, one iteration is almost always sufficient for the distributed update of color patterns for each cell. Transmissions of broadcast signals to multiple beams simultaneously can be used to reduce the amount of resources needed for cell search, or to reduce the latency in finding neighboring cells. The cost of this is reduced SINR in handover measurements, as there are more beams transmitting broadcast signals simultaneously with the potential handover beam in a neighbor cell, and the broadcast signals have smaller coverage. This leads to an increased number of Radio Link Failures (RLFs).





(d) Manhattan scenario

Figure 7- Beams for planar array and the Manhattan scenario with mmWave base stations.

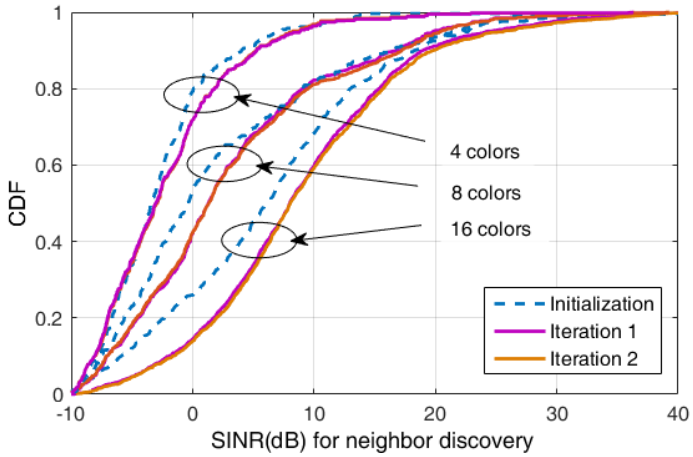


Figure 8 Handover SINR performance for cell search with different number of beam colors. The number of colors refers to the number of time slots used for beam discovery transmissions. Base stations in a Manhattan grid have 8x8 arrays constructing 16 analog beams.

3.4 System model and simulation scenario for PCCS and PCI

We perform extensive network simulations in a cellular communications network consisting of a total of 972 cells, of which 108 are macro cells and 864 are small cells. The scenario consists of a grid of macro BSs with three macro cells (sectors) per BS site. Each macro cell operates in a SISO mode with antennas in each sector. Small cells are uniformly distributed within the macro cells coverage area, adopting the model of [3GPP TS 36.872], assuming two clusters per macro cell, and four small cells per cluster. Simulations are executed in 2000 network instances. In each simulation instance, BSs and users are distributed as indicated in Table 3 shows the layout of an example network instance.

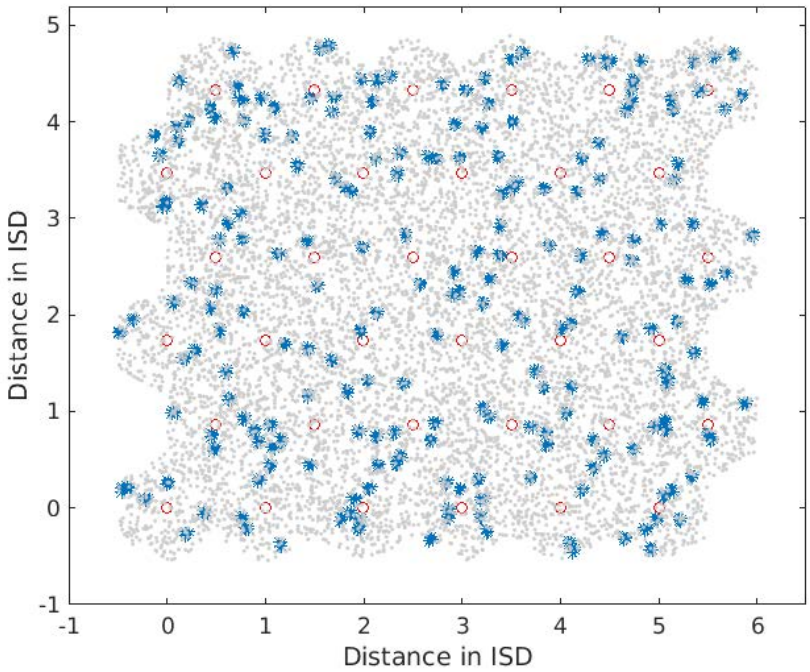


Figure 9 Example of network instance depicting 36 macro cells sites (o), 864 small cells (*), arranged in clusters, and 9720 users (.)

Table 3 System simulator parameters

Parameter	Description
Macro BS Inter-Site Distance (ISD)	*ISD* = 1732 m.
Macro cell network	Grid of 6 x 6 sites separated by *ISD*, with 3 cells per site.
Number of macro cells	M = 108
Small cells distribution	C = 2 cluster uniformly distributed in each sector of macro BS. S_c = 4 small cells uniformly distributed in each cluster.
Number of small cells	S = 864 (M x C x S_c)
Number of users per macro BS	U_c = 90
Total number of users	9720 (U_c x M)
User distribution	p x U_c , uniformly in macro cell, (1-p) x U_c uniformly distributed in small cell of radius 50 m, with p uniform in [0,1].
Macro (small) cell transmit power	43 (30) dBm
Macro BS inter-site dist	9 dB
Thermal noise level	-174 dBm/Hz
Path loss model macro (small) cell	Uma (UMi) from [3GPP TS 36.814] Table B.1.2.1-1
Shadow fading	Standard deviation 8 dB
Synchronization threshold	-7 dB
Number of network instances in simulation	2000
Initial network situation	10 small cells are turned off

3.4.1 PCCS / PCI problem process flow

The PCCS and PCI problems are solved by centralizing information in the C3. The steps to follow to determine the PCCS or PCI are as follows:

1. Each User Equipment (UE) reports to the serving BS, the BSs with which it can synchronize, determined by a synchronization threshold (Table 2).
2. Each BS reports to the C3 the list of BSs gathered by UEs' reports.
3. The C3 constructs an adjacency matrix between BS, for the cases of PCCS (1 hop), or PCI (2 hops).
4. The adjacency matrix is used to construct the network graph for coloring.
5. A GA is executed with the network graph, to solve the PCCS or PCI problem.
6. C3 communicates the solution (PCCS or PCI) to each BS.

3.4.2 Genetic algorithm for primary carrier allocation and Cell ID assignment

The GA takes as input the network graph for coloring, and returns as output the PCCS or PCI for each BS. Figure 10 GA flow (adapted from [Kozel 2015]) depicts the flow of a generic GA in the context of the PCCS and PCI problems. In GA, the state develops as a collection of alleles, and related mutations. The GA is implemented with a cost function that computes the number of conflict-nodes with the same color, where a color is the allele for the problem under consideration (PCCS, or PCI). As shown in the figure, the GA iterates creating new generations of the population. The offspring selection criterion is based on selecting the genes that reduce the cost (i.e. minimize the conflicts). Additional parameters for the GA are listed in Table 2.

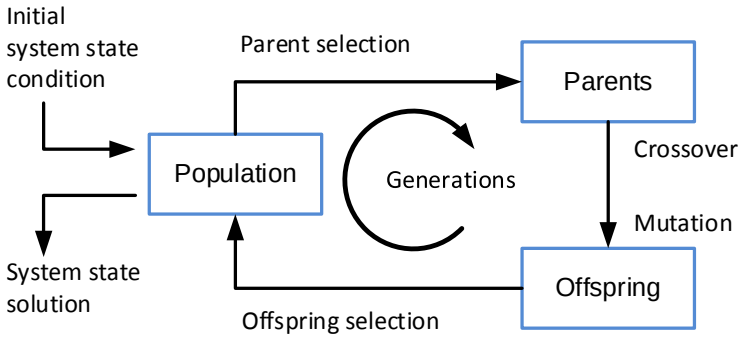


Figure 10 GA flow (adapted from [Kozel 2015])

The performance of the algorithm is evaluated by its capability to solve conflicts when new cells appear in the network. Thus, we study the case in the network described in the system model above, where 10 small cells, out of 864, are turned off in the initial state. These cells are selected randomly with a uniform distribution. Then, two cases are analyzed, where 1 and 10 of these small cells are activated (turned on) at once. The presence of these new cells creates conflicts in the graph. The GA is used to solve these conflicts, and return a new solution.

First, a conflict free solution for the initial state is found, by executing the GA with the active cells, namely 108 macro cells, and 854 small cells. In this case, the same color (i.e. carrier ID or PCI) is assigned to all the cells. The solution to this initial state is then used as starting point when activating the cells that were turned-off.

Table 4 Genetic algorithm parameters

Parameter	Description
Alleles	Carrier IDs / PCIs
Population size	128
Selection	Rank weighting
Crossover	Single-point crossover
Mutation rate	1%
Population kept ratio	30%
Max. number of generations	1000
Number of carriers evaluated	4 to 12
Number of PCIs evaluated	40 to 60

3.5 Results for primary carrier allocation

In this section, we evaluate the performance of the GA algorithm for Primary Component Carrier Selection. We aim to distribute a small number of carrier IDs as efficiently as possible to each BS in a conflict-free manner. The resulting distribution of the number of neighbors per cell considering 1-hop neighbors, is shown in Figure 11, collected in 2000 network instances.

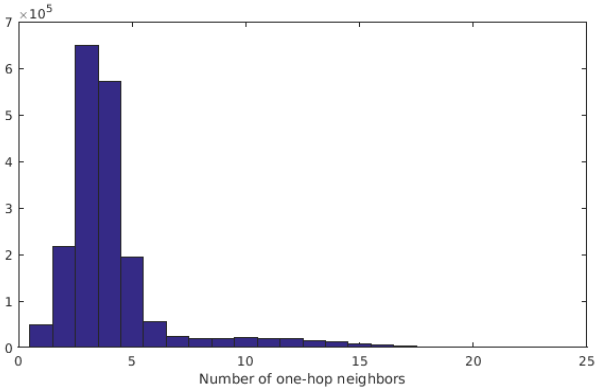


Figure 11 Distribution of the number of neighbors.

The performance metrics to be evaluated are 1) the probability of convergence, and 2) number of generations, averaged over all the network instances. The GA is executed until convergence, or until 1000 generations are reached. As initial network state, we assign the same carrier ID to all the active BSs, and obtain the statistics of the performance metrics, for different number of carrier-IDs.

Then, from the conflict-free solution found for the initial state, or from state reached after 1000 generations, we execute the GA again for the cases of activating 1 and 10 small cells at once. The convergence probability, and mean number of generations are shown in Figure 12 and Figure 13, respectively. We observe that when activating 1 and 10 cells, it is necessary a minimum of 12 carrier IDs for 100% convergence. For the case of 12 carrier IDs, when activating 1 cell, it is necessary an average of 20 generations to find a conflict-free solution. In contrast, when activating 10 cells, it is necessary an average of around 120 generations to find a conflict-free solution. The initial case with all the cells taking the same carrier ID, can be used as an upper bound in performance. In this case 13 carrier IDs are necessary, and less than 200 generations to converge to a conflict-free solution. To get an understanding on the computational cost, the average execution time for one generation with 13 colors is 0.04 s on one processor on a conventional laptop with Intel i3 processors.

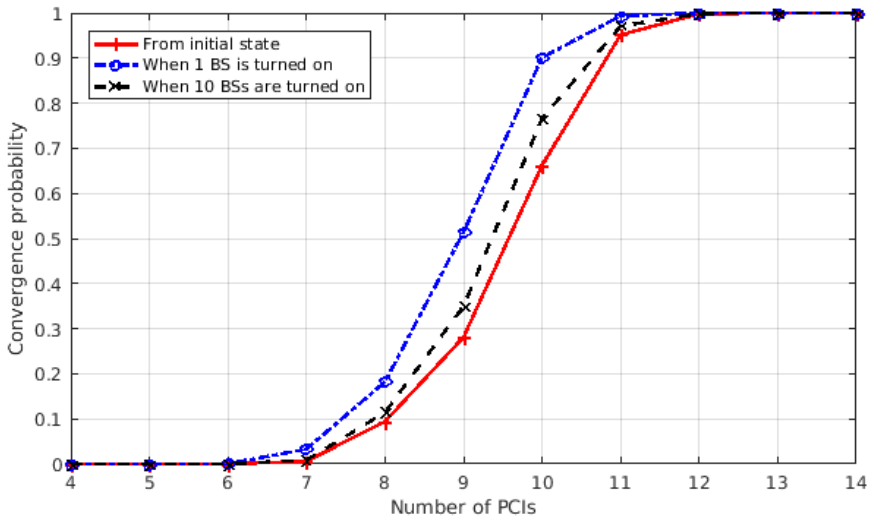


Figure 12 Convergence probability against number of carrier IDs.

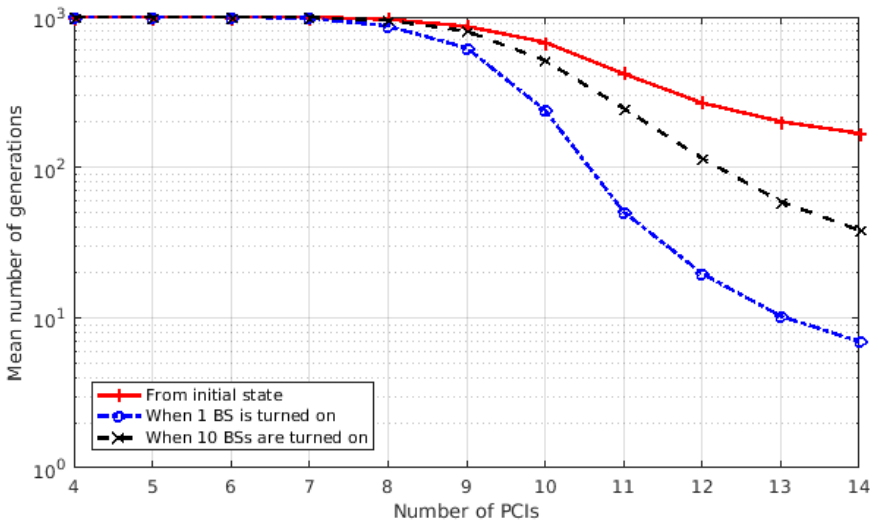


Figure 13 Mean number of generations against number of carrier IDs.

3.6 Results for PCI assignment

In this section, we evaluate the performance of the GA algorithm for PCI assignment. We aim to find the number of PCIs that are sufficient to allow newly entering cells to configure their PCI in a conflict-free manner. The resulting distribution of the number of neighbors per cell, considering 2-hop neighbors, is shown in Figure 14, collected in 2000 network instances.

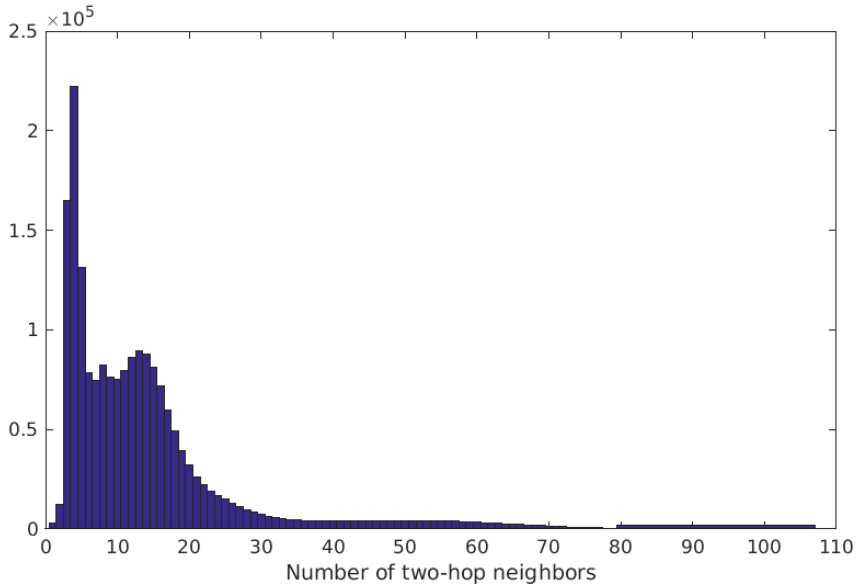


Figure 14 Distribution of the number of two-hop neighbors.

The performance metrics to be evaluated are 1) the probability of convergence, and 2) number of generations, averaged over all the network instances. The GA is executed until convergence, or until 1000 generations are reached. As initial network state, we assign the same PCI to all the active BSs, and obtain the statistics of the performance metrics, for different number of PCIs.

Then, from the conflict-free solution found for the initial state, or from state reached after 1000 generations, we execute the GA again for the cases of activating 1 and 10 small cells at once. The convergence probability, and mean number of generations are shown in Figure 15 and Figure 16, respectively. We observe that when activating 1 and 10 cells, it is necessary a minimum of 53 and 54

PCIs, respectively, for 100% convergence. When activating 1 cell with 53 PCIs, on average 20 generations are needed to find a conflict-free solution. In contrast, when activating 10 cells at once, with 54 PCIs, on average around 70 generations are needed to find a conflict-free solution. The initial case with all the cells taking the same PCI, can be used as an upper bound in performance. In this case, 58 PCIs are necessary, and less than 200 generations are needed to converge to a conflict-free solution. To get an understanding on the computational cost, the average execution time for one generation with 58 colors is 0.06 s on one processor on a conventional laptop with Intel i3 processors.

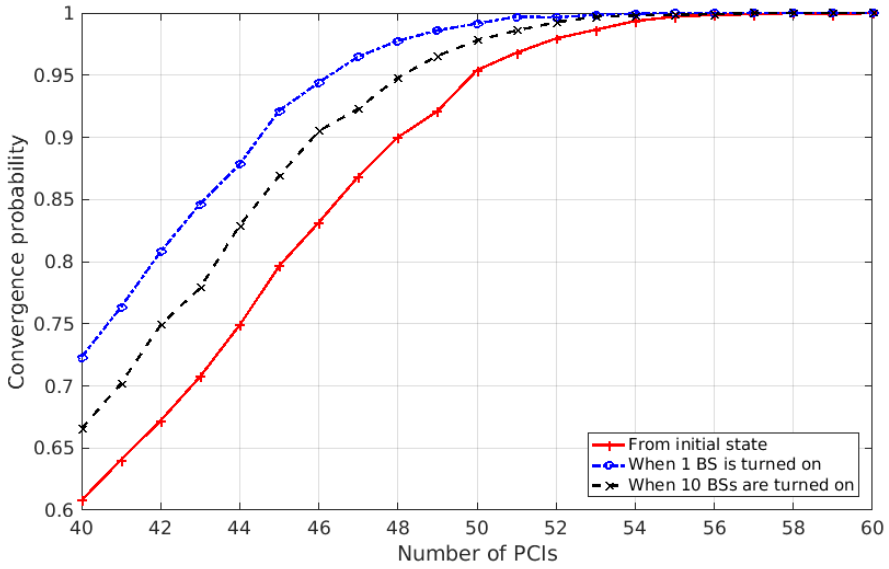


Figure 15 Convergence probability against number of PCIs.

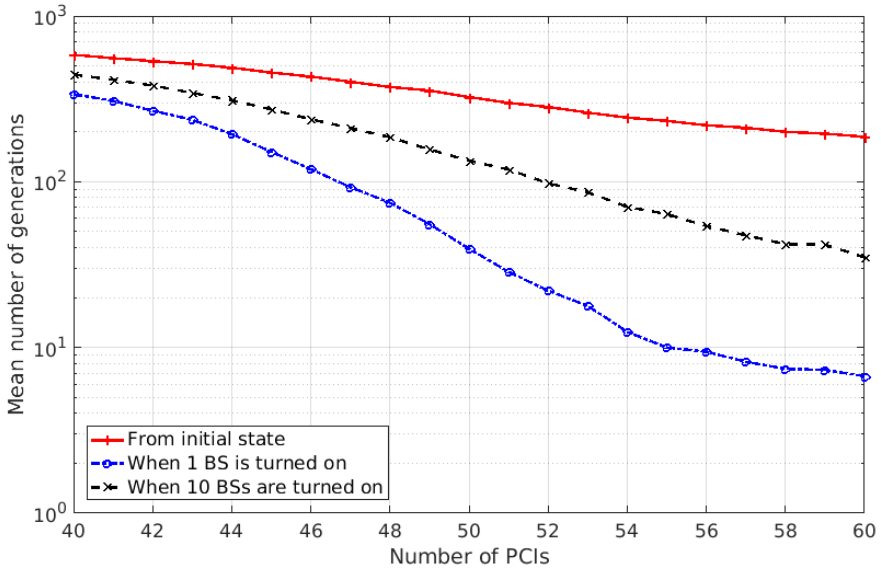


Figure 16 Mean number of generations against number of PCIs.

3.7 System model and simulation scenario for TDD switching point assignment

Here, we analyze two algorithms for changing dynamically the switching points of base stations in a cellular network, where there is an overall synchronization of (sub-)frames. Accordingly, we model the TDD switching point as an integer resource. We consider dynamic optimization of the switching point.

When the switching points are dynamic, two or more links belonging to different base stations may have switching points at different positions. In this case, some timeslots are allocated for UL and DL at the same time instant for these links. UL and DL timeslots overlapping at the same time instant give origin to the so-called crossed-slots.

Figure 17 (a) shows an example of a scenario with two base stations (BS) operating with different switching points. BS 1 is receiving in UL, and BS 2 transmitting in DL, causing TDD crossed slots. Part (b) of the figure shows an example of the arrangement of UL/DL timeslots, and the position of switching points, in a sub-frame.

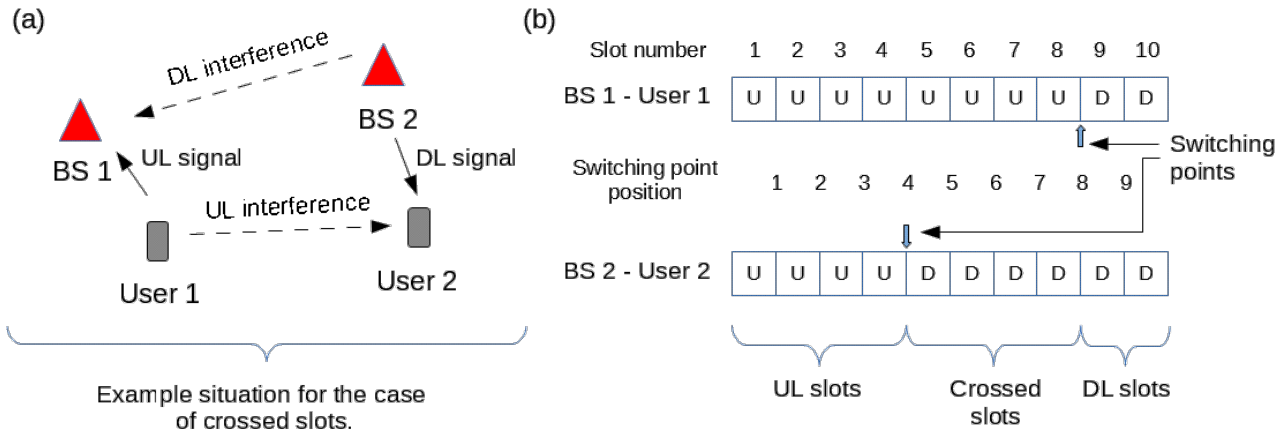


Figure 17 - Example showing (a) scenario with two base stations (BS), one operating in UL, and the other in DL, causing TDD crossed slots, and (b) arrangement of UL/DL timeslots and position of switching points.

In a given network, with a fixed geography and positions of cells, the level of interference in crossed and non-crossed slots depends on the network instance, i.e. the precise location of the active users in the network. In dynamic optimization, the optimal switching point is selected for a given network instance, reflecting the crossed-slot and conventional interference situation in the instance.

Different applications and services available in wireless devices may require a symmetric UL/DL rate, whereas others may require an asymmetric traffic. Therefore, in simulations, we assume two types of users, with different traffic requirements:

A) Users requiring only DL traffic (referred to as "DL-traffic-users").
B) Users requiring as the same data rate in UL and DL (referred to as "*Dual-traffic-users*").

Disregarding minor UL load caused by control signaling, DL-traffic-users require only DL traffic, however it is assumed that the first timeslot is always reserved for UL, which generates UL interference, but the transmission is not part of the rate calculations. Timeslots to the left of the switching point contain no transmission, except for the first one transmitting in UL. Figure 18 (A) shows an example for this case. In this example, the switching point in position 5, therefore timeslots 2 to 5 have no transmission. An effective rate is defined for a user u of this type as follows,

$$\textit{Effective rate of DL-traffic-user} = r_{DL,u} = \textit{expected DL rate}$$

Since base stations and users are assumed with the same transmission power, in ideal and isolated conditions, dual-traffic-users would have the switching point dividing the sub-frame in the middle, thus allocating 50%/50% of timeslots for UL/DL traffic. However, in non-isolated cells, the number of timeslots for UL and DL may vary as consequence of surrounding interferences. An example of UL/DL distribution for these users is shown in Figure 18 (B). These users use always all the timeslots. An effective rate is defined for a user u of this type as follows,

$$\textit{Effective rate of dual-traffic-user} = r_{Dual,u} = 2 * \min(\textit{expected UL rate}, \textit{expected DL rate}).$$

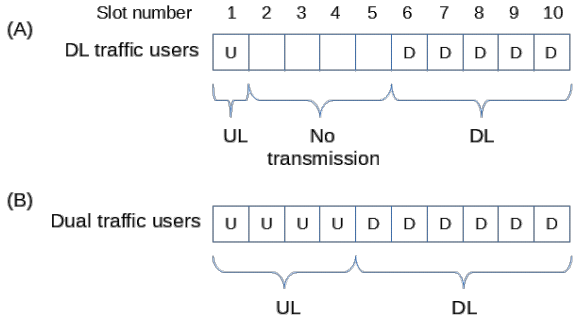


Figure 18 - Example of UL/DL distribution in timeslots for a A) DL traffic user, and B) Dual traffic user.

Aiming to a proportionally fair distribution of resources, a system sum utility is defined as the sum of the logarithm of the expected rates:

$$System\ sum\ utility = \Sigma_u \log(r_{DL,u}) + \Sigma_u \log(r_{Dual,u})$$

where the rates are calculated from the SINR experienced, according to the interference situation for each timeslot.

3.7.1 Switching point allocation schemes

In our simulation analysis we consider 3 switching point allocation schemes, namely:

1) Fixed switching point
2) Dynamic switching point per cell, per network instance, using a Simulated Annealing (SA) algorithm.
3) Dynamic switching point per cell, per network instance, using a Genetic Algorithm (GA).

In "Fixed switching point" the same fixed position is used for the switching point in any network instance, for all the base stations. The optimum fixed switching point of the system is obtained by calculating the mean sum utility over several network instances, for each timeslot. The timeslot with maximum mean sum utility determines the optimum position for the fixed switching point.

The SA algorithm is applied to maximize the system sum utility. Parameters for the SA algorithm are listed in Table 5. Algorithm 6 shows the pseudocode for the neighbor-function used in the SA algorithm. In the neighbor-function, a move is valid if the position of the switching point does not decrease more than the initial position, or does not increase more than the last position.

Table 5 - Simulated annealing algorithm parameters

Parameter	Description
Number of iterations (maximum temperature)	1000
Acceptance probability, p	if new state > previous state, $p = 1$ otherwise, $p = e^{(new\ state\ -\ previous\ state)\ /\ T}$ where T is the temperature (current iteration index)

Algorithm 6 - Neighbour function for simulated annealing (Table 6).

```
b ← Select base station with a uniform distribution
s ← Select switching point move from the set {-3, -1, +1, +3} with a uniform distribution
if the move is valid for the switching point of base station b
    Apply move +s to the switching point of base station b
end
```

Furthermore, GA is applied to maximize the system sum utility. The GA iterates creating new generations of a population. The offspring selection criterion is based on selecting the genes that reduce the cost. Parameters for the GA are listed in Table 7.

The SA algorithm and GA take as input the switching points, and rates experienced in each timeslot, of each base station, and return as output the optimal position of the switching points that maximize the system sum utility. The algorithms run in the COHERENT C3, with centralized knowledge of the rates. To this end, the C3 requires the collection of received powers of strong interferers of each node in the network.

Table 7 - Genetic algorithm parameters

Parameter	Description
Alleles	Number of switching point positions
Population size	20
Selection	Rank weighting
Crossover	Single-point crossover
Mutation rate	20%
Population kept ratio	30%
Max. number of generations	50

3.7.2 Path-loss model

Dynamic optimization of TDD switching points is beneficial when one has multiple small cells with strong interference couplings. In small cells, the load varies rapidly, so that the cell-specifc need for UL/DL resources varies in time, and from cell to cell. When the interference between cells is large, which may happen in uncoordinated deployments, interference in crossed slots is expected to be most harmful. Accordingly, we investigate switching point optimization in an indoor small-cell scenario where users have different traffic requirements. In such a scenario, both high load variations may happen, and where interference between cells may be high.

Resource allocation is handled by maximizing a system level sum utility, in which the utility of each user is the logarithm of the data rate experienced by the user in the allocated timeslots (proportional-fair utility). The switching points in the system are selected to maximize the system sum utility. The performance of a dynamic switching point system is compared to a system where there is a permanent fixed switching point.

The simulations are performed by Monte Carlo methods in 500 network instances. In each instance, we distribute randomly users in the simulation scenario as described below. Then calculate the optimum switching points with the SA algorithm and GA, aiming to maximize the system sum utility. At the end of the simulation, statistics of the system sum utility and switching points are collected.

The simulation scenario and path loss model are based on the Winner indoor scenario A1 [Winner]. A building with three-floors is assumed, with Winner A1 layout in each floor, and floor height of 3 m. The layout in each floor consists of 40 rooms and 2 corridors covering an area of 50m x 100m. The size of each room is 10m x 10m. The corridors have a length of 100m and width of 5m. Inter-cell interference is particularly strong along the corridors. Figure 19 depicts the building and Winner A1 layout. In each floor, there are four base stations at fixed positions inside rooms, located at the coordinates (25.0, 5.0), (75.0, 5.0), (25.0, 45.0), (75.0, 45.0) [meters], respectively. Users are uniformly distributed per floor, and only inside rooms. From the wall boundaries in A1 layout, we consider a forbidden zone of 10 cm where a UE cannot be located, thus representing the thickness of the wall in the model. We assume that only one user is scheduled per sub-frame, at a given network instance. Users are located randomly, such that by cell selection there is only one user served per base station.

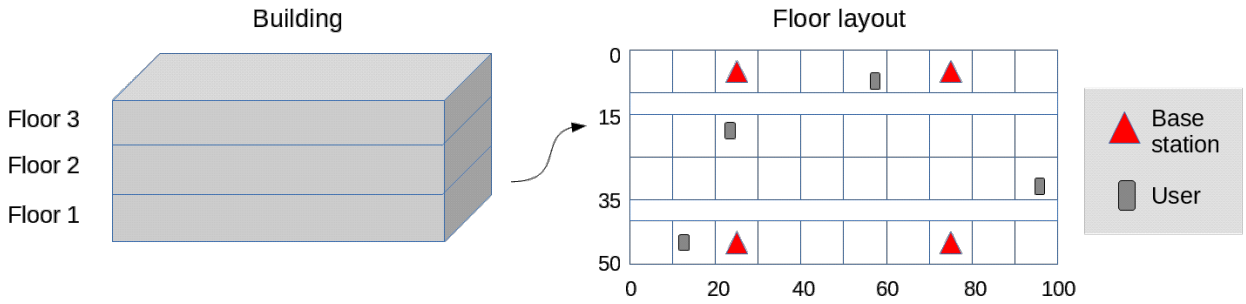


Figure 19 - Building with 3 floors and Winner A1 layout. Base stations are at fixed positions. Location of the users from a particular network instance.

The serving base station of a user is the one with the maximum received power. Accordingly, cell selection finds the base station providing best DL SINR. The received power is calculated considering path loss, shadow fading and fast fading. The simulator starts with a considerable number of users per floor, and verifies whether it is possible to allocate one served user for each base station. Users without a main link are discarded. In each network instance, a user has a 50 % probability to become a DL-traffic-user or Dual-traffic-user. All possible combinations of links between users and users, users and base stations, and base stations and base stations are modelled. Table 8 summarizes simulation details and parameters.

Table 8 - Simulation details and parameters

Parameter	Description
Interference modelling	All links modelled user ←→ base station user ←→ user base station ←→ base station
Duplexing	TDD, 10 timeslots
Carrier frequency	2 GHz
Thermal noise level	-174 dBm/Hz
Path loss model	Winner A1 [Winner] Except with fixed shadow fading std. dev. 4 dB
Shadow fading correlation	0.5
Fast fading	Spatial Channel Model [3GPP TR 25.996]
UL power control	No power control.
Modulation	QPSK – 16QAM – 64QAM (Gray mapping)
Coding	Turbo code with generators G1=1011 (feedback), G2 = 1101. Random interleaver. Block length 256 symbols.
MCS (The MCS used is the MCS that gives the maximum rate)	QPSK rates: 2/3 8/11 4/5 8/9 1 8/7 4/3 16QAM rates: 16/11 8/5 16/9 2 16/7 8/3 16/5 64QAM rates: 24/7 4 24/5
Link to system mapping	Effective Exponential SINR Mapping
Base stations distribution	4 per floor, in 3 floors. Winner A1scenario [Winner]
Base station transmit Power	21 dBm
Base station height	2 m. above the floor level
Base station antenna configuration	Not modelled, assumed omnidirectional
Base station number of antennas	1
Base station receiver noise figure	6 dB
Users distribution	Uniformly per floor, in rooms.
User transmit Power	21 dBm
User height	1 m. above the floor level
User antenna configuration	Not modelled, assumed omnidirectional
User number of antennas	1

User receiver noise figure	6 dB
Number of network instances	500

The position of a switching point determines the number of timeslots for UL and DL. When a user is receiving information from its serving base station (on DL timeslots), it may experience interference from other base stations with DL transmission and from other users with UL transmission. When a base station is receiving information from its served user (on UL timeslots), it may experience interference from other users with UL transmission and from other base stations with DL transmission. In a system with base stations and users adopting a dynamic switching point scheme, and with the proposed traffic types, it is expected that the SINR level will vary in different timeslots. For a given SINR level each receiver feeds back a Channel Quality Indicator (CQI) which the transmitter uses to select a Modulation and Coding Scheme (MCS) that maximizes the rate per timeslot. Then, for a given switching point position, we are able to know the rate experienced in UL and DL for each user. By moving the switching point position, we can vary the relative rate of UL and DL.

In the system under consideration, the goal of DL-traffic-users is to get as much DL traffic as possible; these users will try to move the switching point such that their rate increases, by increasing the number of DL timeslots. In the other hand, dual-traffic-users will try to get as much UL as DL by moving the switching point to a position that balances the UL and DL rate. A proportional fair utility in the system is considered, to avoid situations such as, users with good conditions taking most of the timeslots, and users with bad conditions getting a minimal amount or no timeslots at all. An application running on C3 gather aggregated UL & DL CQI information from the network, and calculated the proportional fair utility. This is performed on a short time scale, tentatively a radio frame of 10 ms. In the simulation model, this modeled for each network instance, i.e. a specific set of active users at fixed locations. The utility is calculated for the different switching point schemes. The optimization objective is to maximize the system sum utility.

In each network instance, first we determine the position of the switching points, common to all base stations, that give the maximum system utility. This solution is used as starting point in the search for a dynamic switching point per cell, with the two algorithms (i.e. switching point allocation schemes 2 and 3). At the end of the simulation, the optimal fixed switching point (i.e. switching point allocation scheme 1) for the system is calculated, from stored system sum utilities for all the instances and switching point positions.

3.8 Results for TDD switching point assignment

The performance of the system is evaluated by looking at the Cumulative Distribution Functions (CDF) of system sum utilities for the different switching point allocation schemes. Figure 20 shows the CDF of the system sum utility in all network instances, for the three switching point schemes. The switching points calculated with the GA produce a result that outperforms the result obtained with the SA algorithm. Both SA and GA results are better than the fixed switching point scheme. For the fixed switching point scheme, we found that the switching point at position 4 is the one that maximized the system sum utility. The results show that in terms of system sum utility, the system has a better performance by allowing a dynamic switching point.

In the SA algorithm and GA, we made comparable the number of cost function evaluations. In the SA algorithm, the utility function is evaluated 1000 times, once per iteration. In the GA , the cost function is evaluated once per generation in a population of 20 genes. The GA iterates in 50 generations, resulting in 1000 cost function evaluations, analogous in number to the SA algorithm. In these terms, both algorithms have a comparable computing cost, however, the GA has the additional cost of sorting the population once per generation. Accordingly, we can argue that this additional cost is the one needed to observe higher gains with the GA.

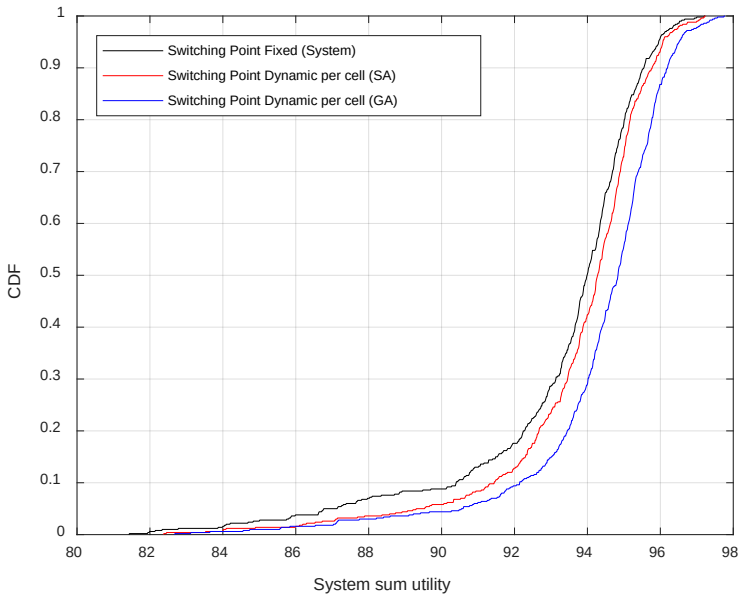


Figure 20 - CDF of system sum utility.

Figure 21 and Figure 22 show the Probability Mass Function (PMF) of switching point positions for the SA algorithm and GA. We observe that the SA algorithm has a peak at the position 4, which was observed to be the most frequent optimal solution in all the network instances. The GA started with the same initial arrangement of switching point than in the SA algorithm. However, in contrast of the SA algorithm, the GA finds a more spread distribution of positions, that produce a better system utility. When comparing this result to the PMF of the SA algorithm, we can infer that the neighbor function in the SA algorithm, and number of iterations, was not efficient to explore other regions. However, the result merely exposes the trade-off between a simple SA algorithm with low storage requirements, to a more complex GA algorithm, requiring more storage to store partial results from the population.

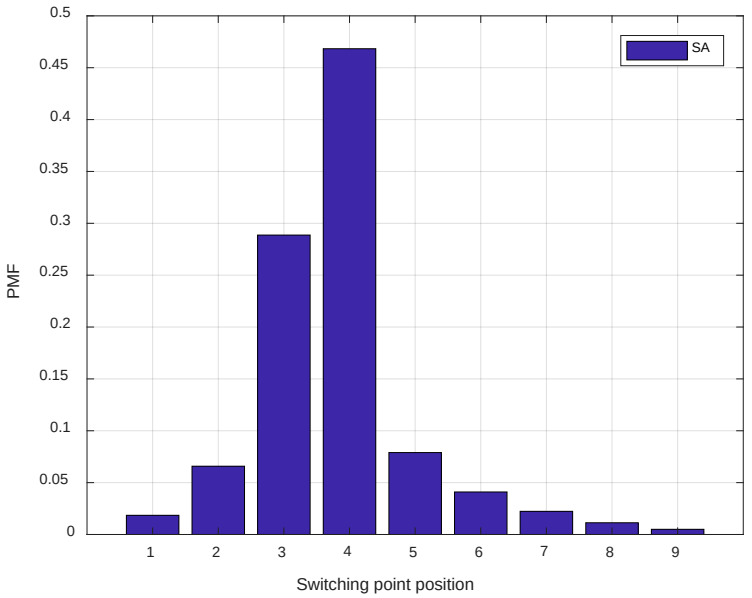


Figure 21 - PMF of switching point positions for the SA algorithm.

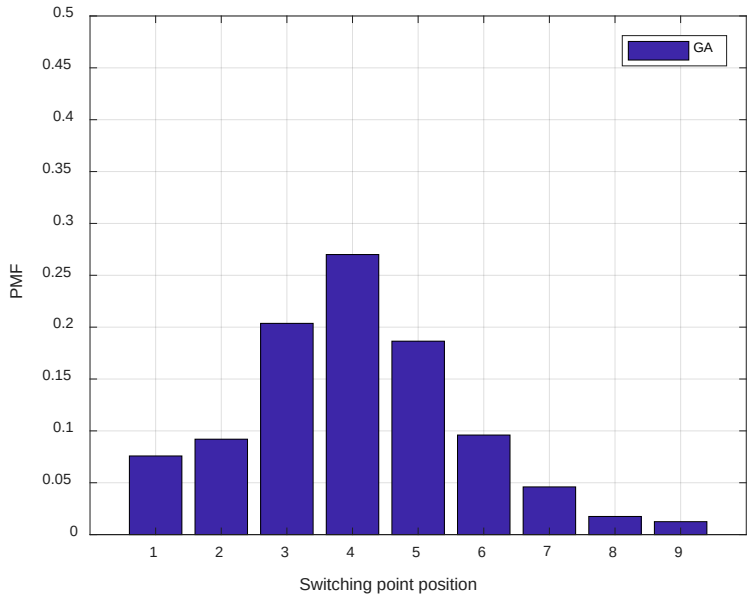


Figure 22 - PMF of switching point positions for the GA.

3.9 Conclusion

3.9.1 Technical/Scientific Impact

In this section, application of centralized algorithms in problems where multi-cell management of discrete-valued radio resource management is performed. The first problem is related to Beam ID allocation in 5GNR networks. This is an important problem for mobility management in 5GNR. First solutions based on centralizing hand-over measurements to C3, and performing a local search algorithm, are provided. The research performed in COHERENT identifies the problem, provides an architectural view and first results based on a set of fixed beams. There is significant potential for future work in this domain. The second and third problem address cell-specific identifiers, related to a primary carrier, and physical cell IDs. These problems have been widely investigated in the SON setting. The fourth problem is related to dynamically changing the TDD switching point. The switching points can be selected with the objective to maximize a system sum utility. Here, a centralized view is taken for these problems, enabling the use of developed search algorithms. Genetic algorithms are applied, as well as simulated annealing. These problems can be seen as test instances for C3 protocols, and algorithms. Genetic algorithms are found to be promising candidates for these problems, and are readily implementable in C3.

3.9.2 Feedback toward development

The discrete resource allocation techniques presented in this section work upon the abstracted network graphs which do not depend on the details of the mobile network implementation. The network graphs are constructed based on semi-dynamic network statistics which can be retrieved based on currently available network measurements such as CQI. The C3 algorithms can be developed using general virtual machine running on different hardware platforms. All of the algorithms discussed here can be implemented on top of measurements which are available in 3GPP networks today, will be available in 5GNR, and can be implemented in Wi-Fi networks. For TDD switching point assignment, two centralized metaheuristic algorithms were applied. The SA algorithm is a simple solution, requiring limited storage, whereas the GA produces a better solution at the expense of requiring more storage, implementation complexity and an additional computing cost over the SA algorithm to sort the population in each generation. Nevertheless, these additional costs may not be significant when considering the general complexity, at least in the considered TDD switching point problem.

3.9.3 Expected business impact

BeamID optimization, as discussed here, will be an important component of mmWave 5GNR networks, when they are implemented. The expected business impact will be large. The BeamID, PCCS and PCI optimization methods considered here will improve handover reliability, and the business impact will arise from increased reliability of networks, and less OPEX due to RLFs. The results on dynamic TDD switching point assignment show rather modest gains. This indicates that this particular technology component may have limited business impact.

4. Algorithms for End-to-End Wi-Fi RAN Slicing

4.1 Overview

Applications have different performance requirements in terms of bandwidth, latency, and data rate. This calls for a service–oriented approach to network resource provisioning. For example, ITU identifies three classes of services, namely enhanced mobile broadband, massive machine type communications, and ultra–reliable and low latency communications [ITU2015]. While the ITU white paper specifically addresses 5G mobile networks, it acknowledges that such networks will heavily rely on Wi–Fi as traffic offloading solution. As a matter of fact, at the current rate it is expected that Wi–Fi and mobile traffic will account for 63% of the entire IP traffic by 2021 [Cisco2017]. For Wi–Fi segment alone, a single rigid network architecture will not be sufficient to support the diverse set of services and applications that will characterize future 5G systems. Conversely, to make the service–oriented 5G vision a reality it is mandatory to be able to abstract the physical network into multiple end–to–end virtual logical networks or slices, one for each service category. Software defined networking (SDN) and network function virtualization (NFV) are considered to be two of the most promising enablers for achieving this vision. However, while a number of proposals have been put forward for the mobile core [Santos2016], [Basta2017], [Banerjee2015], [Jin2013] the concept is still at its infancy in the radio access segment.

Here, we present a programmable end–to–end network slicing framework for enterprise WLANs. This framework pursues three objectives: (i) programmability, we want to allow network administrators to specify how a precise portion of the flowspace shall be treated in the wireless access segment; (ii) isolation, we want to make sure that slices are kept isolated both from the logical and the performance standpoints; and (iii) customization, we want to allow each slice to specify its own traffic prioritization policies, e.g. in terms of aggregation, rate selection, etc. While the IEEE 802.11e amendment [IEEE2005] did introduce the concept of traffic differentiation and prioritized access, it did not provide any programming abstraction allowing end–to–end slice management. Similarly, the de-facto standard for SDN in wired networks, i.e. OpenFlow [Kckeown2008], does not extend to the wireless access segment. Taking this into consideration, the contribution presented here is threefold.

1. First, we extend the OpenFlow match rule with some fields from the IEEE 802.11 header.
2. Second, based on this extended match rule, we introduce a new programming abstraction named Traffic Rule enabling the specification of customized slicing policies for a precise portion of the flowspace.
3. Finally, we implement a flexible hypervisor capable of ensuring the required logical and performance isolation between slices while at the same time enabling slice customization and efficient radio resource utilization.

The proposed system has been implemented and tested on top of the COHERENT C3 and in particular on top of its reference implementation 5G–EmPOWER [Riggio2015]. An experimental evaluation encompassing a wide range of usage scenarios and conducted over a real–world testbed has demonstrated the capability of proposed approach to meet the design requirements. The entire implementation has been released under a permissive APACHE 2.0 license for academic use as part of the COHERENT SDK[2].

4.2 End-to-end Wi-Fi RAN Slicing

Supporting the requirements of current services and applications can often result in far–reaching changes in both the network architecture and in the protocol stacks. The emerging SDN paradigm aims to break free of this constraint by decoupling the data–plane and control–plane. The network intelligence is then shifted from the network devices to a central location (the network controller) allowing to implement sophisticated traffic management policies based on the global network view exposed by the controller, while the devices simply apply the rules defined at the control plane. The proposed framework extends the mainstream SDN network slicing concept to the wireless access segment in the specific case of 802.11–based WLANs. The proposed framework aims at ensuring efficient sharing of the same physical infrastructure by different services and applications. More

[2] http://empower.create-net.org/

specifically, we assume that an infrastructure provider owns the physical Wi–Fi Access Points (APs) and network switches which are in time leased to the service providers (the slice owners). The focus here is on Wi–Fi–based networks but the design principles presented in this chapter are quite general and can be easily extended to other radio access technologies. Our design builds upon a programmable hypervisor sitting on top of the standard Linux Wi–Fi stack. The hypervisor is in charge of creating, monitoring, and managing the various network slices ensuring performance isolation and efficient radio resource utilization. The high–level architecture of the hypervisor is depicted in Figure 23. As depicted, each AP can support a variable number of slices. In time each of these slices contains one aggregation buffer for each Wi–Fi client in the network. Each slice can have its own EDCA parameters. For example, one slice can use no aggregation and voice–optimized EDCA parameters, while another slice can enable frame aggregation and use background traffic EDCA parameters.

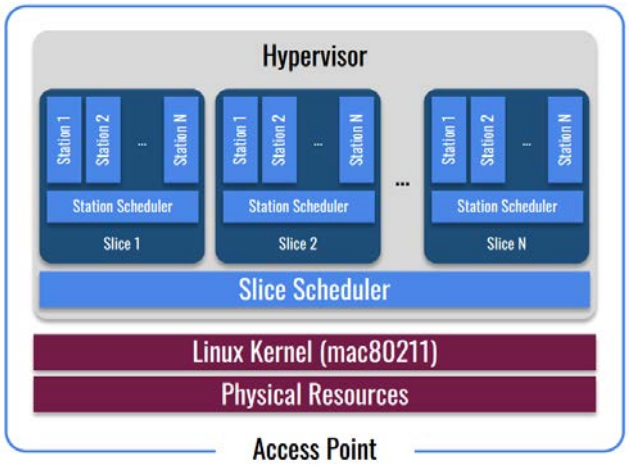


Figure 23. The Hypervisor architecture

4.2.1 The *Traffic Rule* Abstraction

A network slice is defined as a set of radio resources that are assigned to a particular flow. In this way, a network slice can be simultaneously shared among multiple services and applications, and a client can use different network slices simultaneously. The framework introduced in this work provides the ability to create programmable network slices. To this end, a new abstraction named *Traffic Rule* is introduced to map a specific portion of the flowspace to a particular scheduling discipline. The *Traffic Rule* abstraction defines a set of parameters the AP must use when forwarding traffic belonging to a particular slice. Such parameters include:

- EDCA. The EDCA parameters to be used for this slice. This includes Congestion Window, AIFSN, TXOPs, etc.
- Aggregation. The type of frame aggregation to be used for this slice, including A–MSDU, A–MPDU, or none.
- Quantum. The fraction of airtime that can be assigned to this slice in each round.

Any of the parameters of a *Traffic Rule* can be modified at runtime by the controller. In addition, the *Traffic Rule* abstraction can be combined with the *Transmission Policy* abstraction [Coronado2017] which allows the SDN controller to specify the range of parameters the AP can use for its communication with a wireless client. Such parameters include:

- MCSes. The set of Modulation and Coding Schemes (MCSes) that can be used by the rate selection algorithm.
- RTS/CTS Threshold. The frame length above which the RTS/CTS handshake must be used.
- No ACK. The AP shall not wait for ACKs if true.
- Multicast policy. Specifies the multicast policy, which can be Legacy, Direct Multicast Service (DMS), or Unsolicited Restries (UR).
- UR Count. Specifies the number of UR retransmissions.

Transmission Policy configurations can be specified on a L2 unicast destination address basis. As a result, for each destination address and for each slice in the network a specific *Transmission Policy* configuration can be created. Notice how the *Transmission Policy* abstraction allows the controller to specify which set of MCSes can be used by the rate control algorithm implemented by the AP. However, the frame–by–frame selection of the MCS is implemented at the AP and not at the controller. A *Traffic Rule* configuration is identified by the tuple (SSID; DSCP). The next subsection will explain how a precise portion of the flowspace can be assigned to a *Traffic Rule*. Using both the *Traffic Rule* and the *Transmission Policy* abstractions it is possible to ensure that applications and services with the same requirements are provided the same network resources. Note that a default *Traffic Rule* with DSCP set to 0x00 is always present.

4.2.2 End–to–end Slicing

OpenFlow–enabled switches forwarding policy can be configured by a logical network controller by specifying a set of rules. Each rule is composed of a Match, used to identify the flow, and an Action, which specifies the operation to be performed on each packet in that flow, e.g. forward to zero, one, or more output ports, add/remove a header, set a field, etc. The fields that can be used for the Match rule compose the so–called OpenFlow Extended header and include a combination of link, network, and transport header fields. Figure 24 shows some examples of traffic matches that can be defined using the OpenFlow Extended header. The first match includes the traffic with the IP DSCP field set to zero. The second rule includes HTTP traffic with 0x40 as IP DSCP. Traffic matching a given OpenFlow rule is tagged with a unique DSCP value and then dispatched to the Hypervisor. Here, the tuple (SSID; DSCP) is used to redirect the traffic to the correct slice. Notice how, since a Wi–Fi station can be associated to one, and only one, SSID at any given time, the SSID can be easily computed using a packet MAC destination address.

	Ethernet			IP				Transport		
	MAC src	MAC dst	...	IP src	IP dst	IP dscp	....	TCP/UDP sport	TCP/UDP dport	...
Rule 1	*	*	*	*	*	0x00	*	*	*	*
Rule 2	*	*	*	*	*	0x40	*	*	8080	*

Figure 24 The OpenFlow Extended header

4.2.3 Airtime–based Slice Scheduling

The hypervisor presented here, uses a modified version of the Deficit Round Robin policy to schedule *Traffic Rules*. The proposed scheduling policy, named Airtime Deficit Weighted Round Robin (ADWRR), assigns to each *Traffic Rule* a fraction of the airtime according to its relative priority. This is done because, in a wireless network, the cost of transmitting a frame depends on the frame length and on the actual channel conditions experienced by the receiver of that frame. For example, a receiver that is far away from the AP will utilize more radio resources due to the use of less efficient MCSes and/or retransmissions. Our hypervisor uses the transmission statistics maintained by the AP MCS selection module in order to estimate how much airtime will be required to serve a particular *Traffic Rule*. The details of the MCS selection algorithm used by the AP are not important as long as they include, for each client, the link delivery probability for each MCS supported by the AP. In our implementation we relied on the Minstrel [Xia2013] algorithm, which is available in the Linux kernel. Let $P(R_i)$ be the probability of transmitting a frame and receiving the corresponding Wi–Fi acknowledgement using the MCS R_i and let R_{best} be the MCS with the highest delivery probability. The expected transmission airtime A for a packet L bits long can be approximated with:

$$A = \frac{1}{P(R_{best})}\left(DIFS + \frac{L}{R_{best}} + SIFS + T_{ack}\right)$$

Notice how this formula ignores the exponential growth of the congestion window size for each failed transmission. Such a simplification proved sufficient in our measurement campaign. Nevertheless, a better estimate of the transmission airtime can be computed using the results presented in [Bianci2000]. The pseudo code of the enqueue and dequeue processes used by the hypervisor is given respectively in Alg. 1 and 2 below. Variables and data structure used by both algorithms are summarized in Table 9.

Variable	Default	Description
$ActiveQueue$	$\{\emptyset\}$	List of backlogged *Traffic Rule*.
$Q(i)$	$12000\mu s$	*Traffic Rule* i Quantum.
$DC(i)$	0	*Traffic Rule* i Deficit counter.

Table 9 **Hypervisor Data Structure**

The hypervisor maintains a list of currently backlogged *Traffic Rules (ActiveQueue)*. Incoming frames are classified according to the (*SSID; DSCP*) tuple (Alg. 1, row 3) and then fed to the corresponding *Traffic Rule* (Alg. 1, row 5). If such a Traffic Rule does not yet exist, it is created dynamically by the hypervisor. At each round, the deficit counter of the currently visited *Traffic Rule DC(i)* is increased by a fixed quantity *Q(i)* (Alg. 2, row 5). The hypervisor only serves *Traffic Rules* whose expected transmission time is smaller than the deficit counter (Alg. 2, row 8). After a packet is sent, the deficit counter is decreased by the expected transmission time of the transmitted frame (Alg. 2, row 11). A frame whose transmission time exceeds the deficit counter is held back until the next visit of the scheduler (Alg. 2, row 13). Empty *Traffic Rules* are removed from *ActiveQueue* and their deficit counter is set to zero (Alg. 2, row 15).

Algorithm 1 Enqueuing process.

```
1:  procedure ENQUEUE(p)
2:      (SSID, DSCP) ← GetTrafficRule(p)
3:      if (SSID, DSCP) not in ActiveQueue then
4:          ActiveQueue.pushBack((SSID, DSCP))
5:      ActiveQueue((SSID, DSCP)).enqueue(p)
```

Algorithm 2 Dequeuing process.

```
1:  procedure DEQUEUE
2:      (SSID, DSCP) ← GetTrafficRule(p)
3:      if ActiveQueue is not empty then
4:          i = ActiveQueue.next()
5:          DC(i) = DC(i) + Q
6:          while True do
7:              airtime = ActiveQueue(i).computeTxAirtime()
8:              if airtime < DC(i) then
9:                  p = ActiveQueue(i).dequeue()
10:                 p.send()
11:                 DC(i) = DC(i) - airtime
12:             else
13:                 break
14:         if i is empty then
15:             ActiveQueue.remove(i)
```

It should be noted that the possibility of assigning a different quantum to each *Traffic Rule*, and thus to each slice, enables advanced QoS management features. For example, the slices supporting services with stricter performance requirements can be assigned more radio resources by specifying a larger value for the *Traffic Rule* quantum parameter. Notice how, each *Traffic Rule* contains multiple Aggregation Buffers, one for each station in the virtual network. *Aggregation Buffers* share the DC of the parent Traffic Rule. Nevertheless, each *Traffic Rule* can schedule *Aggregation Buffers* in a different manner. For example, one *Traffic Rule* can schedule its stations using a simple Round Robin policy while another *Traffic Rule* could use a best–MCS policy. Notice how, according to the configuration of the parent *Traffic Rule*, *Aggregation Buffers* can generate A–MSDU, A–MPDU, or non–aggregated frames. In case of aggregated frames, the maximum frame length and the aggregation time–out can on a per–*Traffic Rule* basis.

4.3 Implementation Details

The proposed Wi-Fi slicing framework has been implemented and tested on top of the COHERENT C3 and in particular on top of its reference implementation 5G–EmPOWER [Riggio2015].

4.3.1 Control and Data Plane Implementation

Each AP in our implementation consists of two components: one OpenvSwitch instance managing the communication over the wired backhaul; and one Click modular router [Kohler2000] instance implementing the 802.11 data–path. Click is a framework for writing multi–purpose packet processing engines and is being used to implement just the wireless client/AP frame exchange. The network intelligence is implemented at the 5G–EmPOWER controller, which communicates with the APs in the data–plane through its southbound interface using a persistent TCP connection. The protocol used for this communication is outside the scope and a full account of its features can be found online [Empower2018]. Similarly, the OpenvSwitch running within each AP operates under the supervision of an OpenFlow–enabled backhaul controller (Ryu in this particular case [Ryu2018]). The intent–based interface presented in [Riggio2016] is used for the communication between 5G–EmPOWER and the backhaul controller. Such interface has been extended in order to allow the 5G–EmPOWER controller to request the backhaul controller to tag with a particular DSCP code all the traffic matching a certain flow rule and arriving on the backhaul interface of a given AP.

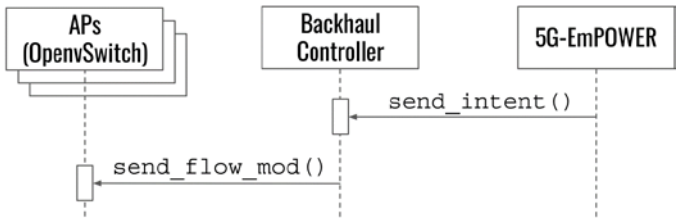


Figure 25 Traffic Rule creation process

4.3.2 Traffic Rule Creation

The *Traffic Rule* abstraction is exposed to the application layer through an object mapping properties to operations. This allows to manipulate the *Traffic Rule* configurations defined for a certain slice by simply accessing the traffic_rules property of a Tenant object (i.e. a virtual network). For example, defining a new Traffic Rule configuration for HTTP traffic can be as simple as shown below.

```
>>> tenant.traffic_rules["tp_dst=8080"] = \
>>>    TrafficRule(tenant, dscp=0x40, quantum=5000)
```

The listing above will trigger two operations: (i) the creation of a new slice for the DSCP 0x40 at every AP in the virtual network using 5000 μs for the slice quantum; and (ii) a message to the backhaul controller through the intent–based networking interface, as depicted in Figure 25. This message has the following structure:

```
{
  "src_dpid": "00:00:00:00:00:0A",
  "src_port": 1,
  "dst_dpid": "00:00:00:00:00:0A",
  "dst_port": 4,
  "matches": {
    "tp_dst": "8080"
  },
  "tag": 0x40
}
```

The pair (src dpid, src port) identifies the AP backhaul port, while the pair (dst dpid, dst port) identifies the virtual port to which the Click instance implementing the Wi–Fi data–path is attached. The semantic of the message is that all the Wi–Fi–bound traffic arriving on the AP backhaul port that matches the specified rule must be tagged with the specified DSCP code. This will allow the hypervisor to dispatch the specified portion of the flowspace to the new Traffic Rule.

4.3.3 Traffic Rule Monitoring and Update

The status of each Traffic Rule can be monitored by the 5G–EmPOWER controller using the slice telemetry framework. More specifically the controller can periodically gather the status of all the Traffic Rules defined in a certain AP. Such status includes for example the current number of backlogged frames, the total airtime spent, the number of transmitted packets and bytes, and the number of dropped packets and bytes. Such information can be used for different purposes. For example the controller could use it to identify if some of the Traffic Rules are inactive thus triggering a reassignment of the available resources to other Traffic Rules.

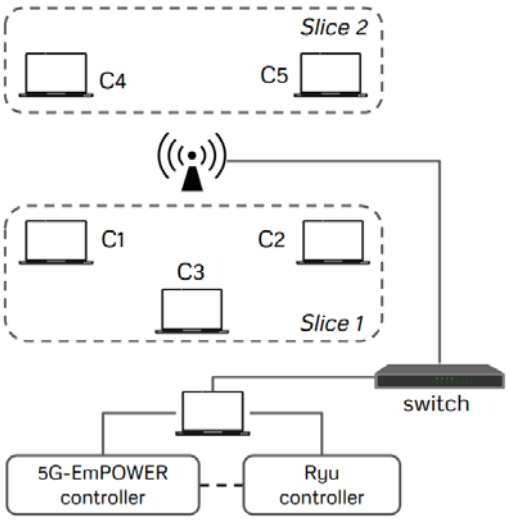


Figure 26 Testbed deployment layout

Test	Users Slice 0	Users Slice 1	DSCP Slice 0	DSCP Slice 1	Channel conditions	Traffic
B1	5	0	-	-	Equal	Const.
B2	5	0	-	-	Different	Const.
1, 9	3	2	0x00	0x00	Equal	Const.
2, 10	3	1	0x00	0x00	Equal	Const.
3, 11	3	2	0x00	0x20	Equal	Const.
4, 12	3	1	0x00	0x20	Equal	Const.
5, 13	3	2	0x00	0x00	Different	Const.
6, 14	3	1	0x00	0x00	Different	Const.
7, 15	3	2	0x00	0x20	Different	Const.
8, 16	3	1	0x00	0x20	Different	Const.
17	3	2	0x00	0x00	Equal	Int.
18	3	2	0x00	0x00	Different	Int.

Table 10 Evaluation scenarios

4.4 Evaluation

4.4.1 Methodology

The proposed Wi-Fi slicing framework has been evaluated on a real–world testbed whose high–level architecture is depicted in Figure 26. The testbed is composed of one AP and five clients. A laptop connected to the wired segment of the network runs both the Ryu and the 5G–EmPOWER controllers. The AP is based on the PCEngines ALIX 2D (x86) processing board and is equipped with two Wi–Fi cards based on the Atheros AR9220 chipset. OpenWRT 15.05.01 is used as operating system for the AP. The experiments are conducted on the 5 GHz band (channel 48) with the cards operating in 802.11n mode. Two slices are created in the network: the first one consists of three users (C1, C2 and C3), while the second one consists of two users (C4 and C5). The wireless clients are located 5 m away from the AP and are standard laptops. The evaluation comprises eighteen different scenarios divided into three groups. In all the scenarios, a saturated UDP stream was generated between the laptop running the controllers and each wireless client. Each measurement was 60s long. Traffic was generated using Iperf tool. The results reported in the next section are the average of 10 runs. The first group of experiments (1 to 8) are carried out without any kind of AP/controller telemetry. Then, the same experiments are repeated with the slice telemetry enabled (9 to 16). In this configuration the controller periodically polls the AP to gather the slice utilization statistics. The polling period was set to 100ms. These two groups of experiments make use of constant bitrate traffic. By contrast, intermittent traffic is used in scenarios 17 and 18. Finally, the scenarios named B1 and B2 correspond to the baseline configuration in which no hypervisor is used. As evaluation metrics we use the aggregated throughput, the bandwidth achieved by each client, and the jitter. Apart from the aforementioned transmissions, no other traffic exists in the network. A summary of the characteristics of the different scenarios is provided in Table 10.

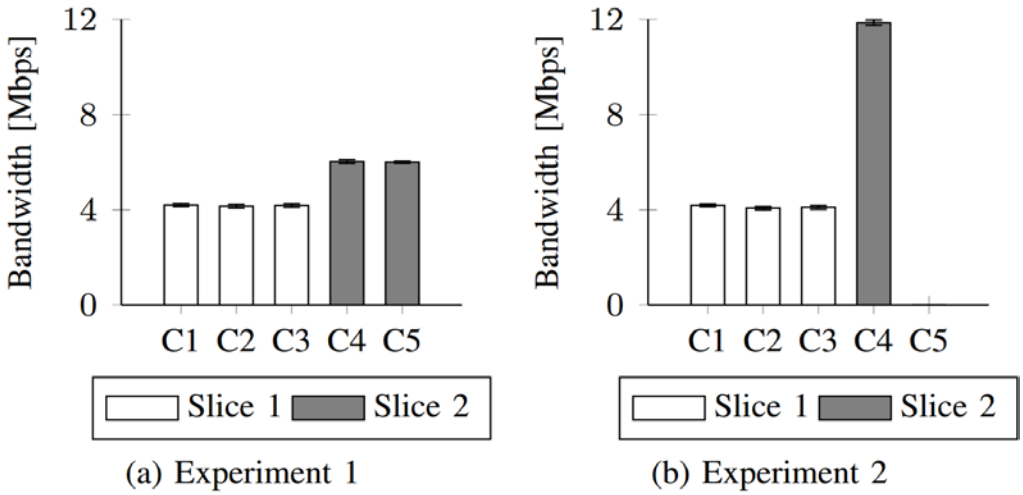


Figure 27 Bitrate comparison for two network slices with same priority and equal channel conditions for the clients

4.4.2 Results

4.4.3 Traffic Prioritization

Experiments 1 and 2 aim to show the isolation across slices and how the number of clients per slice does not affect the performance of the other slice. Experiment 1 demonstrates the isolation between slices when both slices have the same priority and when all the clients experience the same channel conditions. The results for this experiment are reported in Figure 27(a). The Quantum of both slices has been set to the time necessary to transmit a standard Ethernet frame (1500 bytes of MAC payload) at the 6 Mbps basic rate in IEEE 802.11n. As it can be seen from the figure, the Quantum is equally divided between the stations in each slice. Experiment 2 follows the same approach of experiment 1 with the difference that in this case a single client is active in Slice 2. As it can be seen in Figure 27(b), the slice isolation feature provided by our hypervisor allows the client in Slice 2 to fully use the resources assigned to its slice. Moreover, it can also be noticed that the performance of the clients in

Slice 1 are not affected by the redistribution of resources in Slice 2. Experiments 3 and 4 present the same scenarios described in the previous measurements with the main difference that, in this case, Slice 2 is given twice the Quantum of Slice 1. Figure 28 sketches this new situation, and proves how the slices isolation is guaranteed along with the priority assignment.

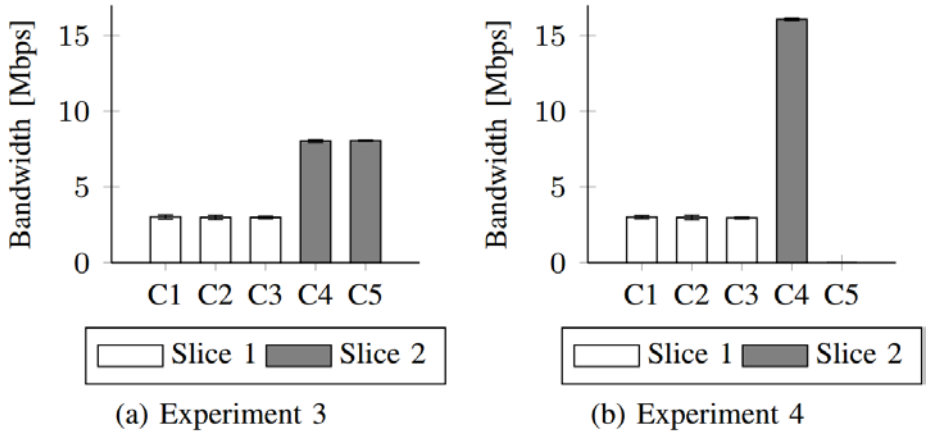


Figure 28 Bandwidth comparison for two network slices with different priority and equal channel conditions for the clients.

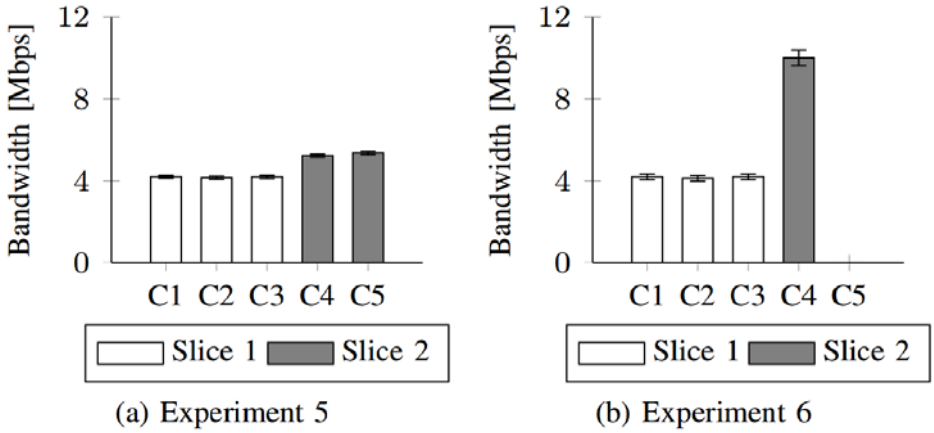


Figure 29 Bandwidth comparison for two network slices with the same priority and equal channel conditions for the clients.

4.4.4 Performance Isolation

This group of experiments aims at demonstrating the hypervisor performance isolation feature when some of the clients experience poor channel conditions. We remind the reader that the hypervisor follows an ADWRR scheduling discipline. Therefore, the transmission opportunities of the users depend on the required transmission time, and hence, on the channel conditions that they experience. The first set of experiments (5 to 8) aims to prove how the hypervisor can ensure the slices coexistence even when one of the users experiences poor channel conditions. Figure 29 plots the results in terms of bandwidth for the experiments 5 and 6, where the test conditions are the same than in experiments 1 and 2. However, in this case, the station C4 is placed 25 m away from the AP. This scenario demonstrates that the performance of the Slice 1 does not present changes with respect to the first scenarios, and therefore, it is not affected by the poor channel conditions experience by the users in the other slice. Moreover, in Figure 29(b) it can be seen that when only the station C4 is connected to the

Slice 2, its performance is, due to the poor channel conditions, lower than in the previous case. However, the throughput in the Slice 1 is still unaffected.

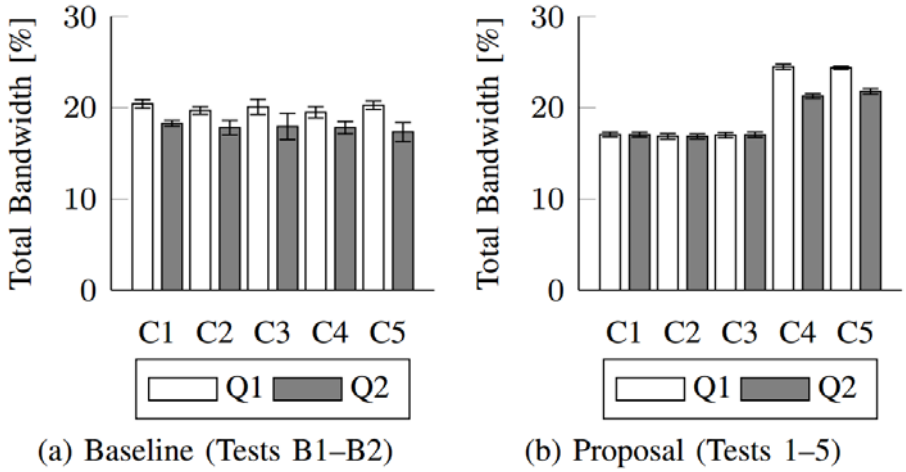


Figure 30 Bitrate comparison of a baseline scenario and the proposal using two same priority slices

The tests 7 and 8 present the same results but having increased the priority of the Slice 2. For this reason, and due to space constraints, the graphical results of these tests are omitted. In a system without slicing capabilities all the stations will equality share the available radio resources only if they experience similar channel conditions. If such conditions are not met, the performance of all the users in the network will be penalized due to the so called IEEE 802.11 *Performance Anomaly* [Heusse2003]. Figure 30(a) shows this situation by using two different channel qualities. In the scenario marked as Q1 all the users experience similar channel conditions, while in scenario marked as Q2 the user C4 experiences poor channel conditions. This issue can be also verified in Figure 31(a), where it can be seen how the transmission jitter for all the stations increases when a user experiencing poor channel conditions is added to the network. Conversely, as it can be seen in Figure 30(b), using our hypervisor only the slice with the client experiencing poor channel conditions is penalized. Furthermore, the transmission jitter is also maintained for the clients in Slice 1, while the one in Slice 2 is just slightly increased when these issues appear in the network (Figure 31b).

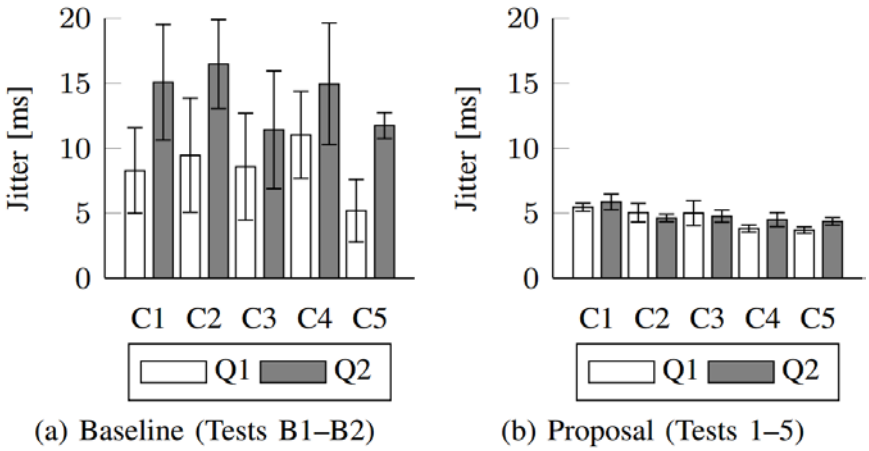


Figure 31 Jitter comparison between a baseline scenario and the proposal using two Traffic Rules with the same priority

4.4.5 Resource Reallocation

Experiments 17 and 18 aim at demonstrating the system ability to dynamically re-allocate radio resources upon changes in the network traffic. To this end, in the experiment 17 the stream addressed to client C5 stops at 30s. In Figure 32 it can be seen that when the transmission finishes, the resources

are dynamically re-assigned to the remaining clients in the same slice. Similar results are shown in Figure 33 for the experiment 18, which differs from the previous one in the fact that the client C4 is experiencing poor channel conditions. As it can be observed, when the stream addressed at client C5 stops, the resources of the Slice 2 are entirely assigned to the client C4

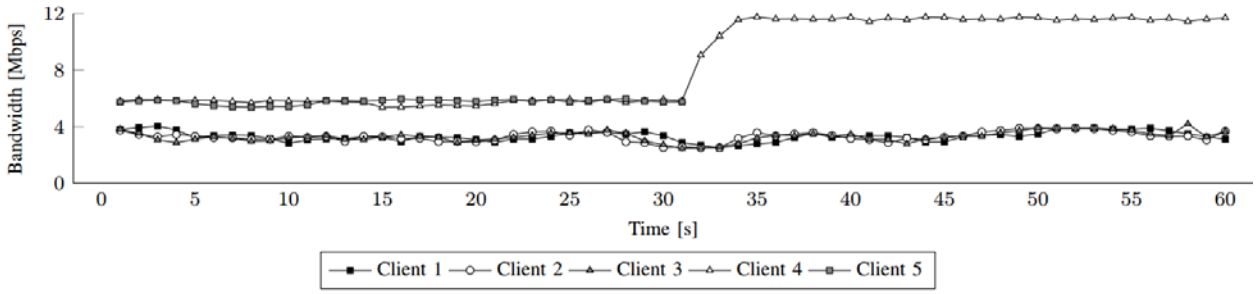


Figure 32 Resources reassignment over time after the end of the transmission of the client C5 for equal channel conditions.

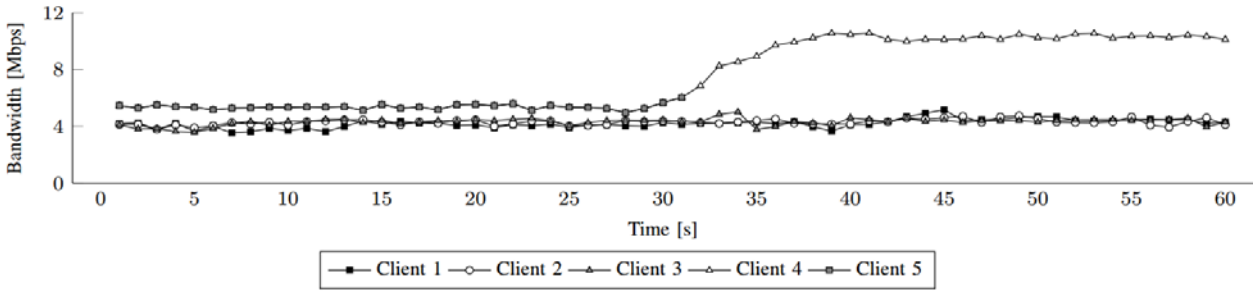


Figure 33 Resources reassignment over time after the end of the transmission of the client C5 for different channel conditions

4.4.6 CPU and Memory Consumption

In this set of experiments we aim at demonstrating that the computational overhead introduced by our hypervisor does not penalize the data–path performance nor it incurs in high CPU and/or memory utilization. In this regard, Table 11 compares the CPU and memory utilization of the baseline system with the ones of the system utilizing the proposed hypervisor with and without the slice telemetry. As it can be seen, the differences found in the CPU consumption are practically negligible. Also, the increase in the memory consumption without introducing slice telemetry is just around 3%, which increases to 7% when the telemetry is used. Finally, the impact of the slice telemetry on the data–path performance has always been lower that 1%.

	CPU [%]		Memory [%]	
	μ	σ	μ	σ
Baseline	77.579	4.946	23.184	0.211
Proposal without telemetry	79.572	7.658	26.637	0.735
Proposal with telemetry	80.481	6.579	30.559	0.260

Table 11 CPU and memory usage results

4.4.7 Slice Telemetry

As stated in the previous section, all the experiments between 1 and 8 have been performed with the slice telemetry disabled, while experiments from 9 to 16 were performed with the slice telemetry enabled. However, the results difference is practically negligible and always lower than 1% (which is a value within the confidence interval of the experiments). Therefore, due to space constraints, only the results without telemetry are shown for all the experiments.

4.5 Conclusion

4.5.1 Technical/Scientific Impact

In this section we reported on the design, implementation, and evaluation of a programmable end-to-end slicing framework for 802.11-based WLANs. We introduced the Traffic Rule abstraction mapping a precise portion of the flowspace to a certain scheduling discipline. An experimental evaluation performed over a real-world testbed and involving a wide range of scenarios has demonstrated the resource allocation and performance isolation features of Lasagna.

4.5.2 Feedback toward development

This work has potential for further extensions. For example, we plan to implement new Aggregation Buffers scheduling disciplines and at the same time to allow the controller to swap such disciplines at runtime. Moreover, we also plan to use the framework to jointly optimize user association and radio resource utilization across the entire network. Finally, we are also considering porting the core slicing framework from Ryu to other OpenFlow controller (e.g. ONOS).

4.5.3 Expected business impact

End-to-end slicing will be an important element if Wi-Fi is to become and effective traffic offloading solution for cellular network. The proposed framework can allow operator to partition specific services and applications in their own slice of resource and to allow isolation between concurrent slices.

5. Algorithms for Channel–Aware User Association in Software–Defined Wi–Fi Networks

5.1 Overview

The past years have witnessed a sustained increase in mobile traffic demands which is forecasted to reach 49 exabytes per month by 2021 [Cisco2017]. Due to its low deployment and operational costs, Wi–Fi has emerged as an efficient way to satisfy such demands. Originally relegated to residential and enterprise scenarios, Wi–Fi is becoming a viable traffic offloading solution for cellular networks. Nevertheless, its unplanned nature coupled with its contention–based channel access scheme leads to sub–optimal performance when the network density increases. Moreover, Wi–Fi networks operate in unlicensed bands as opposed to the licensed spectrum used by cellular networks. While this makes Wi–Fi networks extremely easy to deploy, it also makes them more vulnerable to interference from co–located deployments. The growing popularity of 5 GHz–capable devices is mitigating this issue in indoor settings, where the penetration through the walls of high frequency signals is limited. However, this does not apply to outdoor scenarios or to networks in the 2.4 GHz band. In addition to the mentioned pitfalls, Wi–Fi networks leave clients in charge of selecting the Access Point (AP).

The actual algorithm used by the clients for the AP selection is not specified by the standard and is left as implementation choice for the vendor. RSSI measurements are typically used to perform this operation, i.e. the client selects the AP with the highest RSSI. Such approach however does not consider the AP load and may lead to an uneven clients' distribution across the network. Finally, only a limited number of channels are available in both the 2.4 GHz and the 5 GHz bands. As a result, a severe throughput degradation is expected when multiple APs are in the same collision domain, especially when the number of active APs per unit of area increases. Therefore, an effective collision domain isolation and channel assignment strategy becomes essential to ensure optimal performance [Chieochan2010]. In recent years different solutions have emerged to solve these problems. Nevertheless, the traditional Wi–Fi architectures makes it hard to add new mechanisms without modifying the standard. Software Defined Networks present a possible solution to address these challenges. However, albeit SDN is already an established technology in the wired domain, with OpenFlow playing the role of de–facto standard [Mckeown2010], equivalent solutions for wireless and mobile networks have only recently started to appear [Riggio2015], [Suresh2012].

Here, we present Wi–Balance, a joint channel selection and user association scheme for Wi–Fi–based WLANs. The contribution is two–fold. On the one hand, a constraint programming algorithm is designed to isolate possible collision domains among the APs. On the other hand, we present a user association scheme capable of detecting situations in which the traffic is not efficiently distributed and to transparently reschedule to other APs the clients whose transmissions are causing performance issues. Based on a real–world evaluation we have demonstrated an improvement of up to 25% and 30% in terms of network throughput and channel utilization compared with a standard RSSI–based user association mechanism. We have released the entire implementation, including the controller and the data–path, under a permissive APACHE 2.0 license for academic use.

5.2 Channel-aware user association

Interference and collisions are the most important causes of performance degradation in WLANs [Karupongsiri2017], [Yuan2009]. When several clients attached to the same AP transmit at the same time, the network may suffer delays, service interruptions and performance drops. Figure 34 depicts the relationship between channel utilization and network performance. During the measurement three clients were transmitting with bandwidth requirements ranging from 5 to 50 Mbps towards the same AP. As can be seen, when the channel occupancy is higher than 60%, the delivery ratio dramatically drops. This is due to the collisions in the wireless medium and the decrease in the data rates used for the transmission. We remind the reader that the Modulation and Coding Scheme (MCS) adaptation algorithms tend to select lower data rates upon several failed transmissions, which in time increases the channel utilization. This simple scenario demonstrates the importance of an efficient network resource allocation in terms of both channel assignment and user association. This aspect acquires even more

relevance when considering mobile clients. We address this challenge by presenting an SDN–based joint user association and channel assignment algorithm.

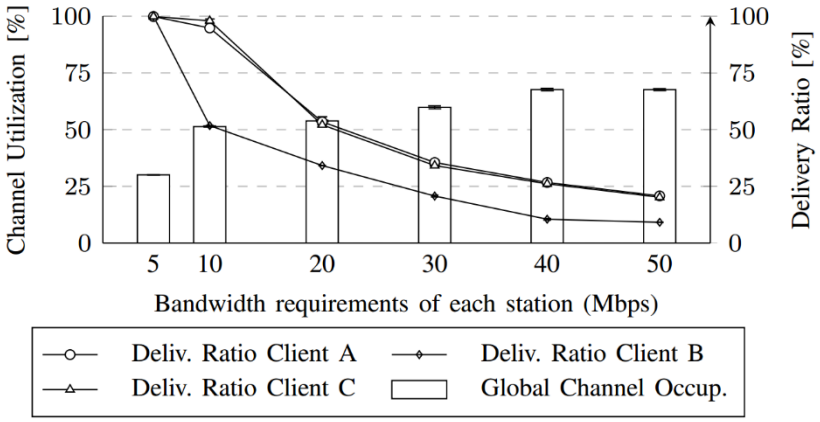


Figure 34 Delivery ratio of three stations attached to a single AP performing uplink transmissions with different bandwidths

5.2.1 Channel Assignment Algorithm

Channel assignment must be done in such a way to minimize interference between APs that are in the same collision domain. Two APs are in the same collision domain if they are tuned on the same channel and if they are within carrier sensing range of each other. In this case, if multiple transmissions start at the same time they can either collide or one of the transmissions must be delayed. In either case a reduction in the aggregated network throughput is to be expected. The efficiency of a channel assignment procedure depends on the number of available channels and on the number of APs in the same collision domain. The higher the number of available channels, the lower the probability of finding two APs using the same one. Therefore, it is crucial to identify the channels used by the APs in the neighboring networks, since they may share the same collision domain. However, after identifying these channels, the set of available channels for the assignment may be very limited, especially in congested areas such as office buildings or universities. A channel assignment algorithm must have as input the interference map of the WLAN. In other words, it must consider for each AP, the set of surrounding APs that must not operate on the same channel, as well as the list of available channels. The interference map is built in the first step of the algorithm and its data is designated as the constraints of the problem. Moreover, a periodic analysis of the wireless medium must be carried out to update the network information. Notice that SDN–based solutions allow the channel assignment algorithm to have a complete view of the network (which is collected and maintained by the SDN controller). Considering this, a constraint programming algorithm has been designed to solve the channel assignment problem. The recursive algorithm is shown in Algorithm 1 below. The algorithm first tries to assign a channel to the set of APs with the lowest number of available channels. We refer to available channels as those that have not been still assigned to the neighboring APs of a certain AP and do not overlap with the ones already assigned to them. Then, the algorithm selects in this set of APs the one with the highest number of neighbors already assigned. Furthermore, if all the channels have been already taken by the neighboring APs, the algorithm selects the channel that has been used by the lowest number of APs. In case that multiple channels match this condition, the channel with the lowest occupancy ratio is chosen. The algorithm finishes when it finds a configuration that minimizes the number of APs in the same collision domain. Although after performing an efficient channel assignment the network interference may have been significantly reduced, there is still room for improvement.

Algorithm 1 Channel assignment procedure

Input:
$neighbors$: graph storing the neighbours of each AP.
$channels$: list of available channels.
$overlaps$: dictionary storing the overlapping channels.
Output:
$assignment$: dictionary of (AP, channel) assignment

```
procedure SOLVE(neighbors, channels, assignment)
    remainingAPs ← APs ∉ assignment
    if len(remainingAPs) == 0 then
        return assignment        ▷ It becomes the solution
    Sort remainingAPs by the lowest number of available channels and the highest number of neighbors in assignment
    nextAP ← remainingAPs[0]
    possibleCh ← channels
    for each AP ∈ neighbors[nextAP] do
        APCh ← assignment[AP]
        possibleCh ← possibleCh − APCh
        possibleCh ← possibleCh − overlaps[APCh]
    if not possibleCh then
        possibleCh ← min(assignment)
    for each channel ∈ possibleCh do
        assignment[NextAP] ← channel
        return SOLVE(neighbors, channel, assignment)
```

5.2.2 User Association Algorithm

After the channel assignment, the controller performs a neighbor discovery process to build the channel quality map. This map includes for each station the channel quality with respect to all the APs in the network. The channel quality map is built by the SDN controller by retrieving from each AP the list of stations in its coverage area. Similarly, the controller periodically gathers the statistics of the rate adaptation algorithm maintained by each AP. For each station and for each supported MCS, the Exponentially Weighted Moving Average (EWMA) of the delivery probability and the expected throughput in the last observation window are reported. Moreover, the number of successful and failed transmissions are also reported. We remind the reader that this information is maintained by the rate adaptation algorithm implemented by the AP. Therefore, no extra computation is added to the APs. Gathering this statistical data needs some limited signaling between the controller and the APs. The details of this protocol are outside the scope and can be found online [Empower2018]. It is also important to highlight that Wi–Balance does not require any change to either to the IEEE 802.11 protocol nor to the wireless devices. The whole process is sketched in Figure 35.

Let us define U as the set of stations in the network, M as the set of Wi–Fi APs and $\Omega(u) \subseteq M$ as the set of APs within the coverage area of the user $u \in U$. Using the statistical data collected by the controller, Wi–Balance computes the channel utilization $\mu(n)$ for each $n \in M$ and the average channel occupancy across all the APs in the network μ. If a significant difference between μ and any occupancy ratio is found a user re–association process is triggered for the affected AP. In particular, Wi–Balance collects, for each user u attached to the affected AP n, the channel utilization of the surrounding APs, $\Omega(u)$, and the RSSI level between each AP $m \in \Omega(u)$ and the station u, let us call this quantity R_u^m. After that, Wi–Balance selects as candidate AP for the handover the AP offering the lowest result of the product between the current occupancy ratio of AP n, i.e. $\mu(n)$, and the perceived signal strength R_u^m for each $m \in \Omega(u)$. Then, the client handover is performed. The average channel occupancy μ is recalculated to check if the network redistribution was efficient. Otherwise, the handover is reverted. This process is also triggered in case of observing a sudden change in the RSSI value for any client, which could result from the movement of that client.

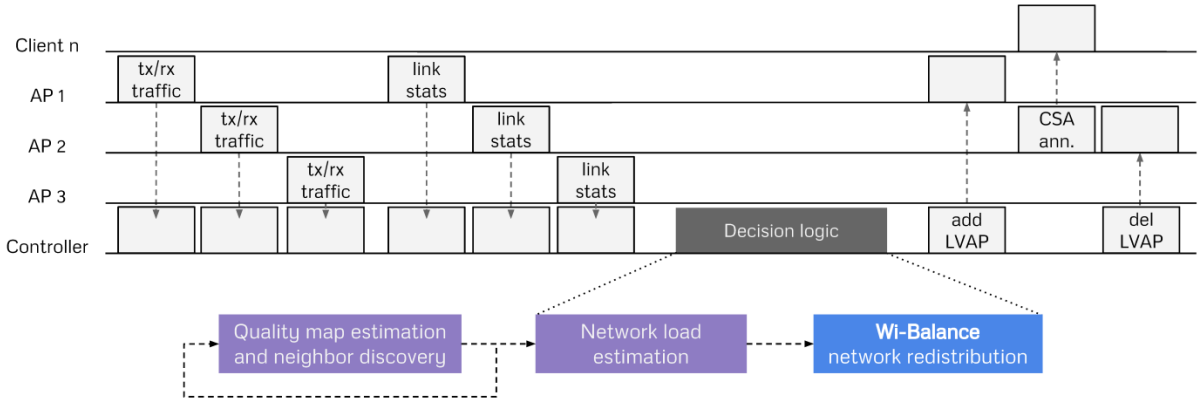


Figure 35 Scheme of the working mode of Wi–Balance

5.2.3 Complexity Analysis

In this section we will analyze the computational complexity of Wi–Balance, distinguishing between the channel assignment and the user association algorithms. The channel assignment algorithm is a recursive procedure that is called n times until a channel has been selected for each AP. The recursive nature makes the algorithm have two cases: a base and recursive case. To solve this problem, we use a recurrence relation denoted as $T(n)$. The base case encompasses the scenario in which all the APs have been visited, and thus, $n = 0$. At this point, the complexity of $T(0)$ is essentially constant and equals to $O(1)$. In the recursive case, i.e. when $n > 0$, two aspects must be considered: i) the function is recursively called with $n - 1$; ii) the channel search operations are internally performed for that AP. The cost of (i) is $T(n - 1)$, while the cost of (ii) must be further explored. First, the n remaining APs are sorted by the lowest number of available channels and the highest number of neighbors. The complexity of this step is $O(n \cdot log(n))$. Then, the list of neighboring APs for the first AP in the list is traversed to discover the available channels, which results in a cost $O(n)$. In the worst case in which there are no available channels, the algorithm will select the channel less used by the neighbors, hence adding a complexity $O(n)$. After that, the algorithm must iterate through the list of possible channels, which in the worst case will be n. On this basis, the cost of (ii) is estimated as $O(n)$, and hence the relation $T(n)$ can be expressed as $T(n) = T(n\text{-}1) + O(n)$. Thus, the complexity of the channel assignment is $O(n^2)$. Every time the user association algorithm is called, the list of APs must be traversed to compute their channel occupancy ratio. Therefore, the complexity of this operation is $O(n)$. Computing this ratio requires to calculate the fraction of time used by the stations attached to each AP. In the worst case, all the stations in the network, s, will be attached to the same AP, which results in a computational complexity $O(s)$. On this basis, the cost of computing the channel utilization will be as high as $O(n \cdot s)$. Moreover, the average channel utilization must be calculated. Notice that this estimation depends on the number of APs, hence it being as complex as $O(n)$. Then, the individual ratio of each AP must be compared with the average one to find imbalances in the distribution of the network load. Therefore, the list of APs must be once again traversed, resulting in a complexity of $O(n)$. In case of finding an imbalanced AP, the list of all its clients must be traversed, and for each client, the algorithm must iterate through all its possible APs to perform a handover. Thus, the complexity of the use association algorithm is $O(n \cdot s)$. Finally, the overall computational complexity of the joint channel assignment and user association algorithm is $O(n \cdot s + n^2)$ which can be approximated as $O(n \cdot s)$ since in most cases $s >> n$.

5.3 Implementation Details

The WiBalance algorithms have been implemented and tested on top of the COHERENT C3 and in particular on top of its reference implementation 5G–EmPOWER [Riggio2015].

The original seamless handover enabled by the LVAP concept [Suresh2012] does not work when the APs operate on different channels. We removed this limitation by using the Channel Switch Announcement (CSA) defined by the IEEE 802.11 standard. The CSA procedure was originally designed to allow APs to inform the attached client that the operating channel of the hotspot was about to change. This information is delivered inside the standard beacons frames. An AP that is planning to

switch the operating channel will start advertising the new channel in its beacons. A countdown is started, and the channel is switched after a configurable number of beacons (three usually).

In traditional Wi–Fi networks, beacons are sent as broadcast management frames. Conversely, in our case each LVAP sends its own beacons using unicast frames. This is possible because an LVAP is created for each station attached to an AP. Such a design choice allows us to target a CSA message to a station by enabling it only for the LVAP that was created for that station. The seamless handover across APs tuned on different channels and/or bands is enabled by first creating an LVAP on the target AP. This LVAP is initially inactive since the station that it is mapping is tuned on a different channel. Then the controller instructs the LVAP on the source AP to start a CSA procedure. At the end of this CSA procedure the LVAP at the source AP is automatically removed. In the meantime, the station will have switched channel and will have found its LVAP on the target AP.

The full process is sketched on the right–hand side of Figure 35. It should be also noted that the performance impact of these unicast beacons is very low given their short duration and length. However, a trade–off can be set between the duration of the handover and the number of beacons in the network. If this feature is disabled, the impact on the network will be decreased at the price of a longer time to perform the handover. If it is enabled, a faster handover is possible at the price of a little increase in the management traffic.

5.4 Evaluation

In this section we report on the results of the performance evaluation. We compare the network performance using Wi–Balance with the network performance using an RSSI–based user association algorithm.

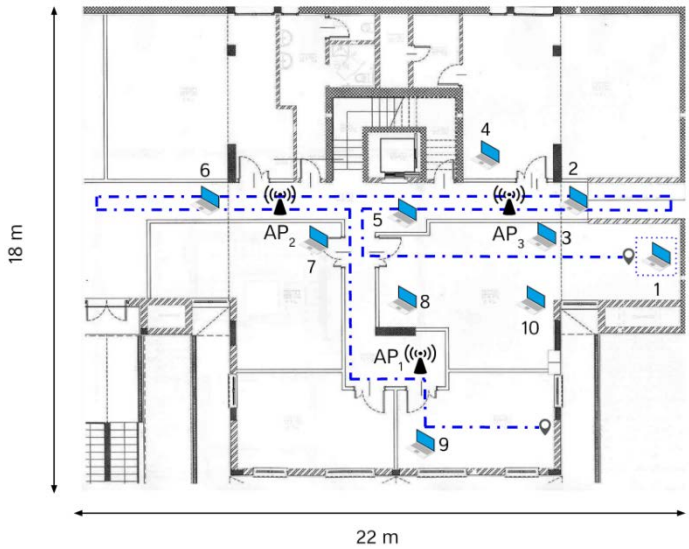


Figure 36 Testbed deployment layout and APs–users distribution.

5.4.1 Evaluation Methodology

The performance evaluation is carried out on a real–world testbed composed of three APs. The layout of the testbed is depicted in Figure 36. The APs are built upon the the PCEngines ALIX 2D (x86) processing board and run OpenWRT 15.05.01. The Wi–Fi cards are based on the Atheros AR9220 chipset. All the experiments are carried out on the 5 GHz frequency band using the IEEE 802.11n physical layer. The channels used by the APs are selected by the channel assignment algorithm presented previously. The scenario also comprises the 5G–EmPOWER controller (not shown in the picture) and a set of 10 stations. One of these stations moves following the path marked in blue in Figure 36. The remaining stations are static and are deployed randomly across the entire floor. Dell–branded laptops powered by an Intel i7 CPU and running Ubuntu 16.04.02 are used as wireless clients. It should be noted that our solution can be applied to other scenarios in the 2.4 GHz band and including both uplink and downlink traffic, as well as different number of stations and APs. Nine experiments,

identified with the letters from A to I, have been conducted. Each test has a duration of 5 minutes and consists of a single UDP or TCP stream between wireless clients and a server sharing the same backhaul with the APs. In the case of UDP traffic, different bitrates are used. The set of 10 users is divided into 2 groups with 5 stations each for the tests from A to F. The first group performs transmissions with a constant bitrate that is maintained for the entire duration of the measurement. By contrast, the second group uses intermittent transmissions. These stations transmit traffic for 40 seconds, and after that, they stop the transmission for 20 seconds. This pattern is repeated until the end of the experiment. Then, the role of the groups is inverted, i.e. the stations with constant bandwidth perform intermittent transmissions, and vice versa. In the experiments from G to I, all the stations generate constant bitrate traffic. A summary of the different scenarios can be found in Table 12. The effectiveness of our proposal is compared with the RSSI–based scheme, in which the stations intend to associate to the AP providing the strongest signal. As evaluation metrics we have considered the delivery ratio, the aggregated throughput, the wireless channel utilization, the Jain's fairness index [Shi2014] and the retransmission ratio. Apart from the uplink transmissions, no downlink traffic exists between the APs and the stations.

Test	Traffic type	User groups traffic dist.	Bandwidth (Mbps)
A	UDP	Constant - Intermittent	10
B	UDP	Constant - Intermittent	5
C	TCP	Constant - Intermittent	-
D	UDP	Intermittent - Constant	10
E	UDP	Intermittent - Constant	5
F	TCP	Intermittent - Constant	-
G	UDP	Constant - Constant	10
H	UDP	Constant - Constant	5
I	TCP	Constant - Constant	-

Table 12 Configuration of the measurements campaign.

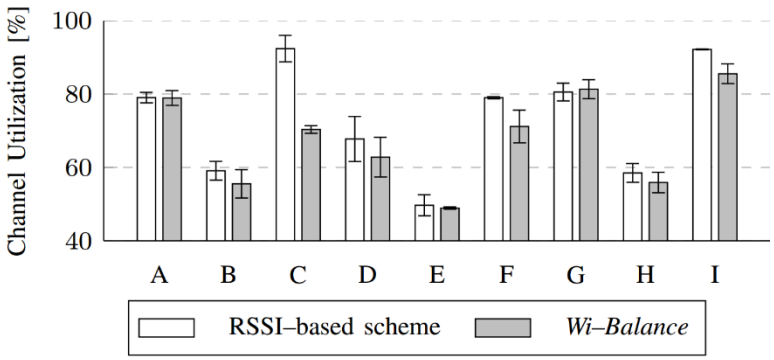


Figure 37 Network-wide channel utilization for UDP and TCP traffic transmissions.

5.4.2 Experimental Results

Especially in situations of congestion, an uneven distribution of the stations may cause some of the APs to be saturated, while some others are idle. Therefore, the users connected to first group of APs will share the available bandwidth which in time could result in a lower aggregated network throughput compared to a situation with an even distribution of the stations across the various APs. From the results shown in Figure 37, it can be observed that the average channel occupancy ratio with Wi–Balance is up to 30% lower than the channel occupancy ratio with the RSSI–based scheme. This is achieved through a more efficient users' distribution, which results in a more balanced network and a decrease in the channel contention. In addition to reducing the overall channel utilization, it is even more important that the APs have an occupancy ratio that is as similar as possible. This situation is displayed in Figure 38, where the average deviation of the channel utilization of each AP with regard to the average network–wide ratio using Wi–Balance is compared with the channel utilization obtained using the RSSI–based scheme. As can be seen, the utilization of each AP widely differs for the reference scheme, while this ratio is more balanced in the case of Wi–Balance. Figure 39 plots the delivery ratio achieved in the tests using UDP traffic. In all the experiments Wi–Balance outperforms the results obtained by the RSSI–based scheme by an average of 17%, and up to 25% in the experiments D and H.

The network–wide aggregated throughput is presented in Figure 40 for the UDP and TCP traffic. The figure shows that the efficient scheduling of the stations leads to an increase in the throughput by an average of 16% and up to 25% in the scenarios D and H. In addition to enhancing the performance, an efficient load–balancing algorithm must distribute the bandwidth evenly among the stations. To demonstrate this effect, Figure 41 compares the Jain's fairness index of the stations throughput using Wi–Balance and the RSSI–based scheme. As can be seen, Wi–Balance delivers a better fairness in all the experiments. Wi–Balance performs better than the RSSI–based scheme also for the mobile users, as can be observed in Figure 42. This is because when a station moves over the coverage area, the AP to which it is connected is not chosen only according to the signal strength, on the contrary also the AP traffic load is considered. For this reason, the throughput improvement is notably higher for the mobile users. Finally, the efficient usage and scheduling of the network resources makes also possible to enhance the network reliability. Since Wi–Balance results in a more uniform wireless client distribution, the retransmission ratio is also decreased by an average of 30%. This phenomenon is displayed in Figure 43.

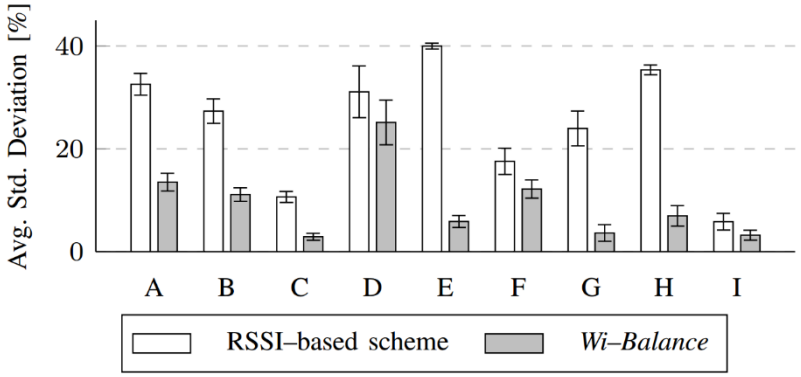


Figure 38 Average deviation of the channel utilization of each AP with regard to the network–wide ratio for both the UDP and the TCP traffic transmissions.

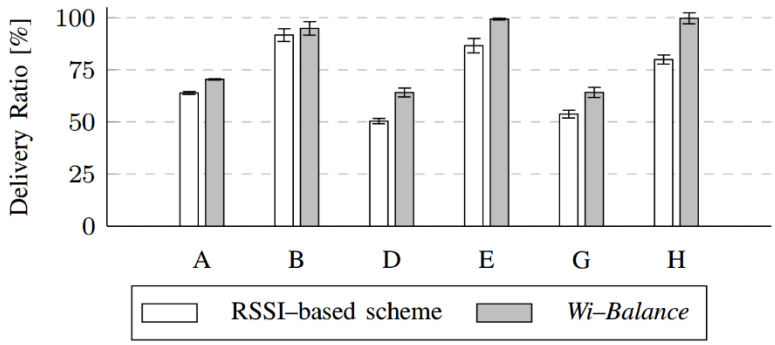


Figure 39 Average delivery ratio for the UDP traffic transmissions at 5 and 10 Mbps

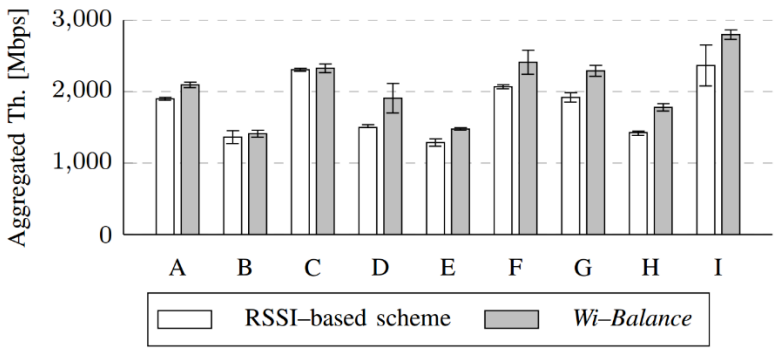


Figure 40 Network–wide aggregated throughput for both the UDP and the TCP traffic transmissions.

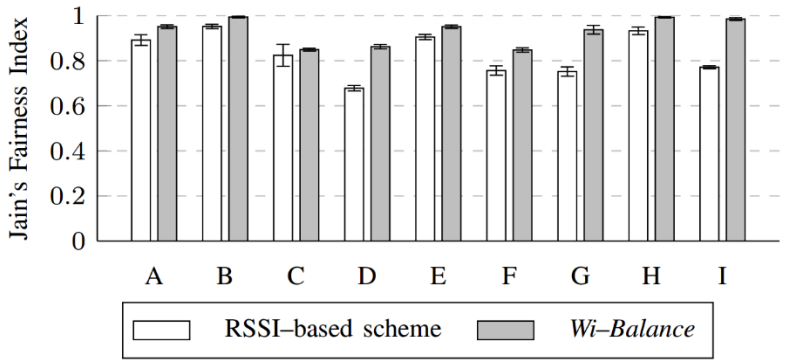


Figure 41 Jain's fairness index of the throughput achieved by all the wireless clients for both the UDP and the TCP traffic transmissions

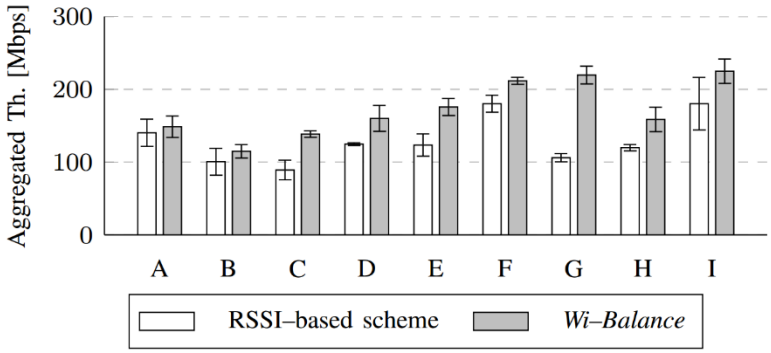


Figure 42 Average throughput achieved by the mobile station for both the UDP and the TCP traffic transmissions

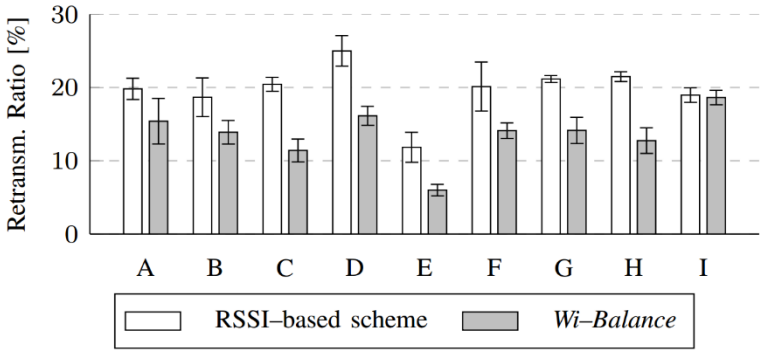


Figure 43 Network–wide average retransmission ratio for both the UDP and the TCP traffic transmissions

5.5 Conclusion

5.5.1 Technical/Scientific Impact

We presented Wi-Balance as a novel SDN-based solution for joint user association and channel assignment in Wi-Fi networks. We also introduced a seamless handover mechanism for Wi-Fi networks capable of operating in a multi-channel environment. The performance of Wi-Balance has been evaluated in a real-world testbed under different scenarios considering mobile and static users. More specifically, compared with RSSI-based user association schemes, Wi-Balance can reduce the channel utilization by up to 30% and can improve the aggregated network throughput by up to 28% without penalizing the network fairness. Conversely a slight improvement in the Jain's fairness index can be noticed when using Wi-Balance.

5.5.2 Feedback toward development

This work can be extended in future to consider the wired backhaul in the user association algorithm. We also plan to support the traffic prioritization and aggregation features supported by the 802.11e and 802.11n standards as well.

5.5.3 Expected business impact

Originally relegated to residential and small networks, Wi-Fi is currently extending its reach to new and challenging use cases. Examples include enterprise networking, cellular networks offloading, and infotainment in high-speed trains and airplanes. Even in its original residential context the demands imposed on Wi-Fi have dramatically changed. If 10 years ago only a limited number of devices (1 or 2) were connected to a single AP, now it is common to have 8-10 data-hungry devices connected to an increasing number of APs (especially if we consider also the range extenders). As a result it becomes increasingly important to be able to properly load-balance and prioritise traffic over 802.11-WLANs if current and future network services and applications must be supported. This paves the way to the utilization of the COHERENT platform in a wide range of Wi-Fi deployments, from home networks, to metropolitan hotspots, to enterprise networks.

6. Algorithm for traffic–Aware User Association in Heterogeneous RANs

6.1 Overview

To accommodate the forecasted growth in mobile communication and associated traffic demands, MNOs will have to upgrade their infrastructure. Boosting mobile network capacity in terms of both coverage and supported data traffic rate can be achieved in several ways such as exploiting additional spectrum, deploying denser radio access networks (RAN), or offloading part of the mobile data traffic to other RANs. Given its low deployment and operation costs, Wi-Fi networks are an efficient traffic offloading solution for mobile networks. Traffic steering between cellular and Wi-Fi networks is possible through the Access Network Discovery and Selection Function (ANDSF) that has been introduced in LTE Release 8. ANDSF is a 3GPP–defined core network entity that allows UEs to select the proper radio access technology (if more than one is available). However, Release 8 does not allow a UE to be simultaneously connected to multiple RANs. The Multi Access Packet Data Network (PDN) Connectivity feature introduced in LTE Release 10 addresses this limitation by allowing UEs to attach to multiple RANs (e.g. Wi-Fi and LTE). MNOs can benefit from this feature by offloading best effort traffic to the Wi-Fi network while keeping demand-attentive traffic on the mobile network. In this context, the contribution presented in this section is two-fold. First, we formulate a traffic–aware user association problem aiming at optimizing resource utilization in a heterogeneous Wi-Fi/LTE RAN. Integer Linear Programming (ILP) techniques are used to derive the optimal solution. Then, a scalable heuristic is proposed to tackle the scalability problems of the ILP–based association algorithm. Second, we implement and test the heuristic in real–world conditions. The implementation of the user association algorithm is released under a permissive APACHE 2.0 License

6.2 System Model

6.2.1 RAN Model

Let $N_n = (N_{enb}; N_{ap})$ be the set of $n_1 = |N_{enb}|$ eNBs and $n_2 = |N_{ap}|$ APs deployed in the heterogeneous RAN. Each node $n \in N_n$ is associated with a geographic location $loc(n)$, as x, y coordinates. A coverage radius $\delta(n)$ is also associated to each node $n \in N_n$. A single weight $\omega_n^c(n) \in R$ with $0 \leq \omega_n^c(n) \leq 1$ is assigned to each node $n \in N$ modelling its available resources. Initially, $\omega_n^c(n) = 1 \; \forall n \in N$. Table 13 summarizes the RAN model parameters.

Variable	Description
N_n	RAN nodes.
N_{enb}	LTE eNBs in $\in N_n$.
N_{ap}	WiFi APs in $\in N_n$.
$\omega_r(n)$	Available radio resources at node $n \in N_n$.
$loc(n)$	Geographical location of node $n \in N_n$.
$\delta(n)$	Coverage radius of node $n \in N_n$.

Table 13 RAN model parameters

Service ID	Service Type	Share [%]	Priority
1	Audio streaming	7.5	High
2	Video streaming	64	High
3	Web	26	Low
4	File sharing	2.5	Low

Table 14 Traffic classes and their priorities

6.2.2 Resource Request Model

Let N_s be the set of services that can be consumed by wireless clients. Notice how, in this work we consider four types of data traffic services: audio streaming, video streaming, web and file sharing. Each service can have either high or low priority. High priority services should be preferably served by LTE eNBs while low priority services can be served by either LTE eNBs or Wi-Fi APs. As it will be clear in the evaluation section, the mix of services in the actual pool of resource requests is derived

considering the global mobile traffic forecast for 2018 found in [Cisco2017]. Table 14 summarizes the services considered in this work and their priorities. Wireless clients can simultaneously consume multiple services. Moreover, we also assume that wireless clients support multi–homing, i.e., different services of the same wireless client can be served by different RAN nodes. For example, a wireless client can watch a live video stream served by an LTE eNB while a file download is served by a Wi-Fi AP. Let N_u be the set of wireless clients. Each wireless client $u \in N_u$ is associated with a geographic location $loc(u)$, as *x, y* coordinates. Multiple weights $\omega_r^s(u) \in \mathbf{N}$ are assigned to each wireless client modelling the amount of resources (in bit/s) requested by the client $u \in N_u$ for the service $s \in N_s$. Table 15 summarizes the resource request model parameters.

Variable	Description
N_u	Wireless clients.
N_s	Available services.
$\omega_r^s(u)$	Resources requested by client $u \in N_u$ for service $s \in N_s$.
$loc(u)$	Geographical location of client $u \in N_u$.
$\Omega(u)$	Candidate RAN nodes for client $u \in N_u$.

Table 15 Resource request model parameters

6.2.3 Resource Allocation Model

Resource allocation in LTE and Wi-Fi networks is vastly different. LTE uses scheduled access and OFDMA while on the other hand Wi-Fi uses random access based on the CSMA/CA protocol. As a result, a homogeneous way of modelling available and used resources is needed before an optimization problem can be formulated. To this purpose, we introduce the *Equivalent Resource Utilization* ratio $ERU_n(\omega_r^s(u))$ defined as the fraction of radio resource required to support a service request $\omega_r^s(u)$. In the case of an LTE RAN, the ERU is computed as the fraction of Physical Resource Blocks (PRBs) required to support a given service at a given eNB. The number of PRBs N_{prb} in a subframe required to support a given request can be computed as follows:

$$N_{prb} = \frac{\omega_r^s(u)\zeta_{ap}T_{prb}}{N_{sbc}N_{ofdms}N_{modb}N_{ant}}$$

where T_{prb}, N_{sbc}, N_{ofdms}, and N_{ant} are, respectively, PRB duration (1ms), the number of subcarriers (12), the number of OFDM symbols per subcarrier (7), and the number of MIMO streams. Notice how these parameters are unequivocally defined for a given version of the LTE standard. ζ_{enb} is the PRB efficiency considering reference signals, synchronization signals, etc. and can be estimated at 1.25. N_{modb} is the number of modulated bits per symbol. For example, if a 64–QAM modulation is used, then $N_{modb} = 6$. Finally, the ERU for an LTE RAN can be computed as the ratio between N_{prb} and the total number of PRBs available in a cell. For example, in a 20 MHz cell there are 1000 PRBs in each 10ms–long Radio Frame. In the case of a Wi-Fi RAN, the ERU is computed as the fraction of the airtime required to support a given service at a given AP. Unlike LTE, Wi-Fi relies on random access with exponential back off as a channel access technique. Therefore, we decide to rely on a transactional model to estimate the time required to serve a given user. Wi-Fi uses a two–way handshake mechanism where each frame must be acknowledged by the receiver. As a result, for each data packet two frames must be exchanged on the air interface: one for the data itself and one for the Wi-Fi ACK. The time to deliver a frame is thus given by:

$$T_{airtime} = \left(T_{difs} + T_{data}\zeta_{ap} + T_{sifs} + T_{ack} + 2\sigma\right)$$

where σ, T_{difs}, T_{sifs}, and T_{ack} are, respectively, the propagation time (1μsec), the Distributed Interframe Space (34μsec), the Short Interframe Space (16μsec), and the time to send an ACK (24μsec). Notice how these parameters are unequivocally defined for a given version of the IEEE 802.11 standard. This model does not consider the random backoff period and the lost frames (e.g., due to collisions). Therefore, we introduce the Wi-Fi efficiency parameter ζ_{ap} to account for these impairments. In our simulation, ζ_{ap} has been set to 1.25. Assuming that data packets are encapsulated in maximum length

Ethernet frames (1500 bytes) and accounting also for the Wi-Fi header (36 bytes) and for the 6 tail bits that are added for each transmission, the time required to send a frame is:

$$T_{data} = T_{hdr} + \frac{(1500 + 36) * 8 + 6}{N_{sbc} N_{modb}}$$

where T_{hdr}, N_{sbc}, and N_{modb} are, respectively, the synchronization header (20μsec), the number of subcarriers (48), and the number of encoded bits per subcarrier. For example, assuming a 64–QAM modulation and a 3/4 coding rate, each subcarrier can encode 4.5 bits. Finally, the ERU for a Wi-Fi RAN can be computed as $T_{airtime}$ (in seconds) times the number of transactions necessary to support the request $\omega_r^s(u)$ over a unit of time. Notice how, in both the Wi-Fi and the LTE cases, an $ERU_n(\omega_r^s(u)) > 1$ means that the request $\omega_r^s(u)$ cannot be satisfied by RAN node n. Notice also how, the N_{modb} parameter present in both resource allocation models represents the number of modulated bits for each symbol/subcarrier. This quantity depends on the modulation and coding scheme (MCS) used for the transmission which in time is correlated with the channel quality between wireless terminals and RAN nodes. Several models linking channel quality and MCS can be found in literature [Du2016]. However, since the focus here is on the formulation of the user association problem, the selection of a particular channel model, although important, takes a secondary role. As a result, in the numerical evaluation we will leverage on a simple MCS estimation model which uses as input just the distance between transmitter and receiver.

6.3 Traffic Aware User Association

6.3.1 Problem formulation

To find the optimal assignment, we introduce the concept candidate RAN nodes $\Omega(u)$ for the wireless client $u \in N_u$ defined as the set of RAN nodes that have the wireless client u within their coverage radius:

$$\Omega(u) = \left\{ n \in N_n | dis(n, u) \leq \delta(n) \right\}$$

Notice that while in the numerical simulations the Euclidean distance between wireless clients and RAN nodes is used to identify the candidate RAN nodes for each wireless client, in the proof–of–concept implementation, the candidate RAN nodes are identified based on the signal strength. We can now provide the optimal ILP formulation for the user association problem. The objective of the ILP problem is to minimize ERU utilization in the network. The chosen objective function is:

$$min\Big(\sum_{n \in N_{enb}} \sum_{u \in N_u} \sum_{s \in N_s} ERU_n(\omega_r^s(u)) \xi_{enb}(u, s) \Phi_n^{u,s} + \\ + \sum_{n \in N_{ap}} \sum_{u \in N_u} \sum_{s \in N_s} ERU_n(\omega_r^s(u)) \Phi_n^{u,s} \Big) \quad (1)$$

The first argument of the objective function aims at minimizing ERU at eNBs. Whereas, the second argument minimizes ERU at APs. $\xi_{enb}(u, s)$ is a coefficient used in order to steer high–priority services toward eNBs. The coefficient $\xi_{enb}(u)$ takes values in (0, 1). When $\xi_{enb}(u, s) \rightarrow 0$ supporting services becomes progressively cheaper. Conversely, when $\xi_{enb}(s) \rightarrow 1$ the full cost of the service s must be sustained. The coefficients $\xi_{enb,ap}(u, s)$ is defined as follows:

$$\xi_{enb}(u, s) = \begin{cases} \alpha_{enb}(s) & \text{if } \frac{dis(u,n')}{dis(u,n'')} \leq \beta_{enb} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

where $n' \epsilon N_{enb}$ and $n'' \epsilon N_{ap}$ are, respectively, the closest eNB and the closest AP (according to the defined distance metric) to the wireless client u, and s is the service class. By tuning β_{enb} and $\alpha_{enb}(s)$ it is possible to steer the algorithm into assigning some service classes to eNBs or APs even when said nodes are not their optimal choice. Essentially, the ξ coefficients make placing a certain service at some RAN nodes cheaper from the radio resource utilization. Wireless clients can attach to RAN nodes if the RAN nodes have enough radio resource:

$$\sum_{u \in N_u} \sum_{s \in N_s} ERU_n(\omega_r^s(u)) \Phi_n^{u,s} \leq 1 \quad \forall n \in N_n \qquad (3)$$

where $\Phi^{u,s}{}_n$ is a binary mapping variable $\in \{0, 1\}$ that shows whether service $s \in N_s$ consumed by wireless client $u \in N_u$ is served by the RAN node $n \in N_n$. The following constraint makes sure that each wireless client $u \in N_u$ is associated with a RAN node that belongs to its list of candidates $\Omega(u)$:

$$\sum_{n \in N \setminus \Omega(u)} \Phi_n^u = 0 \quad \forall u \in N_u \qquad (4)$$

where $\Phi^u{}_n$ is a binary mapping variable that shows if the wireless client $m \in N_u$ is associated to the RAN node $n \in N$. Finally, the last constraint guarantees that each service $s \in N_s$ consumed by wireless client $m \in N_u$ is served by one and only one RAN node:

$$\sum_{n \in N} \Phi_n^{u,s} = 1 \quad \forall u \in N_u \quad \forall s \in N_s \qquad (5)$$

Notice that this formulation does not prevent that different services consumed by the same wireless client can be supplied by different RAN nodes

Algorithm 1 Traffic–aware User Association Heuristic.

```
1:  procedure HEU(N_enb, N_ap, N_u, N_s)
2:      for u ∈ N_u do                          ▷ List of wireless clients.
3:          for s ∈ N_s do                      ▷ List of services.
4:              candidates ← list()
5:              if ω_r^s(u) = 0 then
6:                  continue
7:              end if
8:              cls_ap_dis ← ∞
9:              for n ∈ N_ap do                 ▷ List of APs.
10:                 cost ← ERU_n(ω_r^s(u))
11:                 if dis(n,u) ≤ δ(n) and cost ≤ res(n) then
12:                     candidates(u,n) = cost
13:                     cls_ap_dis ← min(cls_ap_dis, dis(n,u))
14:                 end if
15:             end for
16:             for n ∈ N_enb do                ▷ List of eNBs.
17:                 cost ← ERU_n(ω_r^s(u))
18:                 if dis(n,u) ≤ δ(n) and cost ≤ res(n) then
19:                     if s is high_pr and dis(n,u)/cls_ap_dis ≤ β_enb then
20:                         cost ← α_enb(s) ERU_n(ω_r^s(u))
21:                     end if
22:                     candidates(u,n) = cost
23:                 end if
24:             end for
25:             mappings(u,s) ← arg min_n[candidates(u,n)]
26:         end for
27:     end for
28: end procedure
```

6.3.2 Heuristic

The ILP–based user association algorithm becomes computationally intractable when big networks with tens of thousands of users and hundreds of RAN nodes are considered. For example, associating 10000 users over a network composed of 200 eNBs and 400 Wi-Fi APs takes about one month using the ILP–based assignment algorithm. Conversely, the proposed heuristic can solve similar problems in just a few minutes. The pseudo code of the heuristic is reported in Algorithm-1 given above. Initially, for each service $s \in N_s$ of each wireless client $u \in N_u$, a list of candidate Wi-Fi APs (lines 9–15) are created, considering the client distance from the APs and the required resource availability at the APs, and the distance of the closest AP (i.e., the AP that provides the best channel condition to the client) is saved. The candidate list is then updated, adding also the candidate eNBs (lines 16–24). Notice that after finding valid eNB candidates by checking the candidacy condition (line 18), like the ILP–based algorithm, it is checked whether the distance of the candidate eNBs from the closest candidate AP is

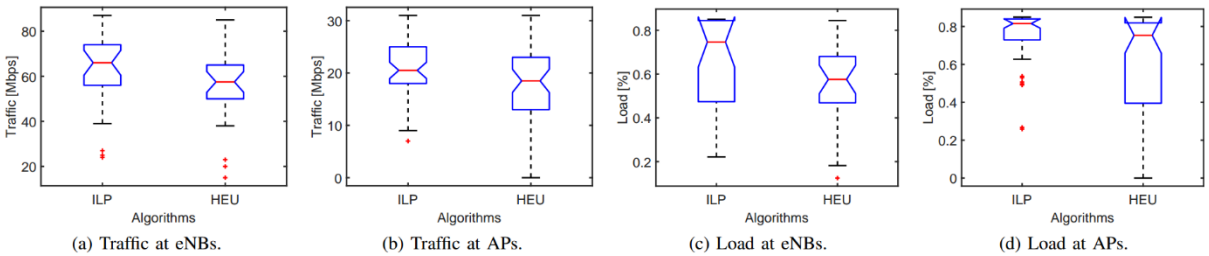
less or equal β_{enb} (line 19). If this condition is estimated true, the cost of the eNBs is reduced by $\alpha_{enb}(s)$ for the service $s \in N_s$, therefore, making the high–priority services cheaper to be supported by the eNBs rather than by APs. After having the complete list of candidates encompassing both the candidate APs and eNBs, the service $s \in N_s$ of the wireless client $u \in N_u$ is assigned the candidate node with the minimum cost. It is worthwhile to note that, since we consider a scenario in which wireless clients can employ multi–homing, different services of a same wireless client may be assigned to different RANs (e.g., the high–priority service of a single client may be supported by an eNB while the low–priority service by an AP), resulting in the client being simultaneously connected to and eNB and an AP

6.4 Numerical Evaluation

The goal of this section is to compare the performance of the ILP–based user association algorithm (ILP) with the performance of the proposed heuristic (HEU). The tuning parameters are selected as α_{enb}= 0.5 and β_{enb} = 2. The rationale behind the selection of the values is to make supporting a high priority service (IDs 1 & 2) half as expensive on LTE eNBs when such eNB are less than twice as distant from the wireless client than an optimal Wi-Fi AP. In this section, we first describe the simulation environment and the performance metrics used in our study. Then, we report on the outcomes of the numerical simulations carried out in a discrete event simulator implemented in Matlab R.

6.4.1 Simulation Environment

The reference RAN used in this work is composed of 26 eNBs and 50 Wi-Fi 802:11n APs. The eNB distribution is derived from an operational LTE network that provides cellular coverage to 1 million people distributed over an area of 5km^2. Conversely, APs are randomly deployed within the same area. For simplicity, it is assumed that single sector (omni) cells are used for both eNBs and APs with a 2×2 MIMO configuration providing a coverage radius of, respectively, 500m and 200m. Wireless client association requests arrive sequentially in batches, and with each arrival, the algorithms re–associate all the clients to the network. Each batch consists of 5 wireless clients each of them consuming up to 2 services randomly picked among the set of services found in Table 14. The actual traffic demand for each service is derived from the global mobile traffic forecast for 2018 [Cisco2017].

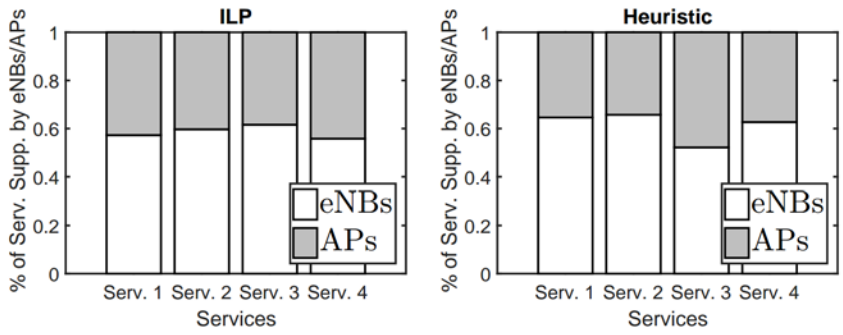


(a) Traffic at eNBs. (b) Traffic at APs. (c) Load at eNBs. (d) Load at APs.

Figure 44 Traffic and load distributions at eNBs and Aps after 150 association requests for all algorithms

6.4.2 Simulation Results

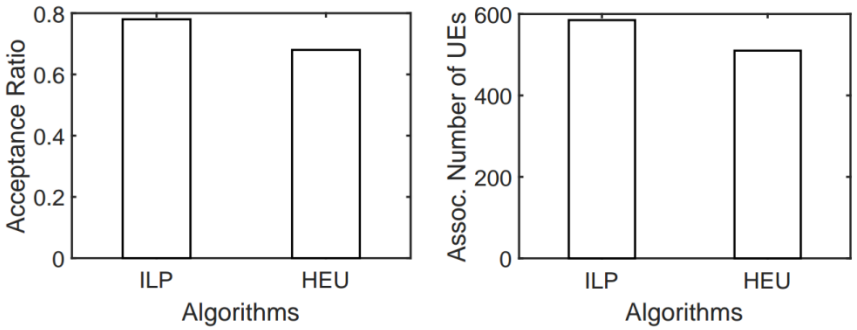
Figure 44 (a) and (b) plot the distribution of the aggregated traffic served by, respectively, eNBs and APs after 150 association requests. As it can be seen, the median of the traffic served by the eNBs is roughly three times the median of the traffic served by the APs for all the algorithms. This is since: (i) the number of APs is twice the number of the eNBs, and (ii) the coverage of the eNBs is larger than the coverage of the APs. Figure 44 (a) shows that ILP–LB distributes the traffic more uniformly across the available eNBs while both ILP–WF and HEU distributions are characterized by more outliers and by a bigger delta between the first and the third quartiles. Conversely, in Figure 44 (b) it can be observed that the traffic distributions for the APs are more uniform. Figure 44 (c) and (d) plot the distribution of node load (as fraction of available radio resources) at, respectively, eNBs and APs after 150 associations. As expected, the median loads for both ILP–LB and ILP–WF are higher than the median load for HEU. This is since the ILP–based association algorithm can accept a higher number of requests. ILP–LB demonstrates the best load–balancing capabilities since it has the narrowest load variation range across all eNBs with the median load of 0.7 at eNBs and 0.8 at APs.



(a) Services supported by eNBs. (b) Services supported by APs.

Figure 45 Distribution of the 4 possible service classes across both eNBs and APs using the different user association algorithms

Figure 45 shows the distribution of the 4 possible service classes across both eNBs and APs using the different user association algorithms. As it can be observed that a higher fraction of services is served by eNBs irrespective of the association algorithm used. This behavior is even more evident in the case of HEU where the fraction of higher priority service classes assigned to eNB reaches approximately the 70%. The rationale behind this is that, although there are 2 APs for each eNBs, the APs coverage is smaller which in time results in less opportunities for services to be served by APs. Moreover, the stations demanding high–priority services have more opportunities to be supported by eNBs.



(a) Acceptance ratio. (b) Associated number of UEs.

Figure 46 Acceptance ratio and total number of served clients

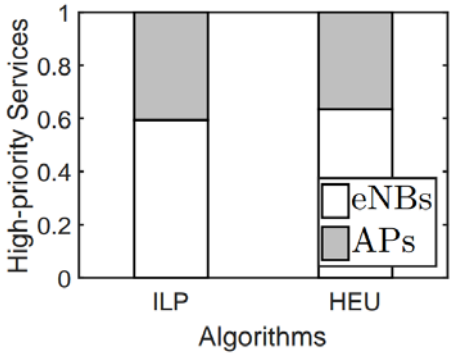


Figure 47 Share of high–priority services at eNBs and APs

The overall acceptance ratio and total number of associated users is plotted in Figure 46 (a) and (b) respectively. As expected, the ILP–LB and the ILP–WF accept the exact same number of wireless clients, approximately 80% of the total association requests. The HEU acceptance ratio is about 60%. Ideally, an access technology (i.e., an eNB or a Wi-Fi AP) for a client should be (re)selected only if the access technology satisfies the service QoS requirements. In our scenario, audio streaming (Service 1) and video (Service 2) are high–priority services. Figure 47 shows the traffic share of the high–priority services at eNBs and APs. Although the number of Wi-Fi APs are twice the number of LTE eNBs, we can observe that the traffic share of high–priority services at eNBs is greater that the traffic share at APs for all algorithms. This is because if a certain condition is satisfied (see eq. (2)), the high–priority

services are cheaper to be supported by eNBs. In essence, this is the effect of using the multi–homing feature of clients on the condition that the QoS of clients is not compromised as a result of low–priority services of clients being provided by Wi-Fi APs. The higher acceptance ratio for the ILP algorithm comes at the expense of an increased execution time. An optimal association can be computed by ILP–LB for the 150 group requests each composed by 5 wireless clients, for a total of 750 clients, in 250.34 seconds. Conversely, the heuristic can perform the association in 0.07, approximately four orders of magnitude less time.

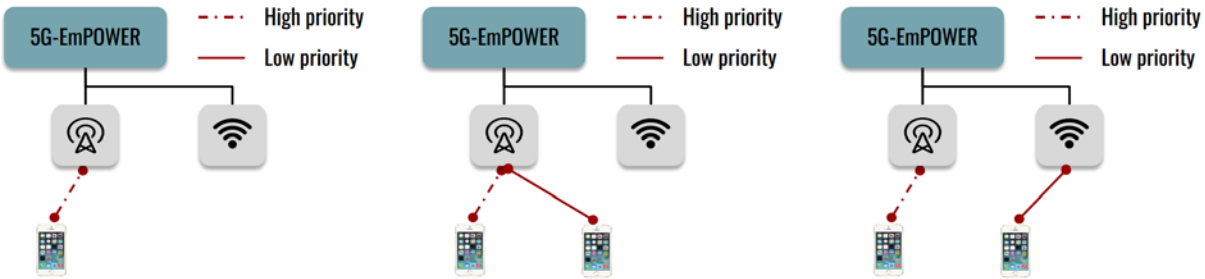
6.5 Proof of concept

6.5.1 Overview

The proposed traffic-aware user association solution proposed in this section been implemented and tested on top of the COHERENT C3 and in particular on top of its reference implementation 5G–EmPOWER [Riggio2015]. It is worth pointing out that the goal of this section is not to demonstrate the algorithm scalability in a large-scale setup, as a matter of fact our deployment consists of one eNB, one AP and two wireless clients. Instead we want to report on a preliminary proof–of–concept implementation of the proposed solution for heterogeneous RANs. To the best of the authors' knowledge this is the first real–world open–source SDN platform supporting mobility management applications over heterogeneous Wi–Fi/LTE RANs.

6.5.2 Evaluation Methodology

The testbed setup consists of one LTE eNB and one Wi–Fi AP. The LTE eNB is based on the Ettus SDR B210 platform and runs the srsLTE software stack [SRS2018]. The Wi-Fi is a commercial 802.11n wireless router running a modified version of OpenWRT [WRT2018]. Two standard smartphones are used as wireless clients. The 5G–EmPOWER MEC OS as well as the wireless client association heuristic run on a dedicated laptop. The overall network setup is sketched in Figure 48.

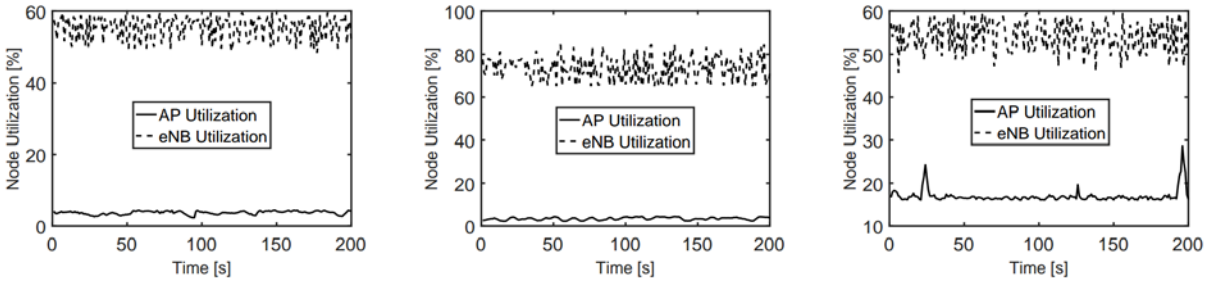


(a) Initial setup. (b) Second client attaches. (c) User re–association.

Figure 48 Testbed setup. Initially only one client is active with a single high–priority flow served by the LTE eNB. When the second client attaches to the same eNB and starts generating a low–priority flow, the controllers detects a handover opportunity and associates the low–priority user to Wi-Fi AP.

6.5.3 Results

Initially, only one wireless client is active and one high–priority video stream with a constant bitrate of 10Mbps is sent in the downlink direction from the LTE eNB (see Figure 48 (a)) to the wireless client. As it can be seen in Figure 49 (a), in this case the eNB utilization is approximately 50% whereas, the utilization of the AP is negligible (the non–zero utilization is due to the beacon frames that are transmitted periodically by the AP to announce its presence). Then, another wireless client is attached to the LTE eNB and a low–priority traffic stream with an average bitrate of 2 Mbps is sent in downlink direction (see Figure 49 (b)). As it can be seen, the eNB is almost saturated with a utilization of approximately 80%. As for the previous case the AP utilization is still negligible. At this point a handover decision is taken by the heuristic and the second client is moved to Wi-Fi (see Figure 48 (c)). As it can be seen in Figure 49 (c), after the offload is executed by the controller the eNB utilization decreases providing more opportunities for the new clients to be associated with the eNB.



(a) Node utilization (Scenario 1). (b) Node utilization (Scenario 2). (c) Node utilization (Scenario 3).

Figure 49 The LTE eNB and the Wi-Fi AP cell utilization for the various scenarios.

6.6 Conclusions

6.6.1 Technical/Scientific Impact

Traffic-aware user association and multi-homing are two promising ways of exploiting the radio resources available in a heterogeneous LTE/Wi-Fi RAN. We tackled this challenge by presenting a novel formulation of the user association problem for heterogeneous LTE/Wi-Fi RANs. Our formulation builds upon a radio access technology agnostic resource request model and accounts for different traffic classes. Moreover, the problem formulation also supports wireless clients multi-homing allowing different streams from the same wireless client to be served by different RAN nodes. An ILP formulation of the user association problem was compared with a scalable heuristic. We also reported on a preliminary proof-of-concept implementation of the proposed solution and on its validation over a small-scale testbed.

6.6.2 Feedback toward development

Going forward, we want to extend the problem formulation to the wired backhaul and to consider clients' mobility using more realistic channel models. We also plan to study the system performances using real traffic traces coming from an operational network and to inject such information into the traffic-aware use association algorithms.

6.6.3 Expected business impact

Wi-Fi networks are known to be a cost–efficient traffic offloading solution for mobile networks. The solution presented in this section demonstrates that the COHERENT SDK can be used to implement practical traffic offloading solution in heterogeneous Wi-Fi/LTE radio access networks. To the best of our knowledge this is the first open platform and programmable platform capable of supporting such capabilities. It is worth also noticing that this traffic offloading feature will also be leveraged in the 5G-PPP Phase 2 project 5G-ESSENCE.

7. Algorithms for LTE RAN Slicing

7.1 Overview

Network slicing is one key pillar of providing the flexibility and is highlighted by ITU, 3GPP, and NGMN, where the objective is to build a novel network architecture that should support not only classical mobile broadband applications and services, but also vertical industry (e.g., automotive systems, smart grid, public safety, etc.) and Internet of Things (IoT) services. A network slice can either be isolated from the others down to the different sets of spectrum and/or cell site or be shared across different types of resource like radio spectrum and network functions (e.g., (sub-)layers of protocol stack), or be customized for a subset of processing with the access to a portion of virtualized radio resources (e.g., radio resource abstraction). To this end, network softwarization and virtualization are keys to flexibly customize each slice and ease the network function development to accommodate end-to-end service requirements. They constitute the foundation of a multi-service architecture, and are realized by adapting software-define networking (SDN), network function virtualization (NFV), and cloud computing principles [Rost2016]. So far, most of the devised network architectures [Samdanis2016] [Zhou2016] share the same principle, with some differences on the way to instantiate and deploy a network Slice. Mainly, a Global Slice Orchestrator is proposed on top of NFV-like architecture, which translates the Slice provider requests by selecting the appropriate Virtual Network Functions (VNF) (e.g., core network functions, firewall, and deep packet inspection) along with their service graph, which specifies how logically these VNFs are connected. Then, the VNFs are deployed over the distributed cloud using a Virtual Infrastructure Manager (VIM) and SDN rules to interconnect them. Each slice might include its own SDN controller to manage the intra-slice traffic. Resources (i.e., infrastructure, radio spectrum, and transport network) may belong to the same administrative or to different domains; the latter case requires a multi-domain orchestrator.

In particular, both radio access network (RAN) and core network (CN) are targeting for slicing in mobile network and several architecture and prototypes have been proposed for CN slicing [Taleb2015], [Qazi2017] and RAN slicing [Ksentini2017], [Foukas2017]. The challenge of CN slicing are also addressed by 3GPP, and realized through the evolved dedicated core network (eDECOR) [3GPP TR 23.711]. Nevertheless, RAN slicing remains challenging in (1) providing different levels of isolation and sharing to allow a slice owner to customize its service processing across different planes while (2) increasing the radio resource utilization and efficiency to enhance the multiplexing gain. Depending on the level of resource isolation, we may mention: dedicated resources and shared resources models. In the dedicated resource model, the RAN slice is built by separating and isolating slices in terms of: control and user plane traffic, MAC scheduler and physical resources. Each slice has access to its own RRC/RLC/PDCP/MAC instances, and the physical resources are strictly dedicated to a specific slice, e.g., a percentage of Physical Resource Blocks (PRBs) is dedicated to each slice, or a subset of the channel is dedicated to each slice. Although dedicated resource model ensures committed elementary resources to the slice, it reduces the slice elasticity as well as scalability, and limits the multiplexing gain. Indeed, using the dedicated resource model does not allow a slice owner to easily modify the amount of resource (i.e., PRB) committed to a slice during its life-cycle. Furthermore, dedicated resources model may lead to a waste of resources, as the PRBs are strictly dedicated to a slice, even if they are not used. The second approach, i.e., shared resources model allows the slice to share the same: control plane, MAC scheduler and physical resources. In this solution, the PRBs are managed by a common scheduler that distributes the PRB to slices' users according to different criteria, like Service Level Agreement (SLA), priority, etc. Whilst this solution exploits statistical scheduling of physical resources, which ensures more scalability and elasticity by report to the dedicated resources model, it may lack the support of strict QoS guarantee for Slices and traffic isolation.

To enable the RAN slicing concept, several 5G RAN design paradigms shall be fulfilled as elaborated in [Marsch2016]. 3GPP also mentions RAN slicing realization principles in [3GPP TR 38.801] such as RAN awareness slicing, QoS support, resource isolation, service level agreement (SLA) enforcement among the others. These principles can be enabled through the software-defined RAN (SD-RAN) concept [Gudipati2013] that is realized in [Foukas2016] as the FlexRAN platform implementing a customized RAN south-bound application programming interface (API) through which programmable control logics are enforced with different levels of centralization. To this end, multiple services over

RAN segment can flexible utilize available radio resource by means of wireless virtualization [Richart2016]. With aforementioned enablers, several RAN slicing works are initiated. The network virtualization substrate (NVS) in [Kokku2012] virtualizes radio resource for different resource provisioning approaches to allow several mobile virtual network operators (MVNOs) to coexist in a single physical RAN; however, it falls short in exploiting multiplexing gain. In [Rost2017], the radio resource scheduling is separated into intra-slice and inter-slice scheduler but without any resource abstraction/virtualization approach. The proposed RAN slicing architecture in [Ksentini2017] provides the virtualized resource blocks (vRBs) to enforce radio resource management at physical resource blocks (PRBs) level, but it cannot abstract resource per slice request in a customized manner. Authors of [Foukas2017] groups the PRBs into vRB groups through a set of abstractions and provides only relevant resource information to the corresponding slice. However, it only focuses on resource isolation without investigating any multiplexing benefits. In summary, an execution environment shall support various slice requirements (e.g., isolation) and elastically improve multiplexing benefits (e.g., sharing) in terms of the set of radio resource abstractions to serve various flavors of slice.

7.2 Network Slicing Architecture: Enforcing the slices

A global overview of the envisioned architecture is depicted in Figure 50. In this figure, we present the network architecture after that the Slice Orchestrator (SO) has instantiated the CN elements, using Physical Network Function (PNF) or VNF. Each CN instance includes a set of VNF or PNFs representing core network functions, such as mobility management, authentication as well as data plane forwarding functions (i.e., gateways to Internet). The CN instances are connected to the shared RAN using the classical S1 interfaces or the newly interfaces (i.e., Gx, Gy) proposed by the 3GPP SA 2 group in [3GPP TR 23.799]. The RAN (i.e., eNodeB) is able to steer the slice traffic to the correct CN instances using the concept of eDECOR, in which the UE indicates a slice ID that allows the eNodeB to select the appropriate CN elements for the UE traffic. The slice ID could be hard encoded in the UE (i.e., USIM) or encoded through the Public Land Mobile Network (PLMN). Moreover, the UE communicates the slice ID during the RRC connection procedure as well as in the Non-Access Stratum (NAS) procedure, which allows the eNodeB to contain the UE within the requested slice(s) and treat it according to the agreed SLA. For instance, use the appropriate MAC scheduler instance which will handle the slice resources to satisfy the required QoS. It is important to note that the slice ID should indicate the slice Type in addition to the Tenant ID. The Tenant ID is the entity that is in charge of the slice. Finally, the eNodeB maintains a mapping between the slice ID and the CN elements (i.e. IP addresses), which is communicated by the SO during the slice instantiation process.

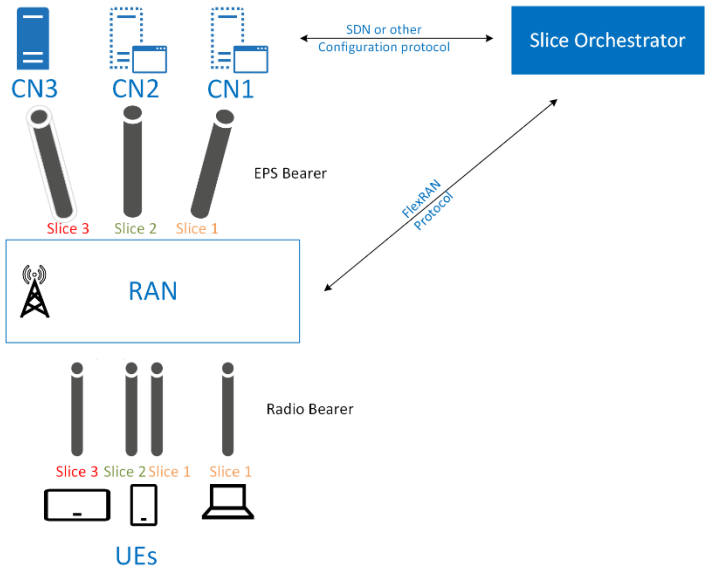


Figure 50 Global overview of the envisioned Network Slicing architecture

Figure 51 then depicts the architecture of the eNodeB when using our proposed network slicing solution. The proposed architecture shares many concepts with the legacy LTE architecture, particularly the usage of logical channel and their mapping to Evolved Packet System (EPS) bearers. The main difference is related to the abstraction (i.e., virtualization) of the physical resource blocks, where an abstraction layer, named **Resource Mapper** (RM), is added. The latter acts as an interface between the shared PRB and the **Slice Resource Manager** (SRM). The SRM is in charge of scheduling resources for UEs belonging to its Slice and any scheduling algorithm (e.g., Proportional Fair, Round Robin, or Priority-based) could be used. Each SRM may use a different scheduler, as configured by the SO. The RM will expose the information to each SRM regarding: (i) the number of packets, per UE and per logical channel, waiting for transmissions, in both UL and DL directions; (ii) the Channel Quality Indicator (CQI) of each UE; and optionally the latency of the oldest packet in the UE's queue as well as the history and statistics of UE traffic. This information will be used by the SRM to schedule UEs over the vRBs. These vRBs should be mapped to PRBs, and giving that the number of PRBs is limited and dependent of the physical characteristics (i.e., throughput), not all the UEs will be served during the Transmission Time Interval (TTI). The RM will be in charge of accommodating the vRBs to PRBs according to the amount of resources that should be allowed to each slice, e.g., slice dedicated bandwidth (SDB). The SDB is the policy enforced by the SO to the RM when the slice is firstly created. It is worth noting that the SDB is dynamic; it could be adapted as a function of workload demand and slice requirements upon a request from the Slice. Moreover, the SDB policy can be expressed in terms of percentage of PRB or the bandwidth to be allocated to a slice among the others.

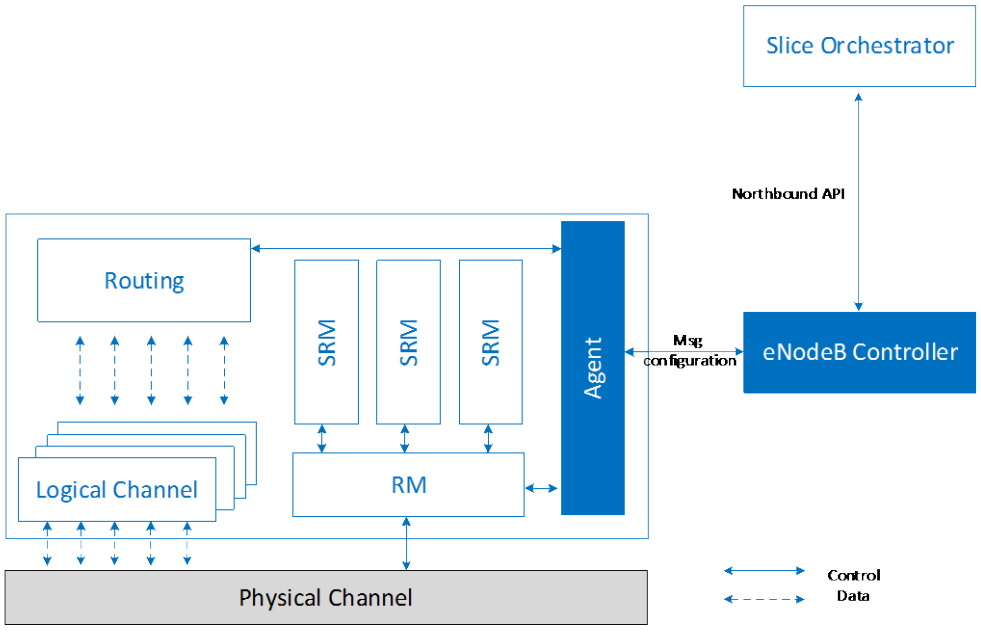


Figure 51 An overview of the eNodeB functions enforcing network slices at the RAN.

7.3 Resource Virtualization

As stated earlier, we introduce the resource virtualization at RM layer for the mapping between virtualized resources and physical ones to each UE via exploiting as much as possible the SRM scheduling output. Accordingly, we partitioned the MAC operation into two levels, the SRM performs the first level by ensuring intra-slice traffic scheduling, while the RM assigns PRBs to UEs according to: the mapping provided by each active SRM, the SDB policy, the actual channel state (i.e., the available PRB) and slice priority. The proposed two-level scheduling is preferred over the jointly scheduling (all in one pass), as the latter is very complex and requires multi-dimensional scheduling. Indeed, this type of scheduling algorithms formulates a multi-objective function that should satisfy

heterogeneous slice requirements (e.g., latency and bandwidth), where the optimal solution is usually NP-hard. In the following, we elaborate on the resource virtualization approach, which is one key concept to provide the required level of isolation and sharing to each slice. More specifically, it performs partitioning on radio resources based on the separated slice requirements, abstraction of physical resources to/from the virtualized ones, and revealing the virtual views to a slice that is customized and decoupled form the exact physical ones.

7.3.1 Inter-slice Resource Partitioning

Resource partitioning is a periodic process that happens every allocation window of T, and it distributes radio resources among multiple slices based on their requirements expressed in the slice context. Such requirement contains three elements: (1) *resources type* that defines whether the requested resources are of type physical/virtual radio resources in time and frequency domains, or capacity in terms of rate, (2) *abstraction type* that maps physical radio resource allocation types, namely fixed position, contiguous, non-contiguous, or minimum resource block groups (RBGs), to virtual RBGs (vRBG) or virtual transport block size (vTBS), and (3) *resource structure* that contains the applicable frame structure numerologies in time and frequency domains. More specifically, different numerologies in terms of TTI and Sub-Carrier Spacing (SCS) can be applied depending on the deployed frequency band and/or maximum user mobility to mitigate the impact of wireless channel non-idealities (e.g., frequency offset due to Doppler shift). For instance, only one type of SCS, i.e., 15 kHz, is applied in LTE system, while there are five applicable SCSs, i.e., 15, 30, 60, 120, and 240 kHz, defined by 3GPP in [3GPP TS 38.211] with their corresponding frame structures.

Besides aforementioned radio resource requirement provided by the slice owner, the resource partitioning shall also respect the policy defined by the infrastructure provider, for instance, the allowable resource allocation type of underlying radio access technologies (RATs). Take the downlink (DL) resource allocation of LTE system for instance, there are three types of resource allocation: (1) Type 0 allocation is based on the minimum granularity as RBG that comprises multiple RBs, (2) Type 1 categorizes RBGs into several subsets and only allocates RBs within the same subset, and (3) Type 2 allocates contiguous virtual RBs (vRBs) that can be physically contiguous (localized vRB) or non-contiguous (distributed vRB). For uplink (UL), there are two resource allocation types: (a) Type 0 is contiguous and similar to Type 2 of DL, and (b) Type 1 allocates non-contiguous RBG with two distinct clusters. Then, four resource abstraction types and their respective mapping to the DL/UL resource allocation types are identified as in Table 16. Note the proposed vRBG and vTBS are a superset of legacy resource allocation types and provide the required flexibility not only for intra-slice resource allocation, but also inter-slice resource partitioning.

Table 16 Mapping between resource abstraction and allocation type.

Requested resources	Abstraction types (Resource granularity)	DL Resource allocation type	UL Resource allocation type
Resource block	vRBG Type 0 (Non-contiguous)	Type 0, Type 1, Type 2 distributed	Type 1
	vRBG Type 1 (Contiguous)	Type 0, Type 2 localized	Type 0
	vRBG Type 2 (Fixed position allocation)	Type 2 localized	Type 0
Capacity	vTBS Type 0 (RBGs with min granularity)	All Types	All Types

The inter-slice resource partitioning can be done via extending some heuristic resource embedding algorithms (e.g., [Yang2012] [Belt2014]) to support various resource granularities and several SCSs. As mentioned earlier, such optimization can be mapped to the 2-dimension Knapsack problem with NP-hard complexity that can be solved sequentially. We hereby provide an algorithm in following with polynomial time complexity that can efficiently deal with the problem. Such partitioning algorithm follows the pre-defined priority levels for different slices that is prescribed in the slice context. Last but not least, such partitioning has two complementary goals: (a) satisfy as many slice resource request as possible, and (b) centralize the unallocated region for further multiplexing (e.g., shared with other slices or transport control information/ cell broadcast message).

Algorithm 3 Inter-slice Resource Partition Algorithm

Step 1. Prioritize the k-th slice, i.e., s_k, in the slice list **S** based on some pre-defined policy with its radio resource requirement including: (a) SCS_k comprises the applicable SCSs, (b) T_k and N_k are the number of requested resource in time (millisecond) and frequency domain (Hz) respectively, and (c) g_k is the granularity which can be contiguous, non-contiguous, fixed position or minimum granularity as the summary in Table 16.

Step 2. Based on its granularity, i.e., g_k, different embedding algorithms are applied aiming to find the place to successfully embed the requested resources (i.e., T_k, N_k, SCS_k) into the resource grid with the largest unallocated region in rectangular form. Such region represents the multiplexing benefits.

Step 3. Remove the k-th slice from the slice list **S** and go back to step 1 if there are some slice left in **S**.

7.3.2 Radio Resource Abstraction

Resource abstraction serves for two main purposes: (1) isolate resources by presenting a virtual view of the resources that is decoupled from its exact physical location, and (2) increase multiplexing gain by adjusting allocation types while sharing unused resources. The former simplifies the intra-slice resource scheduling operation and prevents other slices to access or even infer the resources allocated to others (in favor of slice owner), and the latter allows to increase the resource utilization efficiency (in favor of infrastructure provider). Take the 3 MHz case of LTE as an example in Figure 52, where there are 15 PRBs and PRBG granularity is 2 PRBs, giving a total of 8 PRBGs partitioned among 4 slices. These PRBGs are partitioned for each slice based on the number of required resources and the resource granularity stated in Table 16. Then, they are virtualized into vRBGs based on the abstraction type. For instance, fixed position resources is requested by slice 1 and hence no virtualization is performed (i.e., PRBG). In contrast, slice 4 only requests a capacity, and thus its PRBGs are abstracted into vTBS with the corresponding capacity. PRBGS of slice 2 and 3 are virtualized into vRBGs via abstracting the exact frequency/time locations and dimensions; and are pooled together to maintain their relative frequency dependencies among virtualized resources but without revealing their absolute physical frequency dependencies. Take the slice 3 that uses resource allocation type 1 as an example, only PRBGs within the same subset can be scheduled at the same time. In that sense, vRBGs are pooled to indicate an exclusive condition between vRBG pool 1 (i.e. PRBG0, PRBG6) and vRBG pool2 (i.e. PRBG5). Hence, the intra-slice scheduler of slice 3 will allocate resources to each user from either vRBG pool 1 or pool 2.

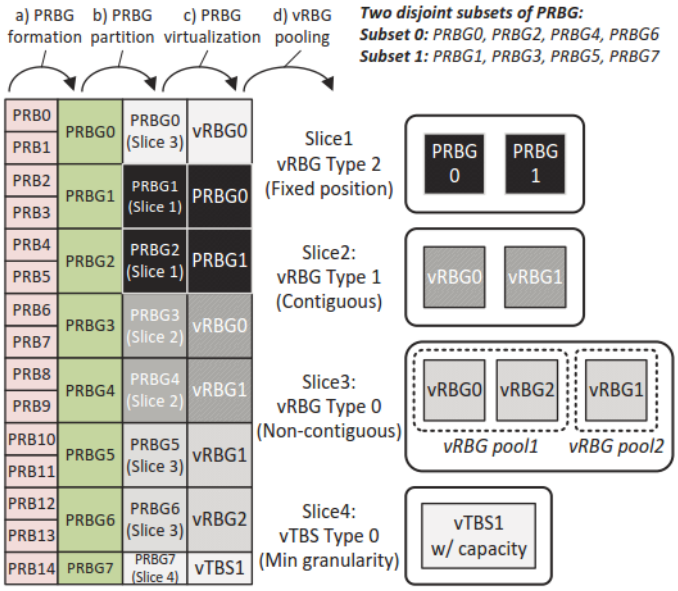


Figure 52 Stages form vRBG and vRBG pool

7.3.3 Radio resource Accommodation and Multiplexing

After the radio resource partitioning and virtualization, each slice can perform intra-slice resource scheduling to users and the scheduling decision will be accommodated into PRBs as shown in Figure 53. Such intra-slice resource scheduling consists in scheduling the intra-slice UEs traffic, by assigning virtualized resource to UEs. These vRBGs/vTBSs are virtual and do not have any link to the available PRB (except for the fixed position granularity). The scheduling algorithm is configured by the SO when the slice is instantiated. Depending on the Slice type, the SRM functions and the needed inputs (from the RM) will be different. In following, we take the 3 most common slice types for instance, i.e., Extreme Mobile Broadband (xMBB), Ultra-reliable and low-latency communications (uRLLC), and Massive Machine Type Communication (mMTC).

- **xMMB type slice**: For this type of slice, the used scheduler could be the popular PF algorithm. In this case, the intra-slice scheduling algorithm requires as inputs: the list of UE with their workload waiting to be scheduled and the UE's CQI. The algorithm will produce, per UE, the number of the needed vRBGs along with the Modulation and Coding Scheme (MCS) to be used per vRB. The proposed scheduling list should be sorted according to UE priority. Indeed, due to resources limitation (caused by physical or SDB limitation), the RM may not schedule the entire list of UE provided by the SRM. For instance, the priority may be based on the difference with the target throughput for a UE. Higher is this value, higher is the priority of the UE.

- **uRLLC type slice**: For this kind of slice, the used scheduler should consider two important criteria. The first one is the latency, which should be minimized (i.e., use a delay-based scheduler). The second one is the service reliability. To maximize the latter, the MCS to be used by UEs should be very robust to channel variabilities; i.e. robust modulations are favored over high data rate modulations. Therefore, the SRM requires the list of UE, and for each UE, the latency experienced by the head-line packet of the logical channel. The outputs will be the mapping vRBG/UE to be scheduled as well the MCS. Note that, the SRM provides also the priority of UEs according to the remaining time before reaching the deadline. Lower is this value, higher is the priority of UE.

- **mMTC type slice**: This type of slice is very special, as it involves more UL traffic than DL traffic. Indeed, since the DL traffic is not very important, a simple scheduler like round-robin or proportional fairness may be used. But for the UL traffic, we may distinguish between periodic update (i.e., MTC are activated during a predefined time interval) and event-driven MTC traffic pattern. For the periodic update, we propose to use a pre-fixed scheduling (e.g., Semi Persistent

Scheduling). In this case, the SO should indicate when the MTC will be triggered by the application to send reports. By consequence, the RM will dedicate specific resources to the MTC devices during the activation period, avoiding high contention on the channel, particularly during the Physical Random Access Channel (PRACH) procedure. It is important to note that a new RRC state needs to be introduced, which allows keeping the radio resources dedicated to a UE active (i.e., C-RNTI, radio bearer, etc.); even if the UE is inactive (i.e. RRC connected, UE inactive). The PRB schedule will be included directly by the RM in the Physical Downlink Control Channel (PDCCH). However, in case of event-driven MTC traffic pattern, the UL also should be handled by the SRM. In this case, the scheduling algorithm depends on the type of channel access (e.g. regular, random or contention access). For regular channel access, any simple algorithm like RR is sufficient, since this type of applications neither requires ultra-low latency communications nor high bandwidth, whereas for content-based access, a group-based PF may be applied.

Such accommodation does not necessary follow the mapping done in the partitioning stage (cf. Figure 52) in order to better utilize available resources. For instance, vRBG1 of slice 2 and slice 3 are accommodated to their vRBG0 in partitioning stage respectively so as to have a larger contiguous unallocated region (i.e., PRBG4 to PRBG6) that can be mapped to any SCSs for further resource sharing. The unallocated region can be multiplexed by other slices (e.g., vTBS2 of slice 4) that request more resources. Moreover, the preemption can also be done by removing the inter-slice scheduling results of other low-priority slices to boost the performance of high-priority slices perceived performance. Finally, PRBGs can be mapped and the corresponding control information (CI) are formed. Note these CIs are used to indicate the user about the positions of allocated PRBs as well as necessary physical layer information (e.g., modulation and coding scheme (MCS), new data indication, HARQ process indication, etc.) for successful reception or transmission. With limited control region for CI transportation, the unallocated resources can also be leveraged to carry CIs.

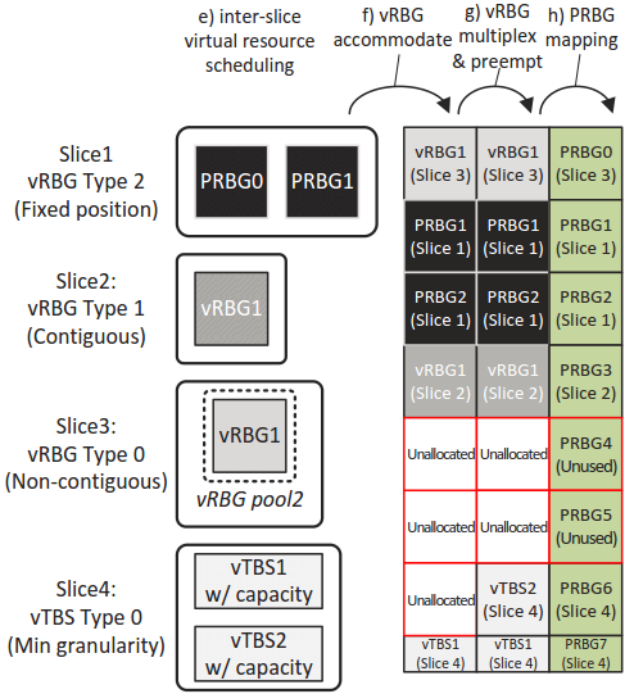


Figure 53 Stages accommodate vRBG to PRBG

As noted, the resource accommodation is performed by mapping the scheduled virtualized resources into the physical ones; however, there is no strictly 1-to-1 mapping between the virtualized and physical ones. Except for the slices with fixed-position granularity, such accommodation do not necessary map to the same physical resource done in the partitioning stage (cf. Figure 53). Moreover, our aim here is to increase the multiplexing benefits and/or satisfying new incoming slice requests. In this sense, the resource accommodation problem is close to the resource partitioning one and the same heuristic algorithm can be applied.

7.4 Resource slicing on LTE system

In this subsection, we validate the RAN slicing concept and explore in different use-cases relying the FlexRAN agent, which operates on the top of OpenAirInterface (OAI) platform that was introduced in [COHERENT D2.4]. To map a user to a slice and populate the slice context information, we used internal OAI user identities together with the radio network temporary identifier (RNTI) generated by the MAC layer to determine user slice identifier. Note that in a real deployment, a user communicates this information through RRC and NAS procedures. A template is used to describe slice resource requirements. In then following, we present the results of two considered use cases.

7.4.1 Radio Resource and control Logic Isolation

To demonstrate the isolation in terms of radio resource and per-slice control logic, we deploy three slices with three different traffic patterns, high bit rate with traffic variability for slice 1, medium bit rate for slice 2, and low bit rate with periodic traffic for slice 3. Each slice contains 5 users and different inters-slice resource partitioning policies are applied: a) **fair partitioning** allocates 33% of total resources to each slice, b) **greedy partitioning** allocates 60% of resources to slice 1, and 20% to slice 2 and 3, and c) **proportional partitioning** allocates 50% to slice1, 40% to slice 2, and 10% for slice 3. For the intra-slice scheduling, we apply a simple fair scheduling among users.

From the results presented in Figure 54, it can be observed that the aggregated good-put and average latency can significantly be changed based on the three different policies in box plot. As the corresponding traffic pattern, each slice shows different good-put and jitter even with the fair partitioning. For instance, slice 1 shows a higher good-put, while slice 3 with the fewest good-put as its low bit rate characteristic. In this sense, the greedy policy can better fit the traffic pattern of slice 1 via partitioning more resources then boosting its good-put, and the proportional one can further partition more resource for slice 2 with a fewer resource for slice 3 as its very low data rate.

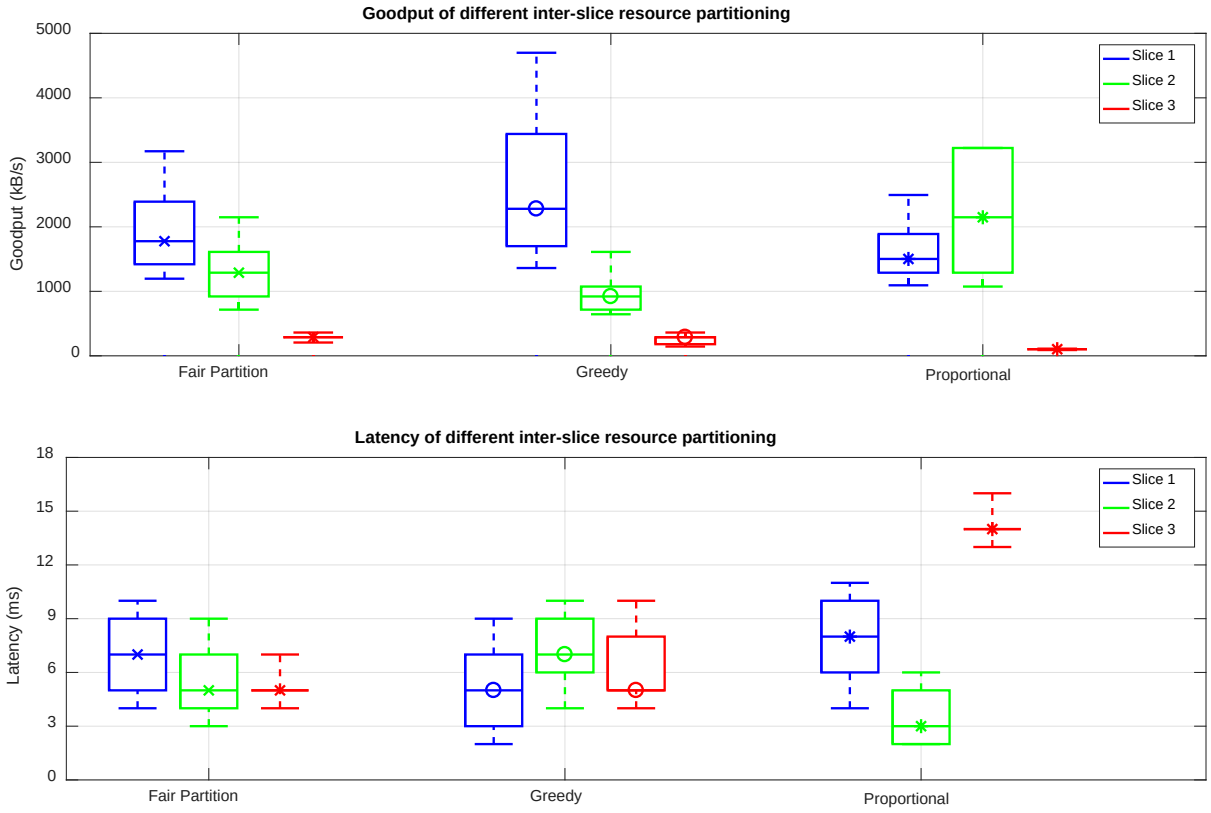


Figure 54 Aggregated good-put and average latency of three different slices

In Figure 55, we present the result of per-user average good-put and latency, we can see that the intra-slice scheduling control logic will not be impacted by different inter-slice resource partitioning, i.e., the

fairness is shown between user 1 to user 5 of slice 1, user 6 to user 10 of slice 2, and user 11 to user 15 of slice 3. This confirms the capability of providing the performance isolation among slices. Moreover, it also indicate that the inter-slice and intra-slice processing can be devised separately to match separated roles as slice owner (i.e., intra-slice processing) and infrastructure provider (i.e., inter-slice processing).

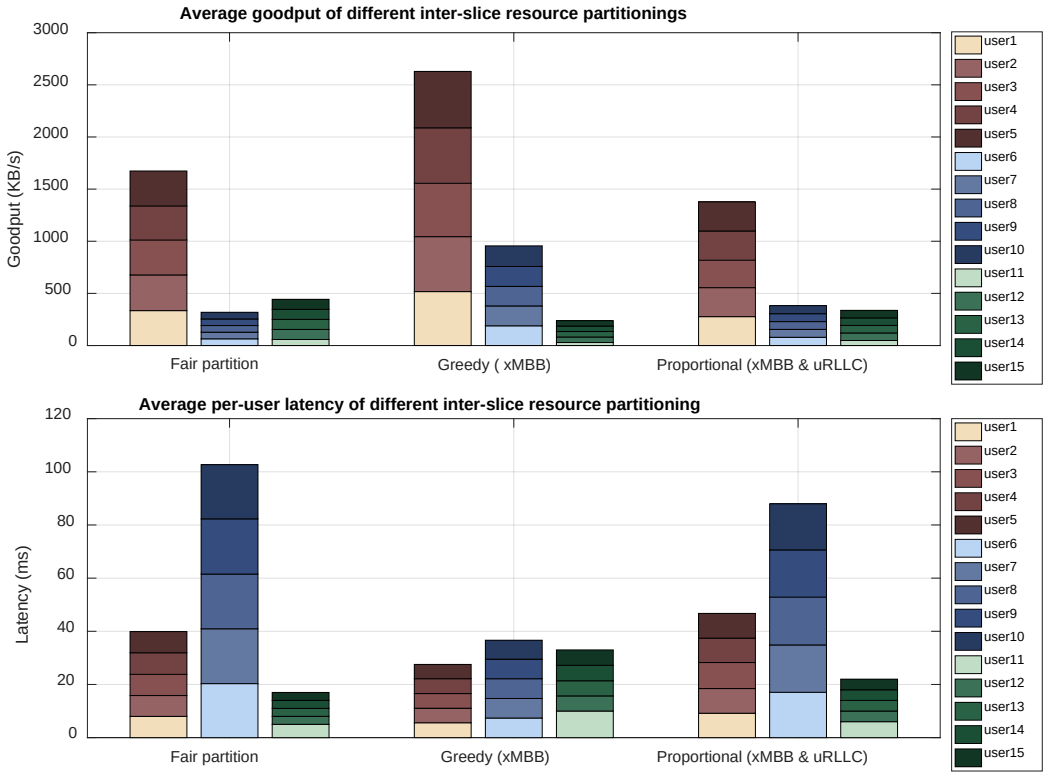


Figure 55 Per-user average good-put and latency of three different slices

7.4.2 Dynamic inter-slice resource partitioning

Figure 56 shows the logical setup of the conducted experiment with two distinct slices associated separately with two different users. Such setup include following entities: commercial of-the-shelf (COTS) UE, OAI RAN, OAI CN, real-time controller and the policy application that is elaborated in [COHERENT D2.4]. The eNB is configured for band 7 (2.6GHz, FDD) with 5 MHz channel bandwidth in SISO mode and there are three quad-core PCs used for different entities: PC1 for OAI eNB, PC2 for OAI CN, and PC3 for the FlexRAN controller and collocated control application. We utilize 2 COTS UE to measure the experienced end-user performance when dynamically changing the policy of resource slicing.

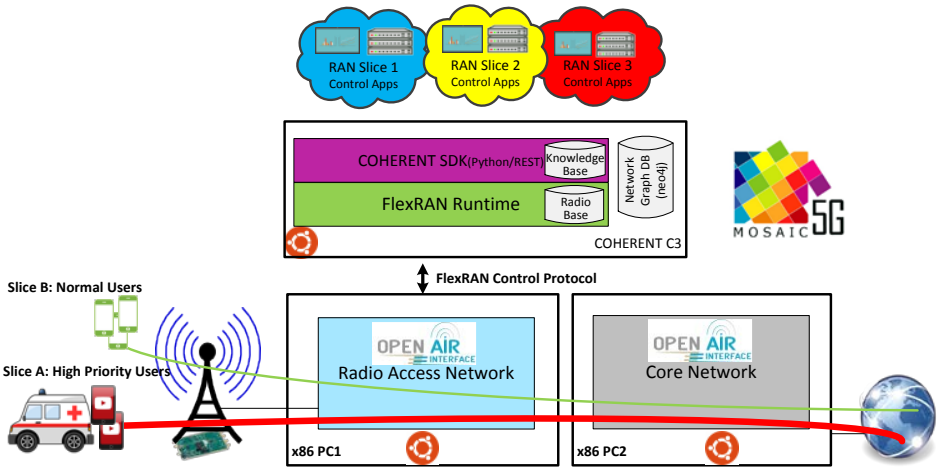


Figure 56 Proof-of-Concept experiment setup

Firstly, we examine the per-slice policy enforcement on the frequency domain resource block allocation (i.e., physical resource block (PRB) in LTE) in following two cases:

1. **Fair slice policy**: It refers to the case that each slice has the same priority and share the same portion of all available PRBs in frequency domain. Hence, it is denoted as 50/50 policy in following.
2. **Prioritized slice policy**: In contrast, different priorities are put into practice on different slices. For instance, the first slices are more favored[3] and can be allocated with 80% of all available resources, while another slice only receive the rest 20% of resources. It is denoted as 20/80 policy in following result.

In Figure 57, we show the result when applying these two policies over time through the policy enforcement control app (see [COHERENT D2.4]) at different time, i.e., from 50/50 policy to 20/80 policy. Such result is measured at the application-layer in terms of the good-put and delay jitter. In the first 45 second, we apply the 50/50 policy, and after this time, then we change it into 20/80 policy in correspondence to prioritize second slice as well as its associated user. It can be observed that the good-put is significantly changed after applying the prioritized slice policy. Further, the delay jitter is also impacted due to a low-prioritized slice only uses fewer resource and increases application-layer delay variance. However, for the slice 2, the delay jitter is significantly reduced. Such result justify that the developed RAN resource slicing scheme can be utilized by control application to better shape their requirement. Moreover, it can also be utilized in accompany with changing other RAN parameters, such as MCS, to better matching to the slice request at RAN segment. For instance, the xMBB type slice can allocate MSC in terms of the extra positive offsets based on the reported wideband channel quality indicator (CQI) in order to further enhance the data throughput but may increase the number of re-transmissions. In contrast, the uRLLC type slice can adopt a more conservative MCS index based on the reported CQI value. Further, the UE-selected sub-band CQI report can be applied to increase the reliability and reduce the delay variation due to fewer re-transmissions.

[3] Such prioritized slice policy can be managed in a dynamic manner as comparing the priority of current active slices.

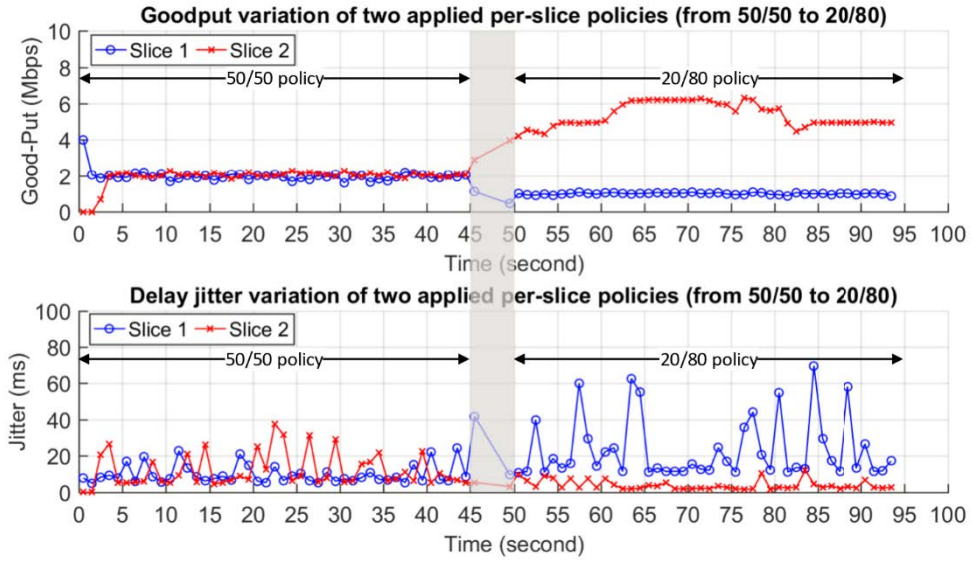


Figure 57 Impact of slice-specific policy reconfiguration on the performance

7.5 Conclusions

7.5.1 Technical/Scientific Impact

In this section, we provide a fully programmable RAN slicing architecture to enforce resource slicing in the time and frequency domain. It enables the two-level MAC scheduler to abstract and share the physical resources among slices via resource virtualization. Four different abstraction types are provided in order to satisfy different resource requirements in a multi-service environment. Such proposed two-level approach, i.e., inter-slice and intra-slice, can leverage the known scheduling algorithms to be flexibly applied for intra-slice SRM scheduling results to serve the traffic characteristic of each service. Whereas a unified inter-slice partitioning and accommodation shall fit as more resource requests as possible and take the multiplexing benefits into account when embedding the requested resource into the resource grid. Such multiplexing can not only enable the sharing between several slices but also enable dense service deployment at a single RAN infrastructure.

7.5.2 Feedback toward development

Within the development, we firstly utilize the FlexRAN and OpenaAirInterface platform introduced in [COHERENT D2.4] to conduct several experiments for examining two characteristics: control logic isolation and dynamic resource slicing. The former one shows that the intra-slice scheduling control logic is not impacted by different inter-slice partitioning policies. Hence, such two-level MAC scheduler is feasible for multi-service execution environment. Whereas the latter one leverages the control application also elaborated in [COHERENT D2.4] to dynamically adjust the inter-slice resource partitioning policies. Such experiment shows that the resource slicing can be done in a real-time manner to dynamically serve the needs of the slice (with attached users). It can also be combined with other control applications to also adjust the MCS mapping, carrier frequency, antenna space for flexibly adjustment.

7.5.3 Expected business impact

The envisioned RAN slicing architecture and resource slicing abstraction types pave the way for slice customization and ensure the slice owner can individually shape its control logics. It will naturally enable the decoupling of infrastructure provider, network function provider and slice owner in service-oriented 5G vision. Such relations between different providers transform the value-chain in telecommunication industry and also reduce both the CAPEX and OPEX.

8. Algorithms for Delay-Tolerant Resource Scheduling for VNO

8.1 Overview

Due to the potential performance gains and the inherent cost-saving benefits by utilizing economies of scale and avoiding duplicated investment on the network infrastructures, RAN sharing among the network operators has recently attracted extensive attention and emerges as a promising mechanism to control both capital expenditure (CAPEX) and operational expenditure (OPEX) [Liang2015]. Specifically, if the UE can freely connect to a base station (BS) of the network operators and the BSs of one network operator can reuse the spectrum of other network operators, the network capacity can be significantly enhanced to accommodate more traffic demands. It is expected that software-defined networking (SDN) will make RAN sharing more flexible [Chen2014, Yu2016, Cusatis2015, Chen2015]. In this chapter, we consider the wireless network as a service under the coordination of C3, where the single ownership of the physical network can be decoupled from the wireless services through enabling network virtualization [Liang2015]. In this way, a mobile network operator (MNO) can build a SDN-based programmable RAN with advanced radio techniques in an area yielding high network throughput. The MNO can then provide the network as a service to multiple virtual network operators (VNOs). For example, an application service provider (e.g., Netflix and Google) can become a VNO to lease wireless resources from the MNO to improve the Quality-of-Service (QoS) for its subscribed UEs and to share the costs for enabling new wireless services.

It has been observed that the UEs' traffic patterns explicitly show temporal and spatial fluctuations, which is responsible for the “tidal phenomena” [Niu2010]. From the perspective of a VNO, it is costly to lease the wireless resources from the MNO to satisfy the peak traffic of its UEs. In literature, there exist a number of works on exploring the delay-tolerance of mobile traffic to improve the usage of network resources [Ha2012, Schulman2010, Wu2016]. The authors in [Ha2012] proposed to move the delay-tolerant traffic to the period when the network is less loaded (load-aware). While in [Schulman2010], the temporal variations in channel conditions experienced by the UEs were used to improve the efficiency of resource utilization (channel-aware). In [Wu2016], the authors addressed the jointly load-aware and channel-aware wireless scheduling to reduce network congestion. However, all these works have been limited to a single UE or a single cell and are not applicable to a large RAN where the traffic loads across cells are weakly coupled.

Different from the works in literature, we focus in this chapter on a large-scale virtualized RAN, from which a VNO leases the network as a service to serve its UEs. The VNO aims at maximizing its revenue by designing an optimal resource scheduling policy to minimize the payments paid to the MNO. The problem is formulated as a finite time horizon constrained Markov decision process (MDP) while taking into account both the delay-tolerance in mobile traffic and the weak load coupling across different BSs. Applying the dual decomposition approach, the problem is decomposed into a series of per-UE MDPs with a much smaller problem size. We further simplify the problem solving for each per-UE MDP by exploring the independence of channel conditions between UEs and BSs.

8.2 System Model

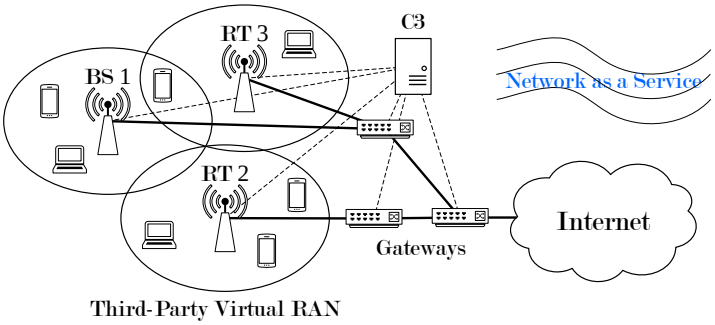


Figure 58 System model of a virtualized radio access network (RAN), in which the VNO has their own virtual network based on a single infrastructure. The VNO level resource allocation is done in C3.

As illustrated in Figure 58, we consider a SDN-based virtualized networking scenario where C3 is managing the control plane of a RAN. The RAN consists of a set $\mathcal{B}$ of BS/RTs, which are owned by a MNO. Note that in COHERENT, RT is a broad concept of BS. We use these two terms interchangeably. We assume that a VNO provides wireless services to a set $\mathcal{N}$ of UEs by leasing wireless resources from the MNO. The network operates over discrete time epochs with each indexed by $t \in \mathbb{N}_+$. Having observed the "tidal phenomena" in mobile data traffic, we focus on the wireless resource scheduling problem over a finite time horizon spanning over one day, namely, $t \in \mathcal{T} = \{1, \cdots, T\}$, where $T \in \mathbb{N}_+$ is the number of time epochs. Note that this watching period can be several hours, the optimal length of which can be learnt from the long-term traffic pattern of the network. The analysis throughout this chapter applies to both the downlink and the uplink scenarios.

For each UE $n \in \mathcal{N}$ of the VNO, we denote the traffic demand and the arrival time by δ_n and a_n, respectively. The $\{\delta_n : n \in \mathcal{N}\}$ and the $\{a_n : n \in \mathcal{N}\}$ are assumed to be identical and independently distributed (i.i.d.) across the UEs. We select A_n^t to record the arrival time of UE n. Then $A_n^t = a_n$ if $t = a_n$ and otherwise $A_n^t = 0$. In a similar way, the waiting time of UE n is tacked by W_n^t, which is defined as $W_n^t = w_n$ if $A_n^{t-w_n} > 0$ and $W_n^t = 0$, otherwise. The UE n is associated with a specific deadline $\bar{w}_n \in \mathbb{N}_+$, which indicates the maximum delay that it can tolerate. In other words, the waiting time w_n should satisfy $0 \le w_n \le \bar{w}_n$. We assume that the demands generated from UEs should be delivered by the end of the watching period, which could be hours or a day, i.e., $a_n + \bar{w}_n - 1 \le T$, $\forall n \in \mathcal{N}$. Each UE experiences time-varying channel conditions, which can be characterized by a homogeneous Markov chain over a finite set $\mathcal{R} = \{r_1, \cdots, r_{|\mathcal{R}|}\}$ of possible data rate states. That is, $R_{b,n}^t \in \mathcal{R}$, where $R_{b,n}^t$ denotes the instant data rate per unit resource that can be used for communications between a RT $b \in \mathcal{B}$ and a UE n during time epoch t and the state transition probability matrix is given by

$$\boldsymbol{\rho}_{b,n} = \left[\rho_{b,n}^{j \to l}\right]_{\substack{1 \le j \le |\mathcal{R}| \\ 1 \le l \le |\mathcal{R}|}}, \tag{1}$$

with each $\rho_{b,n}^{j \to l} \in [0, 1]$ being the transition probability from state r_j to state r_l. The state of the whole network at each time epoch t can be characterized by $\mathbf{X}^t = \left(\mathbf{A}^{(1:t)}, \mathbf{W}^{(1:t)}, \mathbf{R}^{(1:t)}\right)$, where $\mathbf{A}^{(1:t)} = (A_n^\tau : n \in \mathcal{N}, \tau = 1, \cdots, t)$, $\mathbf{W}^{(1:t)} = (W_n^\tau : n \in \mathcal{N}, \tau = 1, \cdots, t)$, and $\mathbf{R}^{(1:t)} = (\mathbf{R}_n^\tau : n \in \mathcal{N}, \tau = 1, \cdots, t)$ with $\mathbf{R}_n^\tau = \left(R_{b,n}^\tau : b \in \mathcal{B}\right)$.

Let $\boldsymbol{\Phi} = (\Phi_n : n \in \mathcal{N})$ represent the policy for scheduling transmissions of UEs. With $\boldsymbol{\Phi}$, the VNO decides which UEs to transmit according to probability $\boldsymbol{\Phi}(\mathbf{X}^t, \mathbf{Y}^t) \in [0, 1]$ after observing the network state $\mathbf{X}^t$ at each time epoch $t \in \mathcal{T}$, where $\mathbf{Y}^t = (\mathbf{Y}_n^t : n \in \mathcal{N})$ is the scheduling decision and $\mathbf{Y}_n^t = \left(Y_{b,n}^t : b \in \mathcal{B}\right)$ with

$$Y_{b,n}^t = Y_{b,n}(\mathbf{X}^t; \boldsymbol{\Phi}) = \begin{cases} 1, \text{if UE } n \text{ is scheduled to transmit via RT } b \in \mathcal{B} \text{ at epoch } t; \\ 0, \text{otherwise.} \end{cases} \tag{2}$$

During the period of each time epoch t, each UE $n \in \mathcal{N}$ can be associated with at most one RT, and thus we have

$$\sum_{b \in \mathcal{B}} Y_{b,n}^t \le 1. \tag{3}$$

We then express the amount of wireless resources $Q_{b,n}(\mathbf{X}^t; \boldsymbol{\Phi})$ consumed by UE n at a RT $b \in \mathcal{B}$ for transmissions as

$$Q_{b,n}(\mathbf{X}^t; \boldsymbol{\Phi}) = \frac{\delta_n Y_{b,n}(\mathbf{X}^t; \boldsymbol{\Phi})}{R_{b,n}^t}. \tag{4}$$

Accordingly, the payment paid by the VNO to the MNO for consuming the wireless resources to satisfy the UEs at each time epoch $t \in \mathcal{T}$ can be defined as

$$F(\mathbf{X}^t;\mathbf{\Phi}) = \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}}\mu_b Q_{b,n}(\mathbf{X}^t;\mathbf{\Phi}), \quad (5)$$

where $\mu_b > 0$ is the constant price per unit resource at a RT $b \in \mathcal{B}$. The objective of the VNO is to configure an optimal policy that maximizes the expected payoff over the finite time horizon. Assume that the price for charging the UEs (per unit traffic demand) is fixed, the problem turns to minimizing the expected payment, which can be formally given by

$$\min_{\mathbf{\Phi}} c(\mathbf{\Phi}) = \sum_{t\in\mathcal{T}} \mathbf{E}\left[l\left(F(\mathbf{X}^t;\mathbf{\Phi})\right)\right] \quad (6a)$$
$$\text{s.t. } v_n(\mathbf{\Phi}) \le \bar{v}_n, \forall n \in \mathcal{N}; \quad (6b)$$
$$\text{constraint (3)},$$

where $l(\cdot)$ is a strictly convex function and $v_n(\mathbf{\Phi})$ is the deadline violation probability with a threshold $\bar{v}_n \in [0,1]$ for each UE n. Constraint (6b) is a QoS-related restriction that incorporates the weak load coupling, owning to the limited wireless resources at a RT. It can be observed that the resource scheduling problem in (6) is a finite-horizon constrained MDP, which in general is difficult to solve due to the huge problem size. In this chapter, we concentrate on the case where there are a large set of VNOs serving a large number of UEs. This makes the problem solving in a centralized way even more challenging. The proposed solution framework in [Wu2016], which though has been extended to multiple cells by simply assuming a Markovian UE-RT association process, fails to solve the Problem (6). The following section will develop a distributed algorithm to approach the optimal solution of (6).

8.3 Proposed Algorithm

To solve the large-scale constrained MDP formulated in the previous section, we first derive a lower bound of (6), which can be obtained by solving a new problem (P-0) below,

$$\min_{\mathbf{\Phi}} \underline{c}(\mathbf{\Phi}) = \sum_{t\in\mathcal{T}} l\left(f^t(\mathbf{\Phi})\right) \quad (7)$$
$$\text{s.t. constraints (3) and (6b)},$$

where $f^t(\mathbf{\Phi}) = \mathbf{E}[F(\mathbf{X}^t;\mathbf{\Phi})] = \sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}}\mu_b q_{b,n}^t(\mathbf{\Phi})$ is the expected payment at each time epoch $t \in \mathcal{T}$ with $q_{b,n}^t(\mathbf{\Phi}) = \mathbf{E}\left[Q_{b,n}(\mathbf{X}^t;\mathbf{\Phi})\right]$ being the expected amount of resource consumption at a RT $b \in \mathcal{B}$ by a UE $n \in \mathcal{N}$. Then we proceed to decouple the dual problem of (7) into multiple subproblems via the Lagrangian dual decomposition. The sub-problems can be solved independently at the VNO and the UEs sides.

A. Dual Decomposition

By introducing a new variable u^t, we rewrite the primal formulation in (7) in the following form (P-1)

$$\min_{\mathbf{\Phi}} c(\mathbf{\Phi}) = \sum_{t\in\mathcal{T}} l(u^t) \quad (8a)$$
$$\text{s.t. } f^t(\mathbf{\Phi}) \le u^t, \forall t \in \mathcal{T}; \quad (8b)$$
$$u^t = \sum_{b\in\mathcal{B}}\mu_b\zeta_b^t, \forall t \in \mathcal{T}; \quad (8c)$$
$$\zeta_b^t \le \bar{\zeta}_b, \forall t \in \mathcal{T}, \forall b \in \mathcal{B}; \quad (8d)$$
$$\text{constraints (3) and (6b)},$$

where $\zeta_b^t > 0$ denotes the expected resource supply of the VNO from a RT b at each time t with a maximum threshold of $\bar{\zeta}_b$. Let $\boldsymbol{\lambda} = [\lambda^t : t \in \mathcal{T}]$ be the Lagrangian multipliers (LMs) to (8b). The Lagrangian then is

$$J(\mathbf{\Phi}, \mathbf{u}, \boldsymbol{\lambda}) = \sum_{t\in\mathcal{T}}(l(u^t) - \lambda^t u^t) + \sum_{t\in\mathcal{T}}\lambda^t\sum_{n\in\mathcal{N}}\sum_{b\in\mathcal{B}}\mu_b q_{b,n}^t(\mathbf{\Phi}), \quad (9)$$

where $\mathbf{u} = [u^t : t \in \mathcal{T}]$. The dual problem is

$$\max_{\boldsymbol{\lambda}} G(\boldsymbol{\lambda}) \ \text{s.t.}\ \boldsymbol{\lambda} \geq \mathbf{0}, \tag{10}$$

where

$$G(\boldsymbol{\lambda}) = \inf_{\boldsymbol{\Phi},\mathbf{u}} J(\boldsymbol{\Phi}, \mathbf{u}, \boldsymbol{\lambda}). \tag{11}$$

In line with the discussions, solving (10) easily leads to the solution of (8). Let $\boldsymbol{\zeta} = [\zeta_b^t : b \in \mathcal{B}, t \in \mathcal{T}]$. For a given $\boldsymbol{\lambda}$, (11}) can be decomposed into the following subproblems:

$$\min_{\boldsymbol{\zeta}} \textstyle\sum_{t\in\mathcal{T}} (l(u^t) - \lambda^t u^t) \ \text{s.t. constraints (8c) and (8d)}, \tag{12}$$

$$\min_{\boldsymbol{\Phi}} \textstyle\sum_{t\in\mathcal{T}} \lambda^t \sum_{n\in\mathcal{N}} \sum_{b\in\mathcal{B}} \mu_b q_{b,n}^t(\boldsymbol{\Phi}) \ \text{s.t. constraints (3) and (6b)}, \tag{13}$$

which can be solved separately at the VNO and the UEs sides without coordination. (12) is easy to solve because of the convexity of the objective function. For each UE $n \in \mathcal{N}$, we apply a backward induction method [Sandholm2012] to (13).

The individual independent scheduling policy Φ_n for each UE $n \in \mathcal{N}$ is a mapping: $\mathcal{X}_n \to [0,1]^T$, where $\mathcal{X}_n$ is the local state space. For a state realization $\mathbf{X}_n^T = \left(A_n^T, W_n^T, \mathbf{R}_n^{(1:T)}\right) \in \mathcal{X}_n$ over the finite time horizon, $\Phi_n(\mathbf{X}_n^T) = [\Phi_n(\mathbf{X}_n^t, \mathbf{Y}_n^t) : t \in \mathcal{T}]$, where each $\Phi_n(\mathbf{X}_n^t, \mathbf{Y}_n^t) \in [0,1]$ is the probability of local scheduling decision $\mathbf{Y}_n^t$ at each time epoch t and can be calculated as

$$\Phi_n(\mathbf{X}_n^t, \mathbf{Y}_n^t) = \begin{cases} \phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right), & \text{if } t = a_n, w_n = 0, \mathbf{Y}_n^t = \mathbf{I}_b; \\ \phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right) \prod_{w\in\{0,\cdots,w_n-1\}} \left(1 - \sum_{z\in\mathcal{B}} \phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_z\right)\right), & \text{if } t = a_n, 0 < w_n \leq \bar{w}_n, \mathbf{Y}_n^t = \mathbf{I}_b; \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

with $\mathbf{I}_b$ indicating that UE n is associated with a RT $b \in \mathcal{B}$ during the time epoch and $\phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right) = \mathbf{Pr}\left\{Y_{b,n}^{a_n+w_n} = 1\right\}$ being the corresponding probability when the traffic arrival time is a_n and the waiting time is w_n. The expected resource consumption of UE n at RT b during the time epoch t can be written as

$$e_{b,n}(\mathbf{X}_n^t; \Phi_n) = \frac{\delta_n \Phi_n\left(\mathbf{X}_n^t, \mathbf{I}_b\right)}{R_{b,n}^t}. \tag{15}$$

Moreover, the UEs should finish their transmissions before the deadlines, that is,

$$\textstyle\sum_{w_n\in\{0,\cdots,\bar{w}_n-1\}} \sum_{b\in\mathcal{B}} \Phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right) = 1. \tag{16}$$

We thus get

$$q_{b,n}^t(\Phi_n) = \textstyle\sum_{\mathbf{X}_n^t} \pi_n(\mathbf{X}_n^t; \Phi_n) e_{b,n}(\mathbf{X}_n^t; \Phi_n), \tag{17}$$

where π_n is the probability distribution of the local state under the scheduling policy Φ_n. Consequently, (13) is equivalent to

$$\min_{\Phi_n} \textstyle\sum_{t\in\mathcal{T}} \lambda^t \sum_{n\in\mathcal{N}} \sum_{b\in\mathcal{B}} \mu_b q_{b,n}^t(\Phi_n) \ \text{s.t. constraints (3), (16) and (17)}. \tag{18}$$

B. Algorithms at UE and VNO side

From the discussions of dual decomposition carried out in the previous section, we propose a distributed algorithm to solve the problem in (7) iteratively. The dual problem in (10) can be solved by the gradient projection method, which yet requires the VNO and the UEs to solve (12) and (13), $\forall n \in \mathcal{N}$, separately.

Note that algorithm run for VNO is executed in C3. The d-th iteration of the distributed algorithm is given in the following.

- **UEs side**

1) Based on the $\boldsymbol{\lambda}^{(d)}$ from the VNO and the local channel dynamics, each UE $n \in \mathcal{N}$ independently solves (18), and hence obtains $\Phi_n^{(d)} = \left\{\Phi_n^{(d)}(\mathbf{X}_n^T): \mathbf{X}_n^T \in \mathcal{X}_n\right\}$ and $\phi_n^{(d)} = \left\{\phi_n^{(d)}(\mathbf{X}_n^T): \mathbf{X}_n^T \in \mathcal{X}_n\right\}$, where $\phi_n^{(d)}(\mathbf{X}_n^T) = \left[\phi_n^{(d)}(\mathbf{X}_n^t, \mathbf{Y}_n^t): t \in \mathcal{T}\right]$.
2) UE n then reports to the VNO the expected resource consumption $q_{b,n}^{t,(d)}\left(\Phi_n^{(d)}\right)$ at each RT $b \in \mathcal{B}$ during each time epoch $t \in \mathcal{T}$ according to (17).

- **VNO side**

1) The VNO collects from all UEs the expected resource consumptions at the RTs across the finite time epochs and calculates the aggregate expected load at each time $t \in \mathcal{T}$ for all RTs, $\sum_{n\in\mathcal{N}} q_{b,n}^{t,(d)}\left(\Phi_n^{(d)}\right), \forall b \in \mathcal{B}$.
2) The VNO proceeds to solve (12) and obtains $\boldsymbol{\zeta}^{(d)} = \left[\zeta_b^{t,(d)}: b \in \mathcal{B}, t \in \mathcal{T}\right]$.
3) The new LMs are updated according to

$$\lambda^{t,(d+1)} = \max\left\{0, \lambda^{t,(d)} + \alpha^{(d)} \sum_{b\in\mathcal{B}} \mu_b \left(\sum_{n\in\mathcal{N}} q_{b,n}^{t,(d)}\left(\Phi_n^{(d)}\right) - \zeta_b^{t,(d)}\right)\right\}, \quad (19)$$

$\forall t \in \mathcal{T}$, where $\alpha^{(d)}$ is the step size.

The LMs can be seen as the control signals that bridge the resource demands from UEs and the resource supplies leased by the VNO from the MNO. When we use a constant step size for LMs updating in (19), i.e., $\alpha^{(d)} = \alpha$, the following theorem shows that our proposed distributed resource scheduling algorithm provides a near-optimal solution to the problem in (7). Let $\underline{c}^{(\mathrm{P}-0),*}(\alpha)$ denote the optimal value of (7). The algorithm is illustrated in Figure 59.

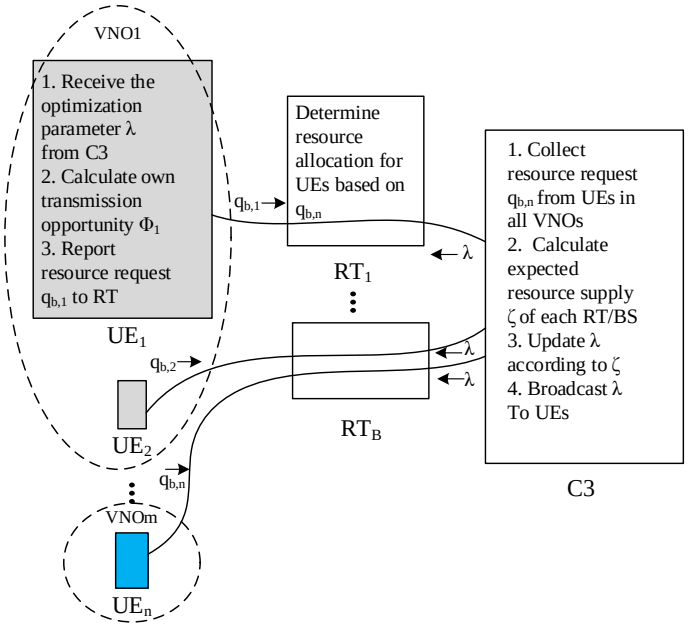


Figure 59 Illustration of proposed algorithm

Theorem 1: For the proposed resource scheduling algorithm with constant step size $\alpha^{(d)} = \alpha$, the optimal value $\underline{c}^{(\mathrm{P}-1),*}(\alpha)$ of (8) is upper bounded by

$$\underline{c}^{(\mathrm{P}-1),*}(\alpha) \leq \underline{c}^{(\mathrm{P}-0),*} + \alpha M, \tag{20}$$

where $M = 2Tu^*$. Herein, $u^* = \max_{t\in\mathcal{T}} u^{t,*}$, where $u^{t,*}$ is the maximum value of (8c) subject to constraint (8d).

Given the control signals $\boldsymbol{\lambda}^{(d)}$ at each iteration d, each UE $n \in \mathcal{N}$ faces a single-agent constrained finite-horizon MDP problem. In the following discussions, we drop the iteration index d for notational simplicity. More specifically, UE n behaves according to the statistics of local states to minimize the expected weighted resource consumption subject to a constraint of bounded deadline violation probability, as indicated by (18). In this chapter, we consider a deterministic deadline constraint, which means that all arriving traffic demands must be transmitted before the deadlines, namely, $\bar{v}_n = 0$. In other words, there exist a $w_n \in \{0, \cdots, \bar{w}_n - 1\}$ and a $b \in \mathcal{B}$ such that $\phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right) = 1$, where the traffic arrives at time epoch a_n. Solving (18) under the deterministic deadline constraint and using the relation between Φ_n and ϕ_n, we obtain

$$\phi_n\left(\mathbf{X}_n^{a_n+w_n}, \mathbf{I}_b\right) = \begin{cases} 1, & \text{if } \frac{\lambda^{a_n+w_n}\mu_b\delta_n}{R_{b,n}^{a_n+w_n}} \leq \min\left\{\mathbf{E}\left[\mathbb{V}_n\left(\mathbf{x}_n^{a_n+w_n+1}\right)\middle|\mathbf{x}_n^{a_n+w_n}\right], \min\limits_{z\in\mathcal{B}\setminus\{b\}}\left\{\frac{\lambda^{a_n+w_n}\mu_z\delta_n}{R_{z,n}^{a_n+w_n}}\right\}\right\}; \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

where the $\mathbb{V}$-function $\mathbb{V}_n\left(\mathbf{x}_n^{a_n+w_n+1}\right)$ defines the future weighted resource consumption conditioned on $\mathbf{x}_n^{a_n+w_n} = \left(A_n^{a_n+w_n}, W_n^{a_n+w_n}, \mathbf{R}_n^{a_n+w_n}\right)$ at previous time epoch and can be calculated via the backward induction,

$$\mathbb{V}_n\left(\mathbf{x}_n^{a_n+w_n}\right) = \begin{cases} \mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n}, \mathbf{I}_b\right), & \text{for } w_n = \bar{w}_n - 1; \\ \min\left\{\min\limits_{b\in\mathcal{B}}\left\{\frac{\lambda^{a_n+w_n}\mu_b\delta_n}{R_{b,n}^{a_n+w_n}}\right\}, \mathbf{E}\left[\min\limits_{\mathbf{Y}_n^{a_n+w_n+1}} \mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n+1}, \mathbf{Y}_n^{a_n+w_n+1}\right)\middle|\mathbf{x}_n^{a_n+w_n}\right]\right\}, & \text{for } 0 \leq w_n \leq \bar{w}_n - 2, \end{cases} \tag{22}$$

and

$$\mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n}, \mathbf{Y}_n^{a_n+w_n}\right) = \begin{cases} \sum_{b\in\mathcal{B}} \frac{\lambda^{a_n+w_n}\mu_b\delta_n Y_{b,n}^{a_n+w_n}}{R_{b,n}^{a_n+w_n}}, & \text{for } w_n = \bar{w}_n - 1; \\ \sum_{b\in\mathcal{B}} \frac{\lambda^{a_n+w_n}\mu_b\delta_n Y_{b,n}^{a_n+w_n}}{R_{b,n}^{a_n+w_n}} + \mathbf{E}\left[\mathbb{V}_n\left(\mathbf{x}_n^{a_n+w_n+1}\right)\middle|\mathbf{x}_n^{a_n+w_n}, \mathbf{Y}_n^{a_n+w_n}\right], & \text{for } 0 \leq w_n \leq \bar{w}_n - 2, \end{cases} \tag{23}$$

and the $\mathbb{Q}$-function $\mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n+1}, \mathbf{Y}_n^{a_n+w_n+1}\right)$ as in (23) defines the future weighted resource consumption given the scheduling decision $\mathbf{Y}_n^{a_n+w_n+1}$ under $\mathbf{x}_n^{a_n+w_n+1}$ conditioned on $\mathbf{x}_n^{a_n+w_n}$ and $\mathbf{Y}_n^{a_n+w_n}$.

However, the challenge remains in the calculation of the $\mathbb{Q}$-functions. Even with a reasonable number $|\mathcal{B}|$ of RTs, we can see that the $\mathbb{Q}$-function of each UE $n \in \mathcal{N}$ is with $S = T \cdot \bar{w}_n \cdot (|\mathcal{B}| + 1) \cdot |\mathcal{R}|^{|\mathcal{B}|}$ values. Exploring the independence of channel conditions between the UE and the RTs, the $\mathbb{Q}$-function $\mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n}, \mathbf{Y}_n^{a_n+w_n}\right)$ can be decomposed to be

$$\mathbb{Q}_n\left(\mathbf{x}_n^{a_n+w_n}, \mathbf{Y}_n^{a_n+w_n}\right) = \sum_{b\in\mathcal{B}} \mathbb{Q}_{b,n}\left(\left(A_n^{a_n+w_n}, W_n^{a_n+w_n}, R_{b,n}^{a_n+w_n}\right), Y_{b,n}^{a_n+w_n}\right). \quad (24)$$

Herein, $\mathbb{Q}_{b,n}\left(\left(A_n^{a_n+w_n}, W_n^{a_n+w_n}, R_{b,n}^{a_n+w_n}\right), Y_{b,n}^{a_n+w_n}\right)$ is the per-RT $\mathbb{Q}$-function that is computed as

$$\mathbb{Q}_{b,n}\left(\left(A_n^{a_n+w_n}, W_n^{a_n+w_n}, R_{b,n}^{a_n+w_n}\right), Y_{b,n}^{a_n+w_n}\right) = \frac{\lambda^{a_n+w_n}\mu_b\delta_n Y_{b,n}^{a_n+w_n}}{R_{b,n}^{a_n+w_n}} + \mathbf{E}\left[\mathbb{Q}_{b,n}\left(\left(A_n^{a_n+w_n+1}, W_n^{a_n+w_n+1}, R_{b,n}^{a_n+w_n+1}\right), \mathbb{I}_{\left\{\frac{\mu_b}{R_{b,n}^{a_n+w_n+1}} \le \Upsilon_{|\mathcal{B}|-1}^*\right\}}\right) \middle| \left(A_n^{a_n+w_n}, W_n^{a_n+w_n}, R_{b,n}^{a_n+w_n}\right), Y_{b,n}^{a_n+w_n}\right], \quad (25)$$

where $\Upsilon_{|\mathcal{B}|-1}^*$ is the first order statistics of $|\mathcal{B}|-1$ i.i.d. random variables and $\mathbb{I}_{\{\Xi\}}$ is an indicator function that equals 1 if the condition Ξ is satisfied and 0 otherwise. Now UE n only calculates the $\mathbb{Q}$-functions with $S' = 2\cdot T\cdot \bar{w}_n \cdot |\mathcal{B}|\cdot|\mathcal{R}|$ values.

After obtaining $\phi_n(\mathbf{X}_n^T) = [\phi_n(\mathbf{X}_n^t, \mathbf{I}_b): b\in\mathcal{B}, t\in\mathcal{T}]$ for each UE $n\in\mathcal{N}$, we are able to calculate the expected resource consumption at each time epoch $t\in\mathcal{T}$ as

$$q_{b,n}^t(\Phi_n) = \sum_{a_n\in\mathcal{T}} \mathbf{Pr}\{a_n\}\,\mathbf{E}[\delta_n]\sum_{b\in\mathcal{B}}\sum_{R_{b,n}^t\in\mathcal{R}} \frac{\pi_n^{a_n}(\mathbf{X}_n^t;\Phi_n) Y_{b,n}(\mathbf{X}_n^t;\Phi_n)}{R_{b,n}^t}, \quad (26)$$

where $\sum_{b\in\mathcal{B}} Y_{b,n}(\mathbf{X}_n^t;\Phi_n) = 1$and $\pi_n^{a_n}(\mathbf{X}_n^t;\Phi_n)$ is the probability of transmitting during time epoch t when the arrival time is a_n and the local state at time epoch t is $\mathbf{X}_n^t$ given by

$$\pi_n^{a_n}(\mathbf{X}_n^t;\Phi_n) = \begin{cases} \prod_{b\in\mathcal{B}} \mathbf{Pr}\{R_{b,n}^t\}, & \text{if } t = a_n; \\ \sum_{b\in\mathcal{B}}\sum_{R_{b,n}^{t-1}=r_j\in\mathcal{R},\, R_{b,n}^t=r_l\in\mathcal{R}} (1-\sum_{b\in\mathcal{B}}\Phi_n(\mathbf{X}_n^t,\mathbf{I}_b))\rho_{b,n}^{j\to l}\phi_n(\mathbf{X}_n^t,\mathbf{I}_b), & \text{otherwise.} \end{cases} \quad (27)$$

To ease the computational burden of calculating expected resource consumption at each time epoch t, we propose to online update the LMs as in [Wu2016], which approximates $q_{b,n}^t(\Phi_n)$ by the accumulated resource consumption $\tilde{q}_{b,n}^t(\Phi_n)$ for a local state realization $\mathbf{X}_n^T$,

$$\tilde{q}_{b,n}^t(\Phi_n) = \delta_n \sum_{b\in\mathcal{B}}\sum_{R_{b,n}^t\in\mathcal{R}} \frac{\pi_n^{a_n}(\mathbf{X}_n^t;\Phi_n) Y_{b,n}(\mathbf{X}_n^t;\Phi_n)}{R_{b,n}^t}. \quad (28)$$

8.4 Numerical Results

In this section, we evaluate the performance of our proposed resource scheduling scheme. The performance achieved from our proposed scheme is compared with the following three baselines:

1) Baseline 1 (immediate) -- the traffic demands from UEs are transmitted once they arrive;
2) Baseline 2 (load-aware) -- the control signals, i.e., the LMs, are identical across the T time epochs for determining the scheduling policy [Ha2012];
3) Baseline 3 (channel-aware) -- the UEs transmit once they have better channel conditions than average [Schulman2010].

The convex function $l(q)$ is chosen as $l(q) = (q/K)^t$, where $K > 0$ is a constant value. We assume that the finite time horizon spans over one day with $T = 96$ time epochs (i.e., 15 minutes per time epoch) and the RAN owned by the MNO consists of $|\mathcal{B}| = 2$ RTs. During each time epoch, the traffic

demands of $|\mathcal{N}| = 120$ UEs arrive in the network according to the profile from [Trace2016]. The data rates between the UEs and the RTs across time epochs follow a Markov chain with 31 states and randomly generated state transition probabilities. In all simulations, the arriving traffic demands are divided into the realtime and the delay-tolerant groups, for which the ratio of the numbers of UEs belonging to the two groups is designated as $\theta > 0$.

We first examine the convergence property of the proposed scheme. Figure 60 depicts the duality gaps between problems P-0 and P-1, which validates that the iterative process of the proposed resource scheduling scheme converges at a reasonable speed. Note that in Figure 60 the convergence rate of duality gap was counted by days because that reflects how long the simulation was taken. It took long time to converge mainly because the low performance computer was used in simulation. We draw the normalized resource consumptions of the two RTs under different resource scheduling schemes in Figure 61(a) and (b). From both plots, we can find that the proposed scheme achieves more flat resource consumptions compared with the other three baseline schemes. By deploying our proposed resource scheduling scheme, the delay-tolerant traffic demands from UEs are moved to off-peak period of a day resulting in less resource consumption or are associated to the RT with better channel conditions and lower price leading to less payments, under the guidance of the control signals by the VNO. Finally, the normalized payments paid by the VNO to the MNO under different settings of θ are exhibited in Figure 62. In the plot, $\theta = 0$ corresponds to the scenario that all arriving traffic demands are real-time. It can be found that our proposed resource scheduling scheme achieves minimum payments when there come the delay-tolerant traffic demands from the UEs, and more delay-tolerant traffic brings more opportunities for the VNO to reduce the payments.

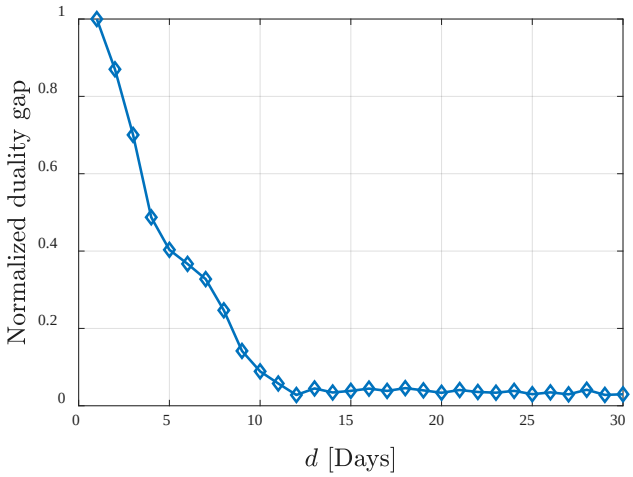
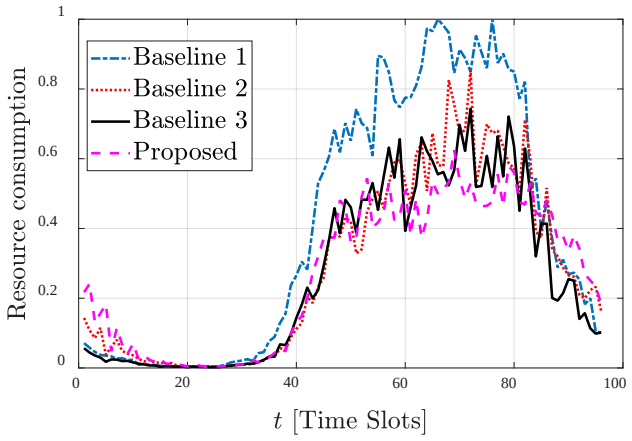
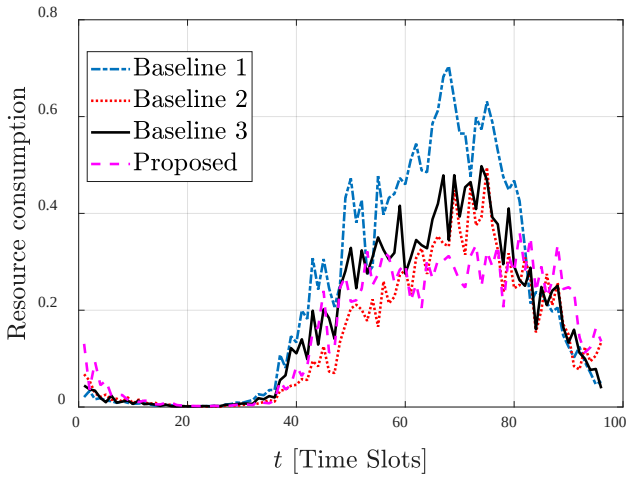


Figure 60 Convergence of the proposed scheme, $\theta = 6$.



(a)



(b)

Figure 61 Normalized resource consumptions at the two RTs across the finite time horizon, $\theta = 12$.

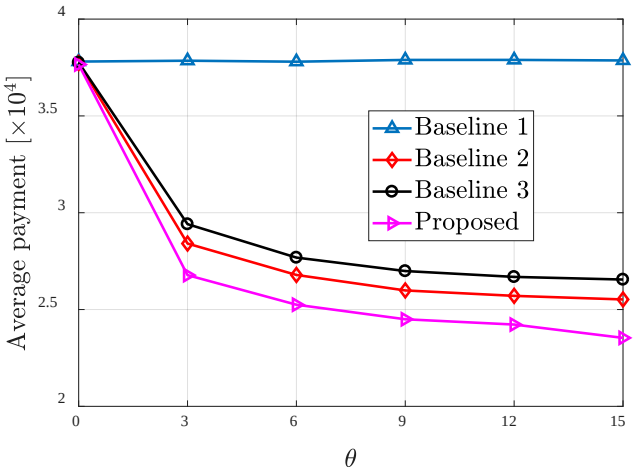


Figure 62 Normalized payments of the VNO to the MNO under different values of θ.

8.5 Conclusions

8.5.1 Technical/Scientific Impact

This chapter considers the scheduling of virtualized RANs built on top of MNO's physical infrastructure, over which a VNO leases wireless resources to serve its UEs. The delay-tolerance in the temporally and spatially fluctuating mobile traffic is fully explored when the VNO designs an optimal resource scheduling policy to minimize the payments to the MNO. The payment minimization problem is formulated as a finite horizon constrained MDP, the solving of which is extremely complicated. To alleviate the computational costs, the problem is decomposed into a series of per-UE MDPs with much smaller problem size via applying the dual decomposition approach. The solving of per-UE MDP can be performed locally at each UE and further simplified by the independence of channel conditions between the UE and the BSs. From the simulation results, we can see that our proposed resource scheduling scheme provides better performance compared with other state-of-art schemes in terms of average network loads at the BSs and the average payments paid by the VNO to the MNO.

8.5.2 Feedback toward development

The work in this chapter focuses on the theoretic study to derive the optimal scheduling scheme for delay-tolerant traffic in the RAN sharing scenario. The assumption of the study is that traffic statistics and scheduling process can be modeled by the Markov process. In the COHERENT control framework, the low layer state information are aggregated at RTC and C3. It is possible to estimate traffic statistics

at RTC and C3 and map them into the proper stochastic model, e.g. the Markov model. Therefore, with the proper methods to derive the stochastic traffic model at RTC and C3, the proposed algorithm in this chapter can be implemented at RTC and C3 to estimate optimal resource allocation strategies for VNO resource sharing. The derived resource allocation strategies can be fed from RTC and C3 to the MAC layer scheduler to optimize the resource usage among VNOs.

8.5.3 Expected business impact

The proposed algorithm in this chapter is an example of how the hierarchy control framework developed in COHERENT can help MNO to optimize the radio resource usage of RAN. In this two-layer control framework, the higher control layer coordinates the resource allocation among RTs based on the centralized knowledge of the network, and provides the guide to the lower control layer to schedule the physical resource. It allows MNOs to use their precious spectrum resource more efficiently.

9. Algorithms for Traffic Steering and Resource Allocation

9.1 Overview

To meet various 5G requirements in terms of ultra-high data rate, low latency and massive connectivity, future 5G mobile networks are envisioned to have multi-dimensional heterogeneity regarding the use of multiple radio access technologies, divergent frequency spectrum bands and diverse cells with different sizes. In these 5G heterogeneous networks, a critical task for the operators is how to efficiently use those diverse network resources to provide satisfactory user experience. In this regard, traffic steering is a promising technique which aims to distribute the user traffic load optimally across the heterogeneous networks by performing resource allocation on a vast variety of network connections including backhaul/fronthaul connections, wireless access connections and device-to-device (D2D) connections. In this section, we consider three traffic steering techniques in three different scenarios. Firstly we consider traffic steering in a multi-RAT 5G network with both mmWave and sub-6 GHz frequencies. The cell association and radio resource problems over the two different bands are considered and a hierarchical solution is proposed. In the second scenario, we consider a network where D2D relaying is applied to enhance the cell edge performance in downlink; the one-hop downlink traffic is offloaded to the two-hop connections enabled by D2D relaying and D2D relay selection and interference management problem is addressed. In the third scenario, Cell Range Extension (CRE) for small cells is considered to offload macro-cell traffic to the small-cells. Part of the work presented here has been reported in [Deng2017mag], [Deng2017ICC], [Lembo2017] and [Lembo2017b].

These traffic steering and resource allocation methods are conceptually different, and may be partially applied in the same network. The first addresses a situation where a RAN has multiple RATs, operating on widely different carrier frequencies, with widely different propagation characteristics and coverage. It is crucial that in this scenario operation on different carrier frequencies is tightly integrated in the RAN, with a RTC directly controlling multiple carriers. The third method, in contrast, applies in any situation with co-channel large and small cells. Here, the objective is to manage interference between these entities, which have a priori different transmit power and coverage area, and where one may steer traffic from a highly loaded macro cell to the small cells by CRE. The capability to serve users in a small cell with CRE is improved by interference management. The first and third scenario thus both apply in a heterogeneous network with cells with variable footprint, but the network topologies are different, and incommensurate. In the first scenario, there are large coverage and small coverage cells operated from the same physical BS sites with different RATs and different carrier frequencies, while in the third scenario, there are large coverage and small coverage cells operated from different sites with the same RAT and the same carrier frequency. The related two methods are thus not likely to be deployed in a network in the same geographical area. The second method applies in a scenario with D2D relaying. This technology component can be added to any RAT and network deployment and could thus be used together with both the first and the third method.

9.1.1 Problem

Future 5G mobile networks are envisioned to exploit both conventional sub-6 GHz and millimeter-wave (mmWave) bands to provide ever-increasing throughput for users. In such a heterogeneous network with both sub-6 GHz and mmWave connections, multi-connectivity and traffic steering over multiple mmWave cells and these different communication bands are essential functions to provide continuous and reliable services. First, mmWave communication is susceptible to the blockage effects: mobile users are expected to experience high fluctuation in throughput performance if they are connected to one single mmWave BS. Second, compared to traditional low frequency bands, the coverage area of mmWave BS is limited and discontinuous. mmWave band alone cannot provide continuous services except that the mmWave BSs are deployed in an ultra-dense way. Using multi-RAT multi-connectivity technologies with both sub-6GHz and mmWave bands will ease the deployment requirements for mmWave and reduce the mmWave deployment cost. In this section, we study the multi-connectivity and traffic steering problems in a 5G multi-RAT network with both mmWave BSs and traditional sub-6 GHz BSs (e.g. a LTE BS). Based on the abstracted parameters proposed in D3.2, a multi-connectivity network graph is used to characterize the multi-RAT connectivity relationship between multi-RAT UEs and BSs. Using the multi-connectivity network graph, a centralized algorithm for cell association is performed by C3 and a multi-RAT resource

allocation algorithm is performed by RTC, to increase a KPI called User Experience Consistency (UEC), which is defined as the ratio of cell-edge user throughput and the network mean throughput.

9.1.2 Algorithm

C3 is the central coordinator which collects global network state information reported by local BSs and controls multiple BSs by sending control commands as shown in Figure 63. The information collected from the BSs includes: (a) the number of connected UEs; (b) the traffic load inside the cell; and (c) link capacity predictions for all connections. The control functionalities in C3 include: (a) definition of utility function for resource allocation; (b) multi-connectivity algorithm; and (c) load balancing algorithm. RTC is located at each BS, and is responsible for (a) performing sub-6GHz & mmWave resource allocation for the connected users based on the scheduling metric set by the C3; and (b) collecting information and channel measurements from UEs and BSs, and do the link reliability and mean throughput estimation, and report them to C3.

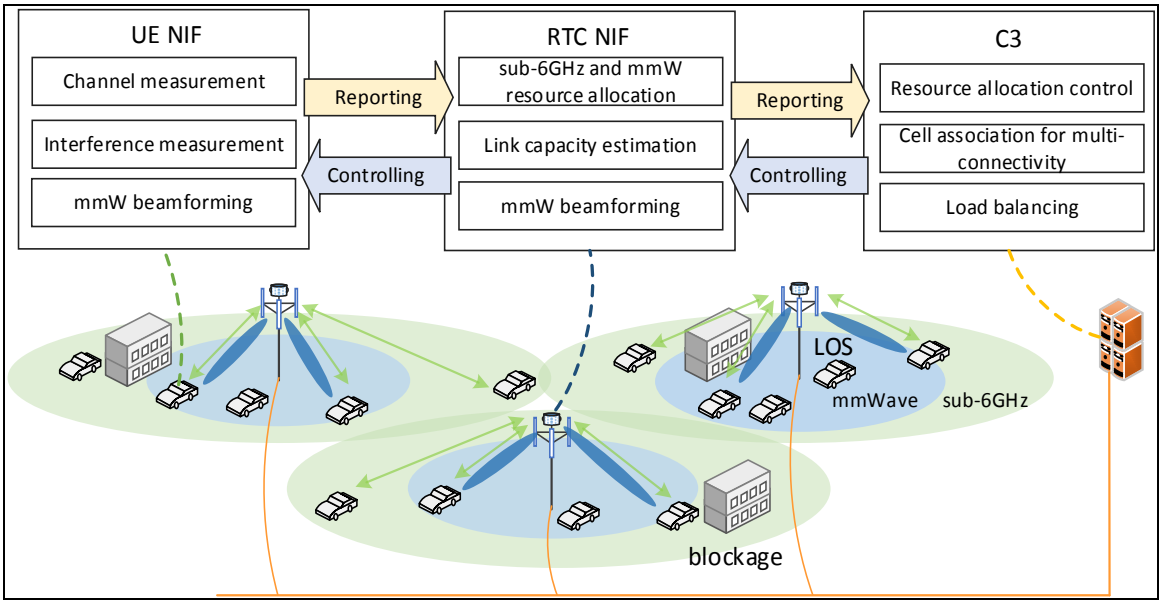


Figure 63 Control framework for sub6-GHz & mmWave multi-connectivity

Table 17 Parameters for RTC/C3 Algorithms

Parameter	Description	Deployment
$\sigma_{i,j}$	Connectivity between node i and j , $\sigma_{i,j} = 1$ if $SNR_{i,j} \geq SNR_{th}$, and $\sigma_{i,j} = 0$ if $SNR_{i,j} < SNR_{th}$, where SNR_{th} is the minimum SNR required for a data transmission.	RTC/C3
$m_{i,j}^{(T)}$	Mean throughput between node i and j, during an evaluation time period T, using one frequency resource block.	RTC/C3
$c_{i,j}$	$c_{i,j}$ = 1 if UE j is associated with BS i, and BS i will allocate resources to UE j, otherwise $c_{i,j} = 0$, and no resource will be allocated to link (i, j).	C3
$N_{i,j}$	Number of time/frequency resources allocated to link between BS i and UE j.	RTC

Assuming that the sub 6-GHz has B_1 resource blocks and mmWave has B_2 resource blocks, there are totally N UEs, and M_1 sub6-GHz BSs and M_2 mmWave BSs in the network. We consider an integrated network, where a mmWave BS is added to each sub-6-GHz BS, so that we will have $M_1 = M_2$. After channel measurements and reporting, C3 will have a connectivity network graph represented by a connectivity matrix $[\sigma_{i,j}]_{N\times(M_1+M_2)}$, and a mean throughput matrix $[m_{i,j}]_{N\times(M_1+M_2)}$. Assuming all

UEs are fully-loaded with full-buffer traffic, and the scheduling metric is proportional fairness for all UEs, then the object function for network control is:

$$U = \sum_{i=1}^{N} \log\left(\sum_{p=1}^{M_1} c_{i,p} N_{i,p} m_{i,p} + \sum_{q=1}^{M_2} c_{i,q+M_1} N_{i,q+M_1} m_{i,q+M_1}\right). \quad (9\text{-}1)$$

Here, $m_{i,p}$ and $m_{i,q+M_1}$ are the estimated link throughputs for sub6-GHz and mmWave respectively. Cell association should satisfy the constraints:

$$\sum_{p=1}^{M_1} c_{i,p} \leq 1, \text{ for } i = 1,2,\dots,N; \quad (9\text{-}2)$$

$$\sum_{q=1}^{M_2} c_{i,q+M_1} \leq 1, \text{ for } i = 1,2,\dots,N; \quad (9\text{-}3)$$

$$c_{i,j} \leq \sigma_{i,j}, \text{ for } i = 1,2,\dots,N \text{ and } j = 1,2,\dots,M_1 + M_2. \quad (9\text{-}4)$$

Resource allocation should satisfy:

$$\sum_{i=1}^{N} N_{i,p} \leq B_1, \text{ for } p = 1,2,\dots,M_1; \quad (9\text{-}5)$$

$$\sum_{i=1}^{N} N_{i,q+M_1} \leq B_2, \text{ for } q = 1,2,\dots,M_2. \quad (9\text{-}6)$$

To maximize the global objective function U, we propose a hierarchical scheme in which C3 tries to find an optimal cell association solution $[c_{i,j}]_{N\times(M_1+M_2)}$ by assuming $N_{i,p} = B_1/(\sum_{k=1}^{N} c_{k,p})$, and $N_{i,q+M_1} = B_2/(\sum_{k=1}^{N} c_{k,q+M_1})$. Then, the global objective function for cell association algorithm in C3 is

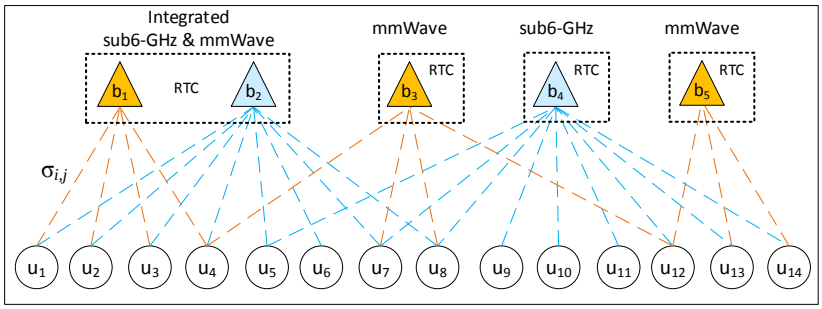
$$U_{C3} = \sum_{i=1}^{N} \log\left(\frac{B_1}{\sum_{k=1}^{N} c_{k,p}} \sum_{p=1}^{M_1} c_{i,p} m_{i,p} + \frac{B_2}{\sum_{k=1}^{N} c_{k,q+M_1}} \sum_{q=1+M_1}^{M_2+M_1} c_{i,q+M_1} m_{i,q+M_1}\right). \quad (9\text{-}7)$$

To maximize U_{C3} under constraints (6-2), (6-3) and (6-4), C3 needs to search over all possible combinations of $c_{i,p}$ and $c_{i,q+M_1}$. Compared to maximizing U, the searching space for maximizing U_{C3} is much smaller. The size of the searching space is smaller than $\prod_{i=1}^{N}(2^{k(u_i)} - 1)$, where $k(u_i)$ is the degree of UE node u_i in the connectivity network graph. The space is still very large, for example in Figure 64 (a), the size of searching space is about 1.75e6. We assume that mmWave bandwidth is much larger than sub6-GHz, to reduce the computation overhead, we propose a heuristic algorithm as shown in Algorithm 18.

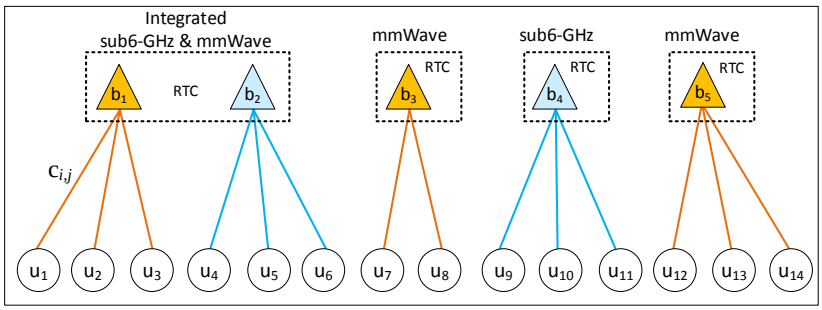
Algorithm 18 C3 Algorithm for cell association

Input: $[\sigma_{i,j}]_{N\times(M_1+M_2)}$, $[m_{i,j}]_{N\times(M_1+M_2)}$.
(1) Set $[c_{i,j}]_{N\times(M_1+M_2)}$ as a matrix of zeros.
(2) FOR $i = 1,2,\dots,N$
(i) If UE u_i has a connection to an mmWave BS $q + M_1$, and the estimated throughput of this connection is larger than any other connections, it would connect to this mmWave BS only, and C3 set $c_{i,q+M_1}$=1.
(ii) If UE u_i has only sub-6GHz connections, it will connect to a sub-6GHz BS p, such that $\frac{B_1}{0.0001+\sum_{k=1}^{N} c_{k,p}} m_{i,p}$ is maximized over all possible p, and C3 set $c_{i,p}$=1.
(iii) If UE u_i has both sub-6GHz & mmWave connections and the maximum-throughput connection is $\sigma_{i,p}$ for a sub-6GHz BS p. Then fix $[c_{k,j}]_{(N-1)\times(M_1+M_2)}$ with $k \neq i$ and search over all possible combinations of $[c_{i,j}]_{1\times(M_1+M_2)}$ with $k = i$, find the combination $[c_{k,j}]_{1\times(M_1+M_2)}$ that maximize U_{C3}.
End of **FOR**
Output: $[c_{i,j}]_{N\times(M_1+M_2)}$.

In the above C3 Algorithm, step **(i)** ensures that a UE chooses mmWave connection preferentially, step **(ii)** performs load balancing for sub-6GHz BSs. An example of the output cell association network graph is depicted in Figure 64 (b).



(a) connectivity graph



(b) cell association graph

Figure 64 Examples of (a) a multi-connectivity network graph and (b) the cell association network graph which is produced by C3 based on (a).

C3 will send the cell association matrix $[c_{i,j}]_{N\times(M_1+M_2)}$ to local RTCs, each will then try to maximize the following local objective function for all UEs associated to the sub6-GHz BS, or mmWave BS or integrated sub6-GHz & mmWave BS, under constraints (6-5) and (6-6),

$$U_{RTC} = \sum_{i:\, c_{i,j}=1} \log\left(\sum_{j:\, c_{i,j}=1} c_{i,j} N_{i,j} m_{i,j}\right). \quad (9\text{-}8)$$

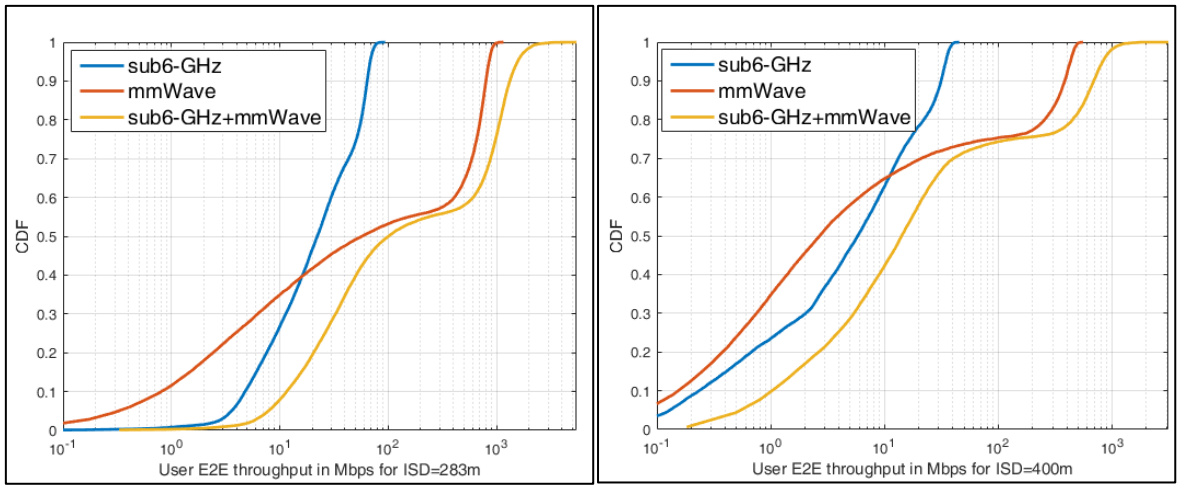
To maximize local U_{RTC}, $N_{i,j}$ is assumed to be a continuous positive real number, and Gradient Descent Algorithm (GDA) is used to find the optimal solution for local resource allocation.

9.1.3 Evaluation

In this section, we study the performance gains achieved via multi-connectivity and the proposed control framework in Manhattan scenario for an integrated sub-6GHz & mmWave network. The UEs are vehicles equipped with mmWave antenna array and a sub-6GHz omnidirectional antenna. Vehicles are uniformly distributed along the streets, BSs are placed at corners of the streets with inter site distances (ISD) as 283m or 400m. The density of active UEs (destinations) is one per 100m (on each street). Table 19 details the simulation parameters.

Table 19 System-level simulation parameters

Scenario	Manhattan street, 1200m × 1200m	
Boundary conditions	Wrap-around in XY-direction	
ISD	283 m, 400 m	
sub-6GHz/mmW frequency	5.9G/28G	
sub-6GHz/mmW bandwidths	40MHz/500MHz	
mmWave LOS PL model	$61.4 + 20 \times \log_{10}(d)$	
mmWave NLOS PL model	$72.0 + 30 \times \log_{10}(d)$	
sub-6GHz PL model	WINNER II B1	
Number of mmW antennas	BS: 8×8 UPA	UE: 8 UCA
Number of sub-6GHz antennas	BS: 1, omnidirectional	UE: 1, omnidirectional
Maximum mmW TX power	BS: 24 dBm	UE: 21 dBm
mmWave Beamforming	Analog Beamforming	
Resource Scheduling	Proportional Fairness (PF)	



(a) (b)

Figure 65 Simulation results for multi-connectivity in an integrated network

To illustrate the benefit of multi-connectivity in future mmWave networks, we consider the multi-connectivity in the integrated sub-6GHz & mmWave network. A proportional fairness utility is assumed, the network scheduler allocation sub-6GHz and mmWave resources to multiple users by maximizing the fairness utility under the constraints of limited resources**.** We compare the throughput CDF performance for three settings: 1) sub-6GHz network with single-connectivity; 2) mmWave network with single connectivity and 3) integrated sub-6GHz & mmWave network with multi-connectivity. Figure 65 shows the simulation results for three settings. Using stand-alone mmWave connection, a lot of users are in mmWave outage. As compared, sub-6GHz & mmWave dual-connectivity can increase the performance of cell edge users significantly. The performance gain is achieved by allocating more sub-6GHz resources for cell-edge UEs since UEs closed the BS would use mmWave resources for their communications.

9.2 Traffic steering and resource allocation for D2D relaying

9.2.1 Problem

Network-assisted Device-to-device (D2D) relaying is a promising way to provide enhanced cell-edge performance in cellular networks. The underlying idea is to exploit the availability of potential proximal D2D connections to setup beneficial multi-hop transmissions to/from base stations (BS). Compared to a multi-hop cellular network with fixed relays, a multi-hop cellular network with mobile D2D relaying is more difficult to manage. First, the active UEs and potential D2D relays are randomly distributed inside multiple cells. A BS controller needs to select a proper D2D relay for a UE who can benefit from multiple-hop relaying, and to allocate proper resources to the D2D links. The following questions have to be addressed: 1) When to relay for a UE? 2) How to select a relay for a UE? 3) How to allocate resources for a D2D relaying transmission? Furthermore, in a multi-cell network where all cells adopt similar relaying strategies, D2D relaying decisions made in one cell would affect a neighbor cell's decision through changing the interference power distribution. It is the objective of this section to study the interference interaction between cells in a network with D2D relaying, and to provide tools for a network controller to set parameters to provide enhanced cell-edge performance. We consider two-hop relaying for downlink communication. The interference characteristics for a multi-cell network with two-hop D2D relaying depends on how much traffic is offloaded to two-hop relaying from one-hop DL transmissions, and the distribution of the selected D2D relays. Furthermore, the distribution of the selected relays depends on the distributions of BSs and cell-edge UEs (which are identified by a network controller).

9.2.2 Algorithm

To enable D2D relaying in cellular network, D2D discovery, relay selection and resource allocation must be considered. We assume that all UEs can perform channel measurements and report the channel information to the serving BS. A real-time controller (RTC) is responsible for the relay selection and resource allocation for D2D relaying operations in its cell. We assume that a coordinated controller (C3) gathers information from the BSs on their interference measurements, and accordingly designs the relaying distance to be used in the network. The BS controller has the responsibility to keep the probability of relaying at the target value. In this regard, C3 will optimize the R_r to be used by each BS, each BS then performs D2D relay selection for those UEs that can use D2D relaying service. We assume that the BS controller has the knowledge of D2D channel path loss for close UE pairs, and the aggregate interference from other BSs. We also assume the same DL D2D relaying strategy is applied in all the cells, i.e. the same probability of D2D relaying, the same principle for relay selection (which determines the optimal relay for a cell-edge UE) and the same fairness metric for resource allocation (which is the objective function to be maximized for multiple UEs). The relaying decisions in one cell are made according to the current observed network state (e.g. aggregate interference at UE and distribution of relay candidates). Figure 66 illustrates the monitoring/command cycle with a clear assignment of the steps of the D2D relaying network management algorithm to the control entities.

Table 20 Parameters for the D2D relaying Algorithm

Parameter	Description	Deployment
D	Minimum inter-BS distance	C3
R_c	Cell radius	C3
ρ_{bs}	BS distribution density	C3
$I_{dl}(d)$	The aggregate interference from all neighboring BSs for a UE located at a distance d from it serving BS	RTC
$I_{d2d}(d)$	The aggregate interference from all D2D relays in neighboring cells for a UE located at a distance d from it serving BS	RTC
$I(d)$	The total interference for a UE located at a distance d from it serving BS	RTC
R_r	D2D relay distance, UE at a distance from its serving BS larger than which would use D2D relaying service	C3
p_r	The probability that D2D relaying is used in a cell with a specific resource block	RTC

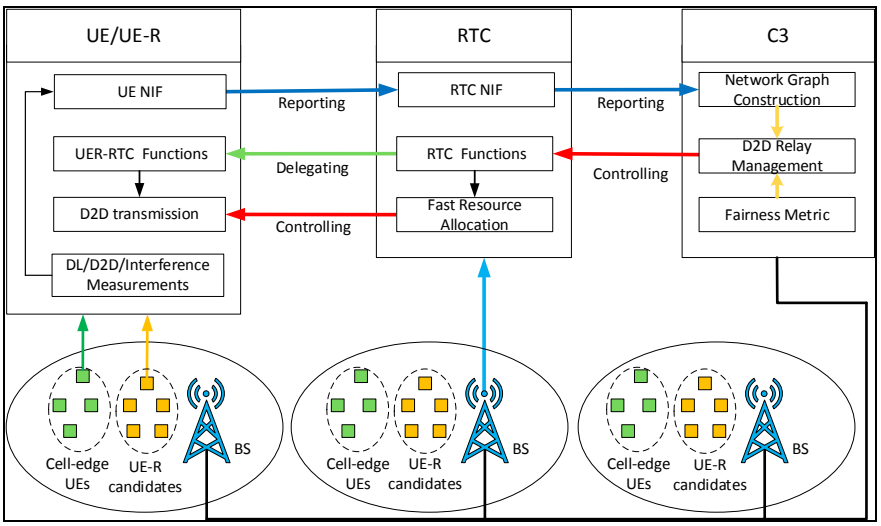
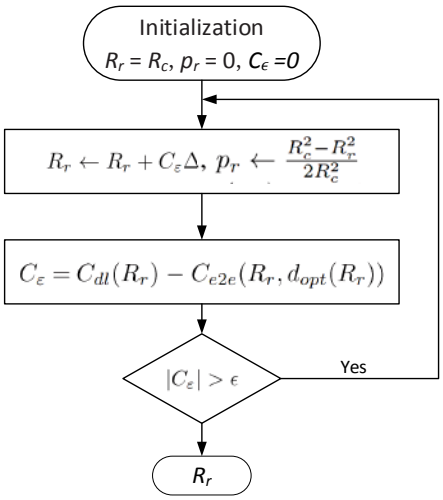


Figure 66 Measurements, reporting and controlling for multi-cell D2D relaying networks

For a multi-cell network with large cells serving many users, and where the cells are all similar, Algorithm 21 describes the algorithm to be performed by C3, where $C_{dl}(x)$ is the estimated direct downlink throughput for a UE located at distance x and $C_{e2e}(x, y)$ is the estimated end-to-end two hop throughput for a UE located at distance x and using a D2D relay located at distance y. The C3 is assumed

to have all the parameters listed in Table 20. The D2D relaying probability is estimated as $p_r = \frac{R_c^2 - R_r^2}{2R_c^2}$. At the initialization of the algorithm, $R_r = R_c$, which means that no D2D relaying is adopted in the network. C3 then decreases R_r and sends this control parameter to all BSs. After that, in each cell, cell-edge UEs at a distance larger than that can start to use D2D relaying service under the control of BS RTC. As some UEs start to offload their traffic from direct DL transmission to 2-hop D2D relaying transmission, the transmit power of BSs will decrease, and hence the interference power from BS to UEs and relays will decrease. At the same time, as p_r increases, D2D interference $p_r I_{d2d}(d)$ will increase. However, as the transmit power of BS is much larger than relay transmit power, the typical aggregate inter-cell interference experienced by a UE will decrease when D2D relaying probability increases. As the interference environment has changed, the RTCs at BSs need to update their relaying strategies and resource allocation scheme according to the new observed network state and report the observed inter-cell interference to C3. C3 then will continuously update R_r, using the algorithm shown in Algorithm 21. The network will then reach an equilibrium state, at which UEs located at distance R_r from serving BS will have the same data rate for both direct DL service and D2D relaying service. At this point, all the UEs that can benefit from D2D relaying will try to use D2D relaying other than direct DL service. The underlying assumption is that the number of user and D2D relays is large, so that the effect of the starting/ending of an individual flow has negligible effect on the network interference state. In a real situation, where an individual flow produces a non-negligible fraction of total interference, a static equilibrium state would not be reached-network interference would fluctuate around the equilibrium point. The stability of an equilibrium w.r.t. an individual flow, and the related admission control are interesting future research topics.



Algorithm 21 Searching for optimal R_r in equilibrium state.

9.2.3 Evaluation

In this section, we consider two multi-cell networks for DL D2D relaying, the triangular lattice (hexagonal cells) and the square lattice (square cells). Both networks have a minimum inter-BS distance D = 500 m and the cell radius used in the analysis is R_c = 250m. The square lattice model has a lower BS density compared to the triangular lattice. Simulation and analysis results are provided are compared in this section. We simulate the DL D2D relaying in the two network scenarios according to Table 22.

Parameter	Symbol	Setting
System bandwidth	W	40 MHz
Minimum distance between BSs	D	500 m
Cell radius	R_c	250 m
BS density of triangular lattice	ρ_{bs}^h	$(\frac{\sqrt{3}}{2}D^2)^{-1}$
BS density of square lattice	ρ_{bs}^s	$(D^2)^{-1}$
Active UEs in each cell	N_u	10
Potential relays in each cell	N_r	100
PL model of cellular link	$\ell_b(x)$	$8.18 \times 10^{-5} x^{-3.67}$
PL model of D2D link	$\ell_d(x)$	$6.34 \times 10^{-5} x^{-3.67}$
TX power of BS	P_b	30 dBm
TX power of UE	P_d	24 dBm
Thermal noise power density	N_0	-174 dBm/Hz
Noise figure	N_f	8 dB

Table 22 Simulation parameters

Relay selections and resource allocation for D2D relaying are based on the actual interference measurements. Simulations are carried out in an iterative way. There are N_u = 10 active UEs in each cell, TDMA is used to support multiple UEs. Each active UE in a cell obtains $1/N_u$ of radio resources to ensure proportional fairness. Each active UE gets one resource with a specific label. This resource is used either for direct transmission to the user, for transmission to the relay, or for D2D transmission by the relay. Without interference coordination, each active UE will get a random resource. For each label, there is one active UE using this resource in a cell and this UE will be randomly distributed inside the cell. The transmissions (including DL and D2D) using the same resource in multiple cells would interfere with each other. Each active UE located at distance larger than R_r from its serving BS would select the best relay from the relay candidates for its DL transmission.

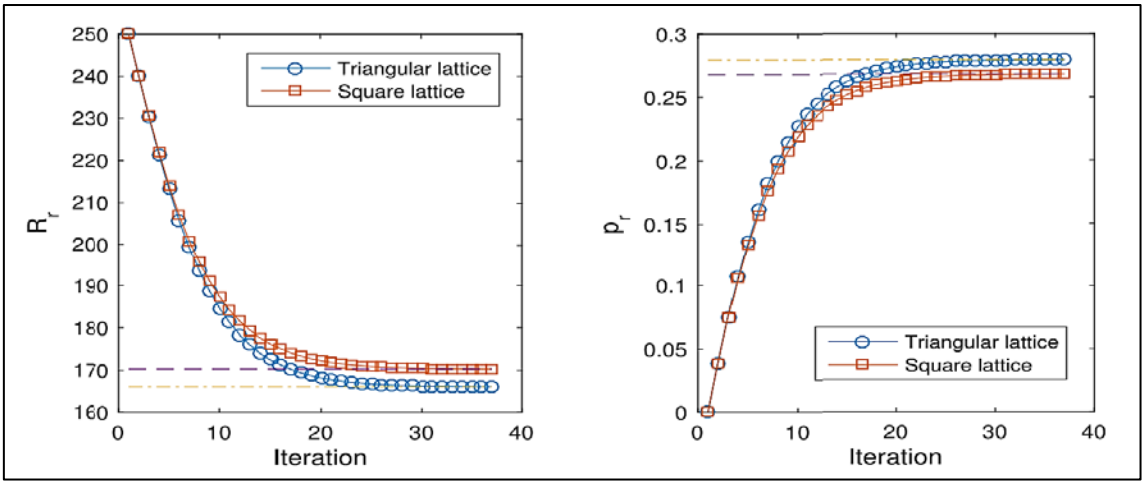


Figure 67 Equilibrium network state

Given the details of network parameters in Table 22, R_r and p_r in equilibrium state can be estimated using Algorithm 21. The search starts at $R_r = R_c$ with no D2D relaying allowed (i.e. p_r = 0). As R_r decreases, UEs at distance larger than R_r start to adopt two-hop D2D relaying. The average interference decreases as p_r increases. Figure 67 shows that Algorithm 21 converges to R_r = 166m, p_r = 0.28 for the hexagonal network, and to R_r = 170m, p_r = 0.27 for the square cells. The estimated radiuses for the relay annulus are r_1 = 94 m, r_2 = 142 m for hexagonal cells and r_1 = 96 m, r_2 = 142 m for square cells.

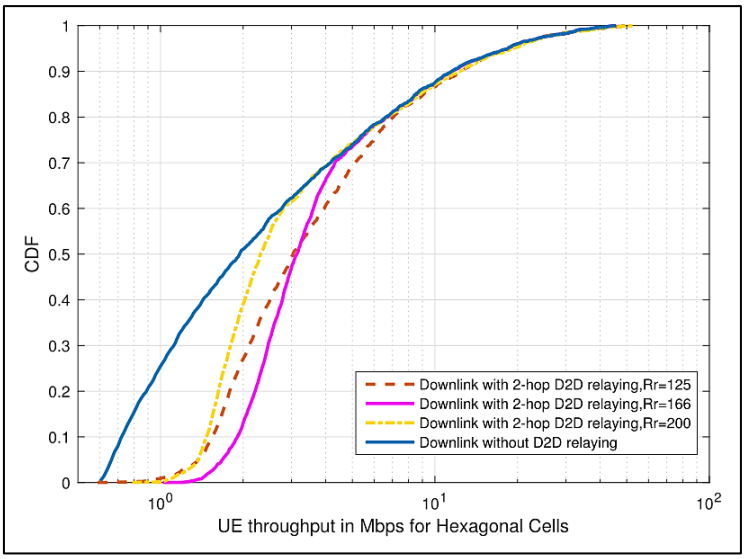


Figure 68 CDF of DL throughput

Figure 68 shows the Cumulative Distribution Functions (CDFs) of UE throughput performance for different D2D relaying strategies in the hexagonal network. The performance in a square network is similar, and thus is omitted. The UEs are uniformly distributed in the 2-D plane. A UE at a distance larger than R_r from its serving BS will adopt two-hop D2D relaying. We use the optimal R_r = 166m according to the analysis and compare it with other values of R_r. D2D relaying helps to boost the performance of cell-edge UEs compared with the strategy that no D2D relaying is used. What's more, the setting of R_r = 166m for the analytical equilibrium network state yields the best cell-edge performance compared to the settings smaller or larger than it, and hence can achieve the best UEC performance. For R_r = 200m, fewer cell-edge UEs adopts two-hop relaying, which increases the aggregate interference as the relaying probability p_r is smaller compared to the strategy with R_r= 166m. For R_r= 125m, more UEs would use D2D relaying and the sum throughput for all active UEs is better than the strategy with R_r= 166m. However, a portion of cell-edge UEs will be affected by the increasing inter-cell D2D interference, which makes the cell-edge performance lower than the strategy with R_r= 166m.

9.3 Traffic steering and resource allocation for HetNets with CRE

Small cells in a HetNet help to offload traffic from macro cells. A Cell Range Extension (CRE) bias helps to tune the amount of offloading, and thus traffic redirection in the network. However, users receiving in downlink in the CRE area, suffer from strong interference from the macro cells, that without CRE, would be otherwise the serving cells. Inter-Cell Interference Coordination (ICIC) is a technique to deal with this problem, consisting in muting resources, in this case in the macro cells, to avoid interference to users in the CRE region. ICIC requires coordination of the different RAN nodes. We aim to solve disputes for resources in HetNets implementing ICIC at network-level. We propose a logical description of the RAN by Logical RAN Entities (LREs), under the control of C3, where each LRE is a set of cells in the HetNet. The state of the RAN is abstracted, and described by a network graph, with nodes representing LREs. We consider a HetNet composed by macro and pico cells, with CRE, transmitting in downlink.

9.3.1 Description of the RAN with a Network Graph

A network graph is constructed to manage the interference situation produced by the strong interference that macro base stations exert into small cell users in the CRE region. LREs can be created and removed dynamically, for example, after cell-user association. Each user associated with a macro cell *m* is further associated with a LRE (m). Each user associated with a small cell *s*, and with the presence of a set M of macro cell strong interferers, is associated with the LRE (s, M). If M is the empty set, then we denote (s,M) by (s), and if M = $\{m\}$ is a singleton, then we denote (s,M) by (s,m).

RAN network entities compute a utility for users, and a sum utility for LREs, from the spectral efficiencies reported by the users (see [Lembo2017] and [Lembo2017b, Chapters 2,3] for details). An adjacency matrix is created, which describes conflicts between LREs, and it is used to construct the network graph. Edges in the graph are established between LREs that have at least one common member. Figure 69 shows an example of a HetNet with three macro cells, two small cells, and associated users. An example of a network graph of logical entities derived for the HetNet is shown in the figure.

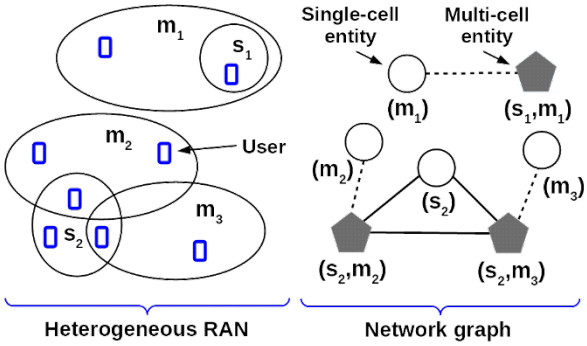


Figure 69 Example of a HetNet and network graph of logical entities

C3 is in charge of constructing the network graph, and implementing a suitable logic for the network coordination. For a given network graph, we say that a resource allocation pattern is conflict-free if two LREs that are connected by an edge are never assigned the same resource. Two different kinds of interactions between LREs generate edges in the network graph:

1. Conflict edge: for set of LREs that share the same physical RAN node (small cells), the resource usage pattern is orthogonal between the different LREs.

2. Weighted edge: if users of one LRE receive dominant interferences from another LRE, these LREs should avoid using the same radio resources. Here, in macro cells *m* that strongly interfere, some users in *s* are muted on the resources on which these users are served.

Resources are interchangeable among LREs respecting conflicts. In each macro cell, resources can be muted with an independent muting ratio. Resources for local scheduling in each LRE are assumed infinitely sub-divisible among the users in the LRE. The resource allocation problem is a multi-coloring problem of the graph. A Tabu Search (TS) algorithm is proposed for the COHERENT C3, to find optimal muting ratios through moves in the graph, subject to the constraints discussed below. The muting ratios are communicated to the cells. The RTCs determine locally the scheduling weights for the users, constrained to the muting ratios stated by the C3. Further details are available in COHERENT D3.2, and [Lembo2017].

9.3.2 Optimization algorithm for resource allocation

A centralized multi-coloring is performed. As the information is aggregated by C3, a global optimization problem can be constructed for resource allocation. We combine the network graph with a TS technique [Glover1993]. The TS algorithm progresses toward a maximum system utility by evaluating moves in the graph. The TS technique has shown good performance for this kind of problems [Hertz1987]. The implementation of the algorithm is summarized in Algorithm 23. A function $f_{AdjacencyMatrix}(.)$ returns the LREs that have conflicts with the selected LRE from the Adjacency Matrix. A function $f_{Moves}(.,.)$ returns the candidate moves for a given LRE. These are the moves to explore neighboring solutions around the LRE. We note that the move is initiated from one LRE, but due to the moves and constraints, there are many LREs participating in the move. Then, $f_{Constraints}(.,.)$ evaluates constrains as described below. The objective function to be maximized is the network utility $f_{Ut}(., .)$. The move is accepted if it maximizes the best utility known at that moment, then the decision is communicated to the participating LREs to update the allocation of resources. The algorithm can iterate

until a completion criterion is met, however, as the network is not static, a dynamic algorithm will iterate indefinitely, aiming to have a long term convergence.

Algorithm 23 Tabu Search in Network Graph

```
TabuList ← [], BestUtility ←−∞
 while completion criterion not met do
    SelectedLRE ← Select a LRE at random
   Conflicts ← f_AdjacencyMatrix(SelectedLRE)
   MovesList ← f_Moves(SelectedLRE)
   for CandMov in MovesList do
      BestMov ← null
      if f_Ut(CandMov,Conflicts) > f_Ut(BestMov,Conflicts) and
         CandMov is not in TabuList and
         f_Constraints(SelectedLRE,Conflicts) respected then
         BestMov ← CandMov
      end if
   end for
   SelectedCandidateMove ← BestMov
   if f_Ut(SelectedCandidateMove) > BestUtility then
      Accept SelectedCandidateMove
      BestUtility ← f_Ut(SelectedCandidateMove)
   end if
   update TabuList, remove older entry
end while
```

The constraints for the moves in the TS algorithm are given by different types of cliques in the graph, as follows.

A) Constraint coming from weighted edges: for a given small cell *s* and a given macro cell *m*, the LREs (s,M) with $m \in$ M, can only be served by resources on which *m* is muting.
B) Constraint coming from conflict edges: for any small cell, the sum of resources used in the LRE (s,M) and the LRE (s), cannot be greater than 1, (maximal number of resources used in a small cell).
C) A LRE with no resources cannot decrement resources, and a LRE with the maximum available resources cannot increment resources.

Resource-moves aim to optimize the muting ratios for a given cell association, resulting in higher data rates. Moves started at LREs (m) and (s,M):

1) LRE (m) mutes: at most one (s,M) per small cell *s* increments resources if the other (m)s with edges to (s,M) are muted enough, respective (s)s decrement resources.
2) LRE (m) un-mutes: one (s,M) per small cell decrements resources, respective (s)s increment resources.
3) LRE (s,M) increments resources: (m)s with edges to (s,M) are muted, (s) decrement resources.
4) LRE (s,M) decrements resources: (m)s with edges to (s,M) are un-muted, if other (s,M)s with edges to (m) are using an amount of muted resources as the total to remain muted, (s) increments resources.

Function $f_{Moves}(.,.)$ returns moves 1) and 2) for a LRE (m), and 3) and 4) for a LRE (s,M).

9.3.3 Simulation results

We tested the algorithm in a HetNet as described in [Lembo2017], performing Monte Carlo simulations. Figure 70 shows the simulation results for two HetNets with (1) M = 6 macro cells, 42 small cells and U=360 users, and (2) M = 12 macro cells, 48 small cells and U=720 users. Performance is evaluated in terms of the CDF of average user rate. We compare the performance of the TS algorithm against the cases of no muting and fixed muting in the whole network. We observed almost no gain in the system

average user rate, but up to 77% gain per user in the 5th percentile, compared to fixed muting (see [Lembo2017] for details).

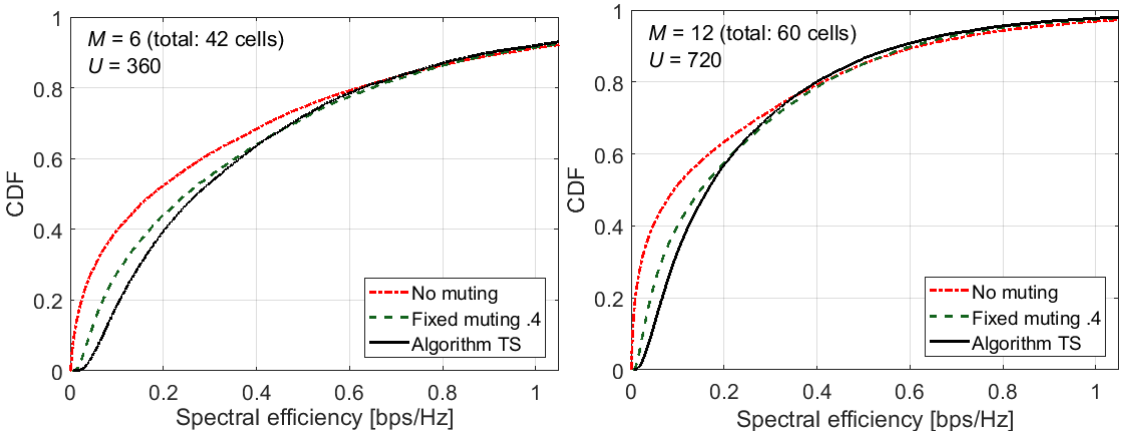


Figure 70 Simulation results for a HetNet with (1) M=6 macro cells, 43 small cells and U=360 users, and (2) M=12 macro cells, 48 small cells and U=720 users.

9.4 Conclusions

9.4.1 Technical/Scientific Impact

In the first subsection, we have considered a hierarchical control framework to address these network management problems related to mmWave & sub-6GHz multi-connectivity. The computation overhead for network optimization is reduced by dividing the optimization problem into two levels, i.e. a centralized cell association problem and the local resource allocation problems. System level simulation in urban micro-cell scenarios illustrates that using mmWave & sub-6GHz multi-connectivity can increase the cell-edge performance significantly, which can satisfy the 5G requirement of consistent user experience. In the second subsection, have considered two-hop D2D relaying to enhance end-to-end throughput in multi-cell downlink networks by offloading the traffic of direct DL transmission to a two two-hop D2D relaying transmission. To capture the complicated interference interaction in the multi-cell networks with D2D relaying, we have proposed an analytical model with several parameters, including a minimum relaying distance and a relaying probability. These parameters are optimized for network-level management. Simulation and analytical results are provided to understand the interference characteristic and throughput performance when D2D relaying is applied. In the third subsection, we discussed cell offloading with CRE in a HetNet, and the need of coordination with ICIC for muting macro cells. We proposed a logical description of the RAN by Logical RAN Entities, and described the state of the RAN by a network graph, to solve disputes for resources in the HetNet at network-level. Results show the feasibility of performing network-level coordination with the proposed network graph and TS algorithm. We observed positive gains for users in the lower percentiles of the users' rates, without loss in system average user rate.

9.4.2 Feedback toward development

The traffic steering techniques presented in this section work upon the abstracted network graphs which do not depend on the details of the mobile network implementation. The network graphs are constructed based on semi-dynamic network statistics which can be retrieved based on currently available network measurements such as CQI. The C3 algorithms can be developed using general virtual machine running on different hardware platforms. For traffic steering based on D2D relaying, D2D channel estimation and D2D interference estimation are required, which are not yet available in current LTE networks. Its development is thus heavily depending on the progress of D2D standardization. The multi-connectivity algorithm in 5G mmWave network requires very basic measurements, it can be readily developed in future 5G mmWave networks once they are implemented and operated.

9.4.3 Expected business impact

Operators can benefit from the traffic steering techniques presented in this section both in saving OPEX and meeting the increasing data rate requirements. They can increase the user experience consistency by implementing these techniques, which is a fundamental KPI in the 5G network.

10. Algorithms for Flexible Functional Splits in 5G Network

10.1 Overview

5G networks are expected to support various applications with diverse requirements in terms of latency, data rates and traffic volume. Cloud–RAN and densely deployed small cells are two of the tools at disposal of Mobile Network Operators to cope with such challenges. In order to mitigate the fronthaul requirements imposed by the Cloud–RAN architecture, several functional splits, each characterized by a different demarcation point between the centralized and the distributed units, have emerged. However, the selection of the appropriate centralization level (i.e., the functional split) still remains a challenging task, since a number of parameters have to be considered in order to make such a decision.

As it has been mentioned, the higher is the split point, the more are the centralization benefits. In order to reap the benefits of different functional splits, the Mobile Network Operators (MNOs), who currently own LTE/LTE-A mobile networks, should at some point migrate to the Cloud–RAN architecture. While migrating to the Cloud–RAN architecture, the main objective for them would be to minimize the required investments. To this end, their available network infrastructure (e.g., site locations, transmission links) could be used while finding optimal locations for CU pools and designing the fronthaul network.

The main drawbacks of the PHY–RF split lie in the tight bandwidth and latency requirements (see Table 2) imposed on the fronthaul (i.e. the network interconnecting CUs with DUs) where protocols like the Common Public Radio Interface (CPRI) [cpri] are typically used to carry the I/Q samples; for example, a 20 MHz LTE FDD channel using a 2x2 MIMO antenna configuration can result in a CPRI rate of $\approx$ 2.5Gbps. Recent advances in microwave communications allow for up to a few Gbps of bandwidth over short distances (less than one Km) to be employed in the E–band (70-80 GHz), making it suitable as fronthaul technology for dense small cells.

In [Harutyunyan18mig], we consider real data (e.g., hourly data traffic demand at each carrier of each eNB for one month, the site locations, the transport network and other network components) of an operation LTE–A network to solve the CU placement problem that minimizes the capital and operational expenses required to transit to Cloud–RAN, making sure that at each moment users traffic demand is satisfied. However, only PHY–RF split has been considered in the Cloud–RAN architecture, leaving the other splits to be examined in the feature work.

In [Harutyunyan16] and [Riggio16], we study the CU placement problem over a reconfigurable wireless fronthaul network for only the PHY–RF split. Then, these works have been extended in [Harutyunyan18mmw] by considering also the PHY split, the MAC split and the PDCP/RLC split in the CU placement problem. However, in the considered virtual network embedding context, the functional split selection for a small cell is static. In other words, there is no possibility for MNOs, depending upon some KPIs, to select different functional splits for a same small cell at different time in a day.

In this section, we formalize and solve a virtual network embedding problem (VNE) for 5G networks supporting different functional split options [Harutyunyan17, Harutyunyan18flex]. We formulate the problem as an integer linear programming (ILP) problem in which virtual network requests are received from mobile virtual network operators (MVNOs) and are embedded by the infrastructure providers (InPs), having an objective of dynamically selecting the appropriate functional split option for each small cell at each time that can enable MNOs to jointly minimize the network–wide inter–cell interference and the fronthaul bandwidth utilization. This work extends our previous study [harutyunyan17] by (i) proposing dynamic and static ILP–based algorithms, and a scalable dynamic and static heuristics called, respectively, ILP–DM, ILP–ST, HEU–DM and HEU–ST, to solve the VNE problem, (ii) discussing the pros and cons and the applicability of each algorithm.

10.2 Flexible functional split in 5G Networks

Compared to LTE and LTE–Advanced, 5G networks are expected to deliver a 1000 time increase in the system capacity, reduced round–trip delay, and enhanced performance of the cell–edge users. MNOs are using network densification as an efficient way to meet the aforementioned goals [nokia]. Albeit the usage of smaller cells has a number of advantages (e.g., decreased distance between nodes, reduced path loss and transmission power, higher frequency reuse factor), it poses also several challenges (e.g., increased total cost of ownership, increased power consumption, more frequent handovers, increased level of interference). By far, the most obvious downside of densely deployed small cells is that it dramatically increases the level of inter–cell interference, which may result in a significant performance degradation unless interference mitigation techniques are used.

Recent advances in Network Functions Virtualization (NFV) enabled MNOs to transit from the fully–decentralized RAN (D–RAN) architecture, where baseband processing and radio elements are co–located, to the fully–centralized Cloud–RAN (PHY–RF split) architecture [cran], where baseband units are decoupled from the radio elements (termed Distributed Unit – DU) and consolidated in large datacenters (termed Centralized Unit – CU). By decoupling baseband processing from the radio elements, C–RAN can lower the total cost of ownership for MNOs.

The C–RAN and D–RAN architectures are two extreme concepts, both with advantages and disadvantages. In fact, while D–RAN requires relatively low backhaul capacity, it does not allow for joint signal processing. Conversely, C–RAN enables joint signal processing techniques, such as coordinated multi-point transmission (CoMP), at the price of higher backhaul requirements (e.g., bandwidth, latency). In order to tackle the aforementioned challenges, a number of intermediate functional splits, each characterized by a different demarcation point between the centralized and the distributed units, have been proposed. Different criteria have to be considered in order to select the appropriate functional split. Following the current galloping pace in the mobile data traffic demand, it is our standpoint that implementing a fixed functional split is not a viable solution in the long run. Therefore, considering the mobile traffic demand and the daily traffic variations, the flexibility of dynamically choosing the optimal functional split is essential in order to efficiently employ the fronthaul bandwidth and baseband processing resources.

10.3 Network Model

This section details the substrate and virtual network models. Figure 71 depicts the reference network architecture used in this work. The main idea of this figure is to show that different functional splits can co–exist at the same network and can be changed dynamically. In the lower left part of the figure we can see the traditional D–RAN architecture in which the DU and the CU are deployed in close proximity. As opposed to the D–RAN case, for all the other functional splits, the DUs are decoupled from the CUs, and an optical[4] fronthaul is used for their interconnection. It is important to say that, regardless of the split option being employed at the given time, apart from the CU pool, also all the DUs possess processing capabilities since we consider a network in which the functional split options can be flexibly changed. For example, although in the case of employing the PHY–RF split there is no baseband signal processing at the DUs, the DUs however have to have baseband signal processing units since after a few hours the MNO, based on some KPIs, might decide to employ another split option such as the PHY split or the MAC split in which cases some part of the baseband signal processing is taking place at the DUs.

Substrate Network Model: Let $G_s = (N_s, E_s)$ be an *undirected* graph modelling the physical network, where $N_s = N_s^1 \cup N_s^2$ is the set of $n_1 = |N_s^1|$ DUs and $n_2 = |N_s^2|$ CU pools, and $E_s \subseteq N_s^1 \times N_s^2$ is the set of fronthaul links. An edge $e^{nm} \in E_s$ if a connection exists between $m, n \in N_s$. Three weights, $\omega_{ant}^s(n)$, $\omega_{prb}^s(n)$ and $\omega_{prc}^s(n)$ are assigned to each node $n \in N_s : \omega_{ant,prb,prc}^s(n) \in \mathrm{N}^+$ representing, respectively, the number of RF front–ends, the number of physical resource blocks (PRBs)

[4] Optical fiber, as the most common fronthauling option, has been selected as the fronthaul medium. However, other fronthauling option such as millimetre wave wireless links or cooper link are also possible.

and the processing capacity supported by the nodes. Each substrate node is also associated with a geographic location $loc(n)$, as x, y coordinates and a coverage radius $\delta(n)$,in meters, indicating the coverage area of the small cell centered on DU n. Another weight $\omega^s_{bwt}(e^{nm})$ is assigned to each link $e^{nm} \in E_s : \omega^s_b(e^{nm}) \in \mathrm{N}^+$ representing the capacity (in Gbps) of the link connecting the two nodes. Table 23 summarizes the substrate network parameters.

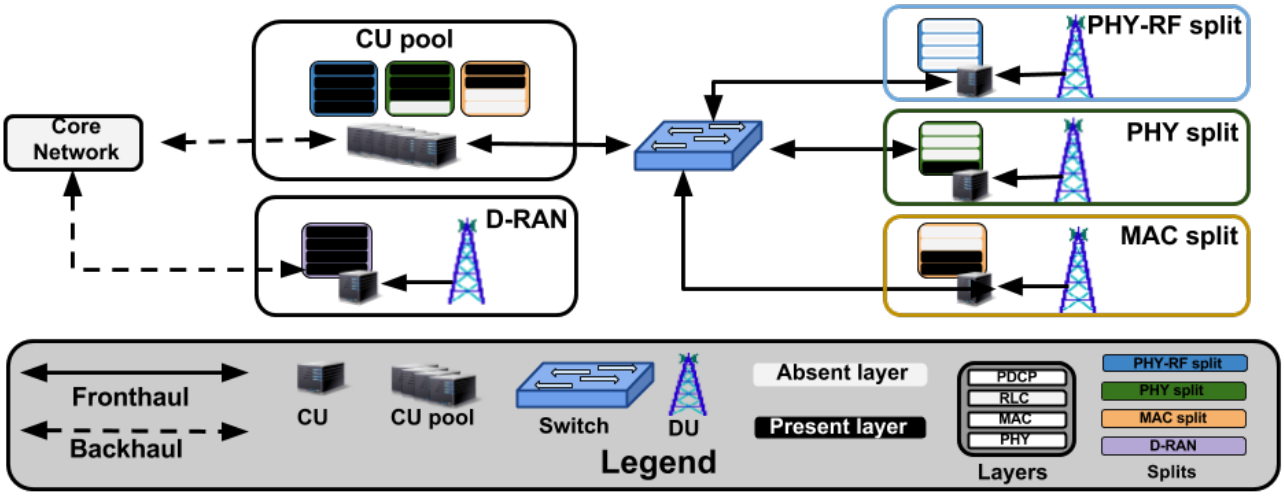


Figure 71 - The different functional splits that can co–exist at the CU pool. Notice that, apart from the CU pool, also all the DUs possess processing capabilities regardless of the functional split option being employed at the considered time.

Table 24 - Substrate network parameters.

Variable	Description
G_s	Substrate network graph.
N_s	Substrate nodes in G_s.
N_s^1	Substrate DUs in G_s.
N_s^2	Substrate CU pools in G_s.
E_s	Substrate links in G_s.
$I(m)$	Acceptable interference level for the m^{th} functional split.
$\omega^s_{ant}(n)$	Number of RF front–ends available at DU $n \in N_s^1$.
$\omega^s_{prb}(n)$	Number of PRBs available at DU site $n \in N_s^1$.
$\omega^s_{prc}(n)$	The processing capacities of the nodes $n \in N_s$.
$\omega^s_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_s$ (in Gbps).
$loc(n)$	Geographical location of node $n \in N_s$.
$\delta(n)$	Coverage radius of DU $n \in N_s^1$ (in meters).
R_n	The set of fronthaul bandwidths of node $n \in N_s$.

Notice how, it is our assumption that the CU pool is equipped with enough computational capacity to support all DUs employing the highest possible functional split i.e., the PHY–RF split, which requires all baseband signal processing to take place at the CU pool. Whereas, the DUs are equipped with enough computational capacity to process the signals with the lowest possible functional split, i.e. the MAC split. The DUs are also equipped with RF front–ends and, as opposed to the CU pool, are connected to the CU pool by means of optical fronthaul links.

Virtual Network Model: There are different approaches to model virtual network requests, from resource–based [Zaki10] [Zaki11] to throughput–based models [Kokku12]. In this work, we use a resource–based model in which MVNOs can request one or more small cells with a particular antenna configuration and a fixed amount of PRBs to be allocated to their small cells. This model does not provide any throughput guarantees to the MVNO's users whose performances can be affected by users distribution and by the time varying nature of the wireless channel.

Virtual network requests are modelled as *undirected* graphs $G_v = (N_v, E_v)$ where $N_v = N_v^1 \cup N_v^2$ is the set of $n_1 = |N_v^1|$ virtual DUs and $n_2 = |N_v^2|$ virtual CUs, and $E_v \subseteq N_v^1 \times N_v^2$ is the set of virtual fronthaul links. Notice that MVNOs do not request processing resource capacity at the CUs or DUs. Nether do they request fronthaul bandwidth. Nodes in the virtual network requests have two weights $\omega_{ant}^v(n)$ and $\omega_{prb}^v(n)$ indicating, respectively, the number of RF front–ends and the number of PRBs requested for each DU $n \in N_v^1$.

Table 25 - Virtual network parameters.

Variable	Description
G_v	Virtual network request.
N_v	Virtual nodes in G_v.
N_v^1	Virtual DUs in G_v.
N_v^2	Virtual CUs in G_v.
E_v	Virtual links in G_v.
$\omega_{ant}^v(n)$	RF front–ends required at DU $n \in N_v^1$.
$\omega_{prb}^v(n)$	PRBs required at DU $n \in N_v^1$.
$\omega_{prc}^v(n)$	Processing capacity required at node $n \in N_v$.
$\omega_{bwt}^s(e^{nm})$	Capacity required for link $e^{nm} \in E_v$ (in Gbps).
$loc(n)$	Desired geographical location for DU $n \in N_v^1$.

Given the chosen functional split and considering the aforementioned parameters ($\omega_{ant}^v(n)$, $\omega_{prb}^v(n)$), the fronthaul bandwidth required to support a given small cell can easily be derived [scf]. Each virtual DU $n \in N_v^1$ is also associated with a geographic location $loc(n)$, as x, y coordinates. This information together with the substrate node location and its coverage radius is used to express how far a virtual DU $n \in N_v^1$ can be placed from the preferred location specified by $loc(n)$. Table 24 summarizes the virtual network request parameters.

For example, if the virtual DU $n \in N_v^1$ requests $\omega_{ant}^v(n)$ = *2 (2×2MIMO)* RF front–ends and $\omega_{prb}^v(n)$ = *50* PRBs, this translates to the fronthaul bandwidth of $\omega_{bwt}^s = 2.4$ Gpbs, $\omega_{bwt}^s = 0.54$ Gpbs and $\omega_{bwt}^s = 0.046$ Gpbs, respectively, in the cases of the PHY–RF split, the PHY split and the MAC split, assuming the transport block size of 21384 bits in an LTE network with 20 MHz bandwidth. Notice that as opposed to the PHY split and the MAC split, the fronthaul bandwidth for the PHY–RF split does not depend on the requested number of PRBs.

10.4 Virtual Network Embedding

In this section, the virtual network embedding problem is introduced followed by the presentation of dynamic and static ILP–based algorithms (ILP–DM and ILP–ST), and scalable dynamic and static heuristics (HEU–DM and HEU–ST). Upon arrival of a new virtual network request, the substrate network must decide whether it is to be accepted or rejected. The embedding process consists of two steps: node embedding, and link embedding. In the first step (node embedding), each virtual node in the request is mapped to a substrate node. In the second step (link embedding), each link is mapped to a single substrate path. In both cases some constraints must be satisfied.

A dynamic ILP--based placement algorithm (ILP–DM): In this subsection, the ILP–DM algorithm is presented. ILP–DM employs a dynamic embedding strategy, meaning that with the arrival of a new virtual network request, the request along with the ones that have been previously embedded are re–embedded. Thus, with every embedding, the optimal embedding solution is found for all the requests. Every virtual DU $n' \in N_v^1$ in the virtual request has a desired location $loc(n')$. Whereas, every

substrate DU $n \in N_s^1$ has both a location $loc(n)$ and a coverage radius $\delta(n)$. For each virtual DU n', we can then define the cluster of candidates $\Omega(n')$ to which the virtual DU $n' \in N_v^1$ can be mapped:

$$\Omega(n') = \{n \in N_s^1 \mid dis\left(loc(n), loc(n')\right) \leq \delta(n)\} \qquad (1)$$

We can now provide the ILP formulation for our VNE problem. The objective of this formulation is to minimize the inter–cell interference at each DU and, at the same time, minimize the fronthaul bandwidth required to serve the request. The chosen objective function is:

$$min\left(\sum_{n' \in N_v^1} \sum_{n \in \Omega(n')} \sum_{n^* \in \Omega(n)} \sum_{p,p^* \in \omega_{prb}^s(n) \times \omega_{prb}^s(n^*)}^{p=p^*} \Phi_p^n \, \Phi_{p^*}^{n^*} + \sum_{n' \in N_v^1} \sum_{n \in N_s} \sum_{m \in R_n} R_n(m) \, \Phi_{n,m}^{n'}\right) \qquad (2)$$

where (with a slight abuse of notation) we use $n^* \in \Omega(n)$ to indicate that a node $n^* \in N_s^1 (n^* \neq n)$ has overlapping radio coverage with the candidate substrate node $n \in \Omega(n')$ (i.e. an interfering node). Moreover, $\Phi_p^n, \Phi_{p^*}^{n^*} \in \{0,1\}$ are binary variables, indicating whether the PRBs p, p^* are in use at the substrate nodes n and n^*, respectively.

The first term in the objective function aims at minimizing the number of overlapping PRBs, while the second term minimizes the fronthaul bandwidth requirements by considering the first term and selecting the most optimal functional split for each host small cell. The rationale behind this approach is that different functional splits can enable different interference management techniques [ltebook] and thus a trade–off exists between fronthaul bandwidth requirements and the level of acceptable interference in the system. It is important to mention that, given the PRB requirement of a virtual node, the fronthaul bandwidth requirement R is fixed for each m^{th} functional split, and its selection (the second term in the objective function) depends on the interference level (the first term). Thus, no weight is required for those terms because of their non homogeneity.

In order to minimize the inter–cell interference, we first need to quantify it. For each small cell/DU, a collision domain is selected, containing all the small cells with which the considered small cell has an overlap in the coverage area. In essence, these would be the small cells whose signals in the downlink may interfere with the downlink signals of the considered small cells. The inter–cell interference[5] can then be estimated based on the information of the PRB chunks allocation to the virtual network requests (i.e., MVNOs).

Note that the objective function contains a quadratic term $\Phi_p^n \, \Phi_{p^*}^{n^*}$ that results in a standard (non–convex) quadratic formulation. To linearize this term, we define a variable Φ_{p,p^*}^{n,n^*} and substitute it to the quadratic term in the objective function:

$$\Phi_p^n \, \Phi_{p^*}^{n^*} \approx \Phi_{p,p^*}^{n,n^*} = \begin{cases} 1 \;\; if \;\; \Phi_p^n = \Phi_{p^*}^{n^*} = 1 \\ 0 \;\; otherwise \end{cases} \qquad (3)$$

We will now detail the constraints used in the ILP formulation. The following constraint ensures that if a same PRB is being used by two or more small cells that are in the same collision domain, then a penalty is applied to each of the small cell that uses the PRB:

$$\Phi_p^n + \Phi_{p^*}^{n^*} - \Phi_{p,p^*}^{n,n^*} \leq 1 \quad \forall n^* \in \Omega(n'), \;\; \forall \mathrm{p} = p^*, \;\; p, p^* \in \omega_{prb}^s(n) \times \omega_{prb}^s(n^*) \qquad (4)$$

[5] In this study, the interference estimation is based on the assumption that the PRBs allocated to MVNOs are always in use. We do not consider the wireless users, and therefore, do not model their channels. More accurate interference estimation is, however, possible taking into account the PRB allocation of each user and their channel quality information.

In essence, $\Phi_{p,p^*}^{n,n^*} = 1$ indicates the presence of the inter–cell interference at $n^* \in \Omega(n')$ and at $n \in \Omega(n')$ substrate DUs. The penalty (i.e., inter–cell interference) increases with an increase in the number of overlapping PRBs used by the small cells belonging to the same collision domain. The next constraints deal with the resources (e.g., the processing resource of the DUs and the CU, fronthaul bandwidth, number
of RF front–ends and the number of PRBs) required to embed the requests, making sure that those resources are at most equal to the resources available at the substrate nodes:

$$\sum_{n' \in N_v^1} \omega_{prc}^v(n')\, \Phi_n^{n'} \leq \omega_{prc}^s(n) \qquad \forall n \in N_s^1 \qquad (5)$$

$$\sum_{n' \in N_v^2} \omega_{prc}^v(n')\, \Phi_n^{n'} \leq \omega_{prc}^s(n) \qquad \forall n \in N_s^2 \qquad (6)$$

$$\sum_{e' \in E_v} \omega_{bwt}^v(e')\, \Phi_e^{e'} \leq \omega_{bwt}^s(e) \qquad \forall e \in E_s \qquad (7)$$

$$\sum_{n' \in N_v^1} \omega_{ant}^v(n')\, \Phi_n^{n'} \leq \omega_{ant}^s(n) \qquad \forall n \in N_s^1 \qquad (8)$$

$$\sum_{n' \in N_v^1} \omega_{prb}^v(n')\, \Phi_n^{n'} \leq \omega_{prb}^s(n) \qquad \forall n \in N_s^1 \qquad (9)$$

where $\Phi_n^{n'}$, $\Phi_e^{e'} \in \{0,1\}$ are binary variables indicating whether the virtual node $n' \in N_v$ and virtual link $e' \in E_v$ has been mapped, respectively, to the substrate node $n \in N_s$ and substrate link $e \in E_s$.

It is worthwhile to note that the processing as well as the fronthaul bandwidth resources are not requested by MVNOs. Conversely, they depend on the actual functional split selected in order to on–board the virtual request.

Each requested virtual node $n' \in N_v$ must be mapped only once (10), while each virtual DU $n' \in N_v^1$ must be mapped only on a substrate DU that belongs to its cluster of candidates (11):

$$\sum_{n \in N_s} \Phi_n^{n'} = 1 \qquad \forall n' \in N_v \qquad (10)$$

$$\sum_{n \in N_s^1 \setminus \Omega(n')} \Phi_n^{n'} = 0 \qquad \forall n' \in N_v^1 \qquad (11)$$

The next constraint prevents the re–allocation of PRBs, making sure that each PRB is allocated maximum once:

$$\sum_{n' \in N_v^1} \Phi_{n,p}^{n'} \leq 1 \qquad \forall n \in N_s^1,\ \forall p \in \omega_{prb}^s(n) \qquad (12)$$

where $\Phi_{n,p}^{n'} \in \{0,1\}$ is a binary variable showing whether the PRB $p \in \omega_{prb}^s(n)$ of the substrate DU $n \in N_s^1$ has been allocated to the virtual DU $n' \in N_v^1$.

Virtual DU embedding and PRBs allocation must be consistent, meaning that if a virtual DU has been mapped to a given substrate DU then only the PRBs of that substrate DU must be allocated to the virtual DU:

$$\sum_{p \in \omega_{prb}^s(n)} \Phi_{n,p}^{n'} - \omega_{prb}^v(n') \Phi_n^{n'} = 0 \qquad \forall n' \in N_v^1,\ \forall n \in \Omega(n') \qquad (13)$$

In order to compute the fronthaul bandwidth requirement for the virtual DU $n' \in N_v^1$, each virtual small cell it has to be mapped to one and only one functional split:

$$\Phi_n^{n'} - \sum_{m \in R_n} \Phi_{n,m}^{n'} = 0 \qquad \forall n' \in N_v^1, \quad \forall n \in \Omega(n') \qquad (14)$$

where R_n is the set of the fronthaul bandwidths that would be required to support the request using the considered functional splits, and its index m indicates the functional split option. $\Phi_{n,m}^{n'}$ is a binary variable indicating whether the virtual DU $n' \in N_v^1$ has been mapped to the m^{th} functional split option of substrate DU $n \in \Omega(n')$.

The next constraint ensures that only one functional split is selected for a given substrate DU. This constraint along with the previous one (14) makes sure that all the virtual DUs that have been mapped to a same substrate node have selected the same functional split of the substrate DU:

$$\sum_{n' \in N_v} \Phi_{n,m}^{n'} - \sum_{n' \in N_v} \Phi_n^{n'} \leq 0 \qquad \forall n \in N_s, \quad \forall m \in R_n \qquad (15)$$

Finally, the last constraint handles the optimal functional split selection for each small cell:

$$\sum_{n^* \in \Omega(n)} \sum_{p,p^* \in \omega_{prb}^s(n) \times \omega_{prb}^s(n^*)}^{p=p^*} \Phi_{p,p^*}^{n,n^*} \leq I(m) \sum_{n^* \in \Omega(n)} \omega_{prb}^s(n^*) \quad \forall n' \in N_v^1, \forall n \in \Omega(n'), \forall m \in R_n \quad (16)$$

where $I(m)$ represents the acceptable inter–cell interference level for each m^{th} functional split option (see Table 25). This constrain effectively puts an upper bound on the number of acceptable overlapping PRB allocations. For example, for the PHY–RF split we are willing to accept as many PRB allocation overlaps as the number of PRBs in a collision domain. This essentially results in a reuse factor of 1, which is acceptable since the PHY–RF split enables several advanced interference mitigation techniques to be employed. Conversely, as the functional split moves up in the protocol stack we reduce the maximum number of allowed overlaps in the PRBs allocation.

Table 26 - DU and CU relative processing capabilities and the acceptable inter–cell interference level for the considered functional splits.

Split	$I(m)$	Processing Capacity	
		DU	CU
PHY–RF split	1	$0 \cdot \omega_c^v(n)$	$1 \cdot \omega_c^v(n)$
PHY split	0.6	$0.5 \cdot \omega_c^v(n)$	$0.5 \cdot \omega_c^v(n)$
MAC split	0.3	$0.7 \cdot \omega_c^v(n)$	$0.3 \cdot \omega_c^v(n)$

A static ILP--based placement algorithm (ILP–ST): The ILP–ST algorithm resembles the ILP–DM algorithm in terms of the fact that both have the same objective function and that all the constrained defined for the ILP–DM algorithm are held also for the ILP–ST algorithm. However, the difference between them is that, in the case of the ILP–ST algorithm, as opposed to the ILP–DM one, the virtual network request are embedded sequentially. In other words, with the arrival of a new virtual network request only that request is embedded. Thus, as the name of the algorithm implies, a static embedding is considered. The ILP–ST algorithm can be used to solve larger embedding problems since the static embedding makes it significantly faster compared to its dynamic counterpart.

A scalable dynamic placement heuristic (HEU–DM): The ILP–DM VNE algorithm becomes computationally intractable as the size of the substrate network and/or of the virtual network requests increases. For example, the ILP–DM algorithm takes one week on Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using the Matlab ILP solver (intlinprog) to map a virtual network request, having 20

virtual nodes, to a substrate network with 20 nodes. In order to address this scalability issue, we also propose the HEU–DM heuristic, which is able to embed the same virtual network request in a limited amount of time. Like in the case of the ILP–DM algorithm, in this case also a dynamic embedding is considered.

The proposed HEU–DM heuristic is composed of four steps. In the first step, a cluster of candidate DUs $candidates(n')$ is selected for each virtual DU $n' \in N_v^1$ by considering its desired location. Then, for each candidate substrate DU $m \in candidates(n')$ of each virtual DU $n' \in N_v^1$, a neighbor list is created $candidates(n')(m)$.

Algorithm 1 HEU–DM

procedure *Input:*(G_s, G_v)
  Step 1: Find the list of candidates and their neighbors.
  for $n' \in N_v^1$ **do**
    for $n \in N_s^1$ **do**
      $d \leftarrow dis(loc(n'), loc(n))$
      if $d \leq \delta(n)$ **then**
        $candidates(n') \leftarrow n$
      end if
    end for
    for $m \in candidates(n')$ **do**
      for $n \in N_s^1$ **do**
        $d \leftarrow dis(loc(m), loc(n))$
        if $n \neq m$ and $d \leq 2\delta(n)$ **then**
          $neighbor(n')(m) \leftarrow n$
        end if
      end for
    end for
  end for
  Step 2: Find all possible combinations of candidates.
  for $n' \in N_v^1$ **do**
    $combs\_cands \leftarrow combvec(combs\_cands, candidates(n'))$
  end for
  Step 3: Find the best combinations of candidates (minimum interference).
  $min\_intf \leftarrow \infty$
  for $comb \in combs\_cands$ **do**
    $substrate\_resources\_copy \leftarrow substrate\_resources$
    $intf\_all \leftarrow 0$
    for $n' \in N_v^1$ **do**
      $cand \leftarrow comb(n')$
      $intf(cand) \leftarrow 0$
      if $\omega_a^v(n') < \omega_a^s(cand)$ or $\omega_{prb}^v(n') < \omega_{prb}^s(cand)$ **then**
        break
      end if
      for $neigh \in neighbor(n')(cand)$ **do**
        for $prb\_idx \in prb(k)$ **do**
          if $neigh(prb\_idx) = cand(prb\_idx) = 1$ **then**
            $intf(cand) = intf(cand) + 1$
          end if
        end for
      end for
      $intf\_all = intf\_all + intf(cand)$
      Update $substrate\_resources\_copy$
    end for
    if $intf\_all \leq min\_intf$ **then**
      $min\_intf \leftarrow intf\_all$
      $best\_cands \leftarrow comb$
    end if
  end for
  Step 4: Allocate resources & select functional splits.
  for $n' \in N_v^1$ **do**
    $mapped(n') \leftarrow best\_cands(n')$
    $sorted(best\_cands(n')) \leftarrow sortprb(intf(best\_cands(n'))\uparrow)$
    $alloc\_prb(n') \leftarrow sorted(best\_cands(n'))[1 : rqst\_prb(n')]$
    for $s \in splits$ **do**
      if $intf(best\_cands(n')) \in intf\_bounds(s)$ **then**
        $split(best\_cands(n')) \leftarrow splits(s)$
        $spt \leftarrow split(best\_cands(n'))$
        break
      end if
    end for
    $fh\_band(n') \leftarrow compute\_band(\omega_{prb}^v(n'), \omega_a^v(n'), spt)$
  end for
end procedure

Algorithm 2 HEU–ST

procedure *Input:*(G_s, G_v)
  Step 1: Compute a list of candidates.
  for $n' \in N_v^1$ **do**
    $candidates(n') \leftarrow \emptyset$
    for $n \in N_s^1$ **do**
      $d \leftarrow dis(loc(n'), loc(n))$
      if $d \leq \delta(n)$ **then**
        if $\omega_a^v(n') \leq \omega_a^s(n)$ and $\omega_{prb}^v(n') \leq \omega_{prb}^s(n)$ **then**
          $candidates(n') \leftarrow n$
        end if
      end if
    end for
    if $candidates(n') = \emptyset$ **then**
      break
    end if
    Step 2: Select the small cell at which the inter–cell interference is minimum.
    $min\_intf \leftarrow \infty$
    for $m \in candidates(n')$ **do**
      for $n \in N_s^1$ **do**
        $d \leftarrow dis(loc(m), loc(n))$
        if $n \neq m$ and $d \leq 2\delta(n)$ **then**
          $neighbor(m) \leftarrow n$
        end if
      end for
      $intf(m) \leftarrow 0$
      for $p \in neighbor(m)$ **do**
        for $q \in prb(p)$ **do**
          if $p(q) = m(q) = 1$ **then**
            $intf(m(q)) = intf(m(q)) + 1$
          end if
        end for
      end for
      if $intf(m) \leq min\_intf$ **then**
        $min\_intf = intf(m)$
        $best\_cand \leftarrow m$
      end if
    end for
    Step 3: Allocate resources & select functional splits.
    $mapped(n') \leftarrow best\_cand$
    $sorted(m) \leftarrow sortprb(intf(m)\uparrow)$
    $alloc\_prb(n') \leftarrow sorted(m)[1 : rqst\_prb(n')]$
    for $s \in splits$ **do**
      if $intf(m) \in intf\_bounds(s)$ **then**
        $split(m) \leftarrow splits(s)$
        break
      end if
    end for
    $fh\_band(n') \leftarrow compute\_band(\omega_{prb}^v(n'), \omega_a^v(n'), split(m))$
    Update substrate resources
  end for
end procedure

In the second step, by using $combvec()$ function in Matlab, a combination matrix $comb\_cands$ is created, covering the entire search space of candidates combination for all the virtual DUs. Each column of this matrix represents one combination of candidate substrate DUs (one candidate substrate DU per virtual DU). The third step aims at finding the best combination of candidates (i.e., best column); that is, the one that after mapping the virtual DUs upon would introduce the least level of inter–cell interference in the network. Specifically, each combination $comb$ of the candidate vectors is

considered, and for each candidate substrate DU of each virtual DU, antenna and PRB resource availability is checked in order to find out whether the considered substrate DU is capable of supporting the resource requirements of the virtual DU. The total inter–cell interference is then estimated by accumulating the inter–cell interference at each candidate substrate DU in $comb$ vector, which is computed by checking the usage of each pair of PRB on the candidate $cand$ and its each neighbor $neigh$. At the end of this step, the combination vector $best\_cands$ is picked that would introduce the minimum level of network–wide inter–cell interference.
In the last step, each virtual $n \in N_v^1$ is mapped to its corresponding candidate substrate DU in $best\_cands(n')$. The PRBs of the host DUs are sorted in the ascending order of likelihood in terms of interference, and then the requested number of PRBs $\omega_{prb}^{v}(n')$ are allocated starting from the one that would introduce the least level of inter–cell interference in its collision domain. After the PRBs have been allocated, the overall inter–cell interference level at the host DU is estimated and the appropriate functional spit is selected. This is followed by computing the fronthaul bandwidth required to host the virtual DU by using $compute\_band()$ function, providing inputs the requested number of PRBs, the number of RF front–ends and the split option of the host substrate DU. Denoting $m = |N_s^1|$ and $n = |N_v^1|$ as the number of, respectively, substrate and virtual DUs considered in this study, the overall time complexity of HEU – DM is $\approx O(n(m+1))$ by considering that the number of splits is constant, and the number of candidates and theirs neighbours is invariable and independent from the substrate and virtual network size.

A scalable static placement heuristic (HEU–ST): The scalability might become a problem also for the HEU–DM heuristic when big–sized substrate/virtual networks with a few hundreds of substrate/virtual nodes are considered. In order to address this problem, we also propose a real–time HEU–ST heuristic. As the name suggests, this heuristic embeds statically the virtual network request. In other word, with the arrival if a new virtual network request, only that request is embedded. The proposed heuristic is composed of three steps.

In the first step, a cluster of candidate DUs is selected for each virtual DU $n' \in N_v^1$, considering its requirements in terms of the antenna configuration, the number of PRBs and the desired location. In the second step, all the candidate DUs $m \in candidates(n')$ of each virtual DU $n' \in N_v^1$ are considered and for each a neighbour list $neighbor(m)$ is created (line 18 through 24 in the pseudocode). This list contains all the substrate DUs whose downlink signal might interfere with the signal of $m \in candidates(n')$ candidate DU. The relative distance between the potential interfering DUs is considered for populating the neighbour list. The heuristic then measures the interference coming from each DU in the neighbor list. At the end of this step, the best candidate DU $best\_cand$ is picked; that is, the one that would introduce the minimum level network–wide inter–cell interference.

In the last step, virtual DU $n' \in N_v^1$ is mapped to the best substrate DU $best\_cand$. The PRBs of the host substrate DU are sorted in the ascending order of likelihood in terms of interference, and then the requested number of PRBs $\omega_{prb}^{v}(n')$ are allocated starting from the one that would introduce the least level of inter–cell interference in its collision domain. After the PRBs have been allocated, the overall inter–cell interference level at the host DU is estimated, and the appropriate functional spit is selected. Lastly, the fronthaul bandwidth, required to host the virtual DU, is computed by using $compute\_band()$ function, providing inputs the requested number of PRBs, the number of RF front–ends and the split option of the host substrate small cell. This is followed by updating the substrate resources and repeating the steps for all the virtual DUs in the virtual network request. Denoting $m = |N_s^1|$ and $n = |N_v^1|$ as the number of, respectively, substrate and virtual DUs considered in this study, the overall time complexity of HEU–ST is $O(nm)$ by considering the number of splits as constant and the number of candidates and their neighbours invariable and independent from the substrate and virtual network size.

10.5 Evaluation

The goal of this section is to compare the ILP–based algorithms with the heuristics. We shall first describe the simulation environment and the performance metrics used in our study. We will then report on the outcomes of the numerical simulations carried out in a discrete event simulator implemented in Matlab.

The reference substrate network is a star–shaped topology with 8 DUs directly connected to a single CU pool via optical fronthaul links (10 Gbps), providing mobile coverage in an area of 2 Km^2. This is a very conservative assumption, in more realistic scenario a ring or a tree topology may be used to connect the DUs with the CU pool. However, the focus of this study is on the flexible functional split, rather than the fronthaul topology. The inter–DU distance is 800 meters, and it is assumed that each DU possesses 6 omni–directional[6] antennas, providing radio coverage with the radius of 500 meters. This means that in some zones there will be 200 meters of area covered by more than one DU, which in turn means that, if the users are located in that area and are connected to different DUs, being scheduled at the same PRBs, they will then create interference on one another, irrespective whether or not those users belong to the same service provider (e.g., MVNOs).

One of the most prominent advantage of the PHY–RF split is that, through better inter–cell coordination, it enables complex interference cancellation/avoidance algorithms such as eICIC to be employed. In some cases where inter–cell interference is missing, the PHY–RF split in the C–RAN architecture would be unnecessary and would result in a waste of resources (e.g., fronthaul bandwidth). For example, when DUs are well–separated, meaning that they have no overlapping coverage area, or if DUs do have an overlapping area, but the users are scheduled at different PRBs. Depending upon the level of inter–cell interference, different functional splits would be appropriate to be exploited. In our model, there are three categories of interference and three corresponding functional splits (see Table 25). Notice that as opposed to the PHY–RF split, in the case of the PHY and the MAC split, some part of the baseband signal processing is taking place at DUs. For example, in the case of the PHY split, it is assumed that the half of the processing capacity is allocated to the DUs and the other half to the CU pools. This is because the most processor–hungry procedure (i.e., FFT/IFFT) is taking place in the PHY layer. The processing requirement increases at the DUs and decreases at the CUs when a fewer layers (e.g., PHY layer, MAC layer) are centralized at the CU pools.

In this study, we assume that a fixed number of virtual requests are embedded sequentially. The reported results are the average of 10 simulations each with 10 embeddings. During each embedding the number of virtual DUs, RF front–ends and PRBs are randomly selected in the set of, respectively, $\{1, 2\}$, $\{1, 2\}$ and $\{30, 60\}$.

Simulation Results: Figure 72 shows the acceptance ratio, the RF front–end utilization, the PRB utilization and the execution time of all the exact algorithms and the heuristics. As expected, both ILP–DM and HEU–DM dynamic embedding algorithms achieve better performance in terms of acceptance ratio, RF front–end and PRB utilization compared to their static counterparts (see Fig. 72a, 72b and 72c). We can see that both dynamic placement algorithms have accepted equal number of virtual network requests. It can be observed that also the RF front–end utilization (see Fig. 72b) and the PRB utilization (see Fig. 72c) is equal for those algorithms. These equalities prove that the HEU–DM heuristic achieves near optimal solutions.

With regard to the static embedding algorithms, Fig. 72a shows that the HEU–ST heuristic has accepted slightly more irtual network requests than the ILP–ST algorithm. This negligible difference can be seen also in terms of the RF front–end utilization. Whereas, it can be observed that their PRB utilization is equal. This again witnesses about the HEU–ST heuristic achieving a good approximation compared to its ILP–ST counterpart. However, as we will see in Fig. 72, this does mean that the HEU–ST heuristic finds higher–quality solutions in comparison with the ILP–ST one. The rationale behind the HEU–ST heuristic performing slightly better in terms of the acceptance ratio and RF front–end utilization is that, since for those algorithms a static embedding is taking place, it cannot be claimed that the performance

[6] Notice that also sectoral antennas at the small cells can easily be considered. However, in this work we consider only omni–directional antennas since sectoral antennas, although more prevalent, would complicate the model without bringing any significant benefit.

of the ILP–ST algorithm after all embeddings must always be better compared to the HEU–ST heuristic because they have different logic to embed virtual network request. Moreover, by performing a static embedding, with the arrival of a new virtual network request, the previously embedded request cannot be re–embedded. Thus, the performance of the static algorithms merely depends on their previous mappings and the requirement of the new requests in terms of desired location, number of PRBs and RF front–ends.

The ILP–DM and ILP–ST exact algorithms become computationally intractable when networks with a few tens of substrate small cells and a few tens of virtual nodes in a virtual network request are considered. Depending on the size of the substrate and virtual networks (e.g., a few hundreds of substrate and virtual nodes), also the HEU–DM heuristic might have a scalability problem in terms of required time to map virtual network requests.

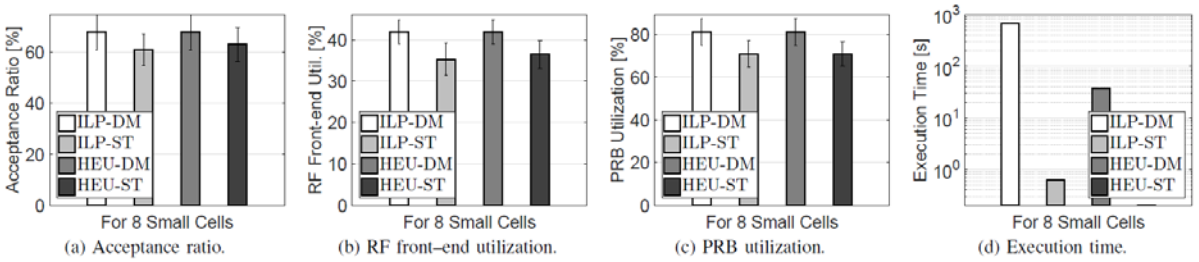


Figure 72 - Acceptance ratio, RF front-end and PRB utilization, and the execution time for the ILP–based algorithms and of the heuristics.

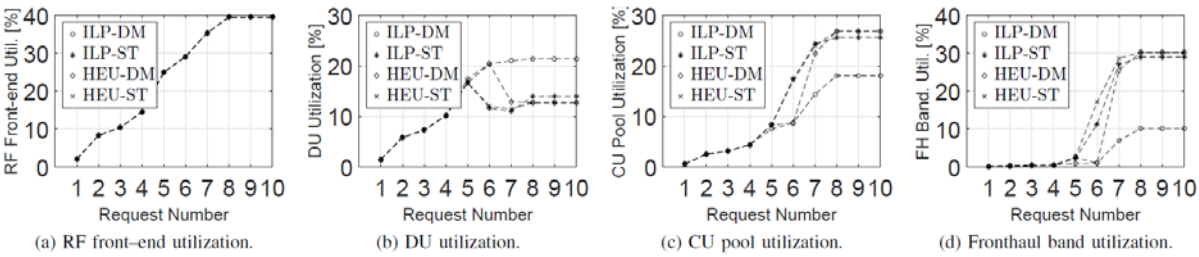


Figure 73 - RF front-end, DU processing resource, CU pool processing resource and FH bandwidth utilization for the ILP–based algorithms and of the heuristics.

With the sole purpose of tackling this problem, a static heuristic (HEU–ST) has also been proposed. Figure 72d displays the total time taken to embed 10 virtual network requests (a single iteration). Note that during that iteration all the algorithms/heuristics have embedded equal number of request (see Fig. 73). It can be observed that among the dynamic embedding algorithm/heuristic, HEU–DM has taken 20 times less time to embed those requests compared to its exact counterpart. Whereas among the static algorithm/heuristic, HEU–ST has embedded the virtual network request twice faster than the ILP–ST algorithm. The embedding time for the ILP-based algorithms will increase exponentially with larger substrate and virtual networks.

In order to get a better insight into how the resources of the substrate network are exploited, we will now examine the same single iteration (i.e., 10 embeddings). Figure 73 depicts the RF front–end utilization, the processing resource utilization of the DUs and the CU pool and the overall fronthaul bandwidth utilization for ILP–DM, ILP–SM, HEU–DM and HEU–ST. In Fig. 73a, it can be seen that all algorithms/heuristics have embedded equal number of virtual network requests, and therefore, have equally utilized the RF front–end resource. In particular, all algorithms/heuristic have successfully embedded the virtual network requests up to the 8^{th} embedding, while, rejected the last two virtual network requests.

With regard to the DU processing resource utilization (see Fig. 73b), we can observe that the ILP–DM algorithm keeps increasing the processing resource utilization gradually, and therefore, exhibits the best performance among all the algorithms/heuristics. In essence, with the arrival of virtual network request, this increase means that, due to the optimal embedding of all the virtual network requests, only MAC

and PHY layer splits are being employed by the substrate small cells/DUs, since among the considered splits, only those splits require processing resource at the DUs. It can also be observed that the second best performance achieves the HEU–DM heuristic, since as opposed to the static algorithms/heuristics, which reduce the DU processing resource utilization from the 6^{th} embedding, it reduces the DU processing resource utilization from the 7^{th} embedding. This essentially means that, in the case of the HEU–DM heuristic, up to the 6^{th} embedding most of the substrate nodes employ either MAC or PHY layer splits. While from the 7^{th} embedding some of the substrate nodes start changing the splits from PHY/MAC to the PHY–RF split in order to be able to exploit advanced algorithms aiming at reducing/cancelling the inter–cell interference.

Regarding to the static algorithms/heuristics, it can be seen that their performance resembles each other in terms of the DU processing resource utilization. However, we can see that the ILP–ST algorithm ultimately achieves slightly higher processing resource utilization at the DUs. This proves that the ILP–ST algorithm has found higher–quality embedding solutions, and therefore, has lead more substrate nodes employing higher–layer functional splits. In essence, this means that the inter–cell interference level at those nodes is lower from the inter–cell interference threshold starting from which the PHY–RF split should be employed.

The picture is totally different for all the algorithms/heuristics for the processing resource utilization at the CU pool (see Fig. 73c). The first observation is that for all the algorithms/heuristics the processing resource utilization increases at the CU pool regardless of the split options employed by the substrate DUs. The rationale behind this is that depending on the inter–cell interference level at the substrate nodes, which increases with the arrival of new virtual network requests, the functional splits change from the higher–layer splits toward lower layer splits, and with this change, the processing resource utilization increases at the CU pool and decreases at the DUs. Since in the case of the ILP–DM algorithm, only MAC/PHY splits are employed by the substrate nodes, the processing resource utilization at the CU pool is the lowest compared to the rest of the algorithm/heuristics. This is because, in the cases of those splits, the processing resource requirement at the CU pool is lower compared to the PHY–RF split in which all baseband signal processing is taking place only at the CU pool. In the cases of ILP–ST, HEU–DM and HEU–ST algorithms, it can be observed that they achieve higher processing resource utilization at the CU pool as opposed to ones at the DUs. Moreover, the ILP–ST and HEU–ST algorithms increase the processing resource utilization at the CU pool before the HEU–DM algorithm. Thus, the more is the number of substrate DUs that employ a lower–layer, the PHY split or the PHY–RF split, the more is the processing resource utilization at the CU pool and the less is the processing resource utilization at the DUs.

The fronthaul bandwidth utilization for all the algorithms/heuristics (see Fig. 73d) somewhat resembles the processing resource utilization at the CU pool. This is because apart from the processing resource utilization at the CU pool, also the fronthaul bandwidth utilization increases with the increase in the number of substrate nodes that employ the PHY split or the PHY–RF split. We can observe that the fronthaul bandwidth utilization for ILP–DM and HEU–DM is very low until the 6^{th} embedding with a small spike at the 5^{th} embedding for HEU–DM. The rationale behind this is that at the 5^{th} embedding one of the substrate nodes starts using the PHY split; while at the 6^{th}, the number of the host substrate nodes increases all of them using the MAC split (this is can be seen in Fig. 73i). We can also observe that for the static algorithm/heuristic the fronthaul bandwidth utilization is very low up to the 4^{th} embedding since at the point all of the host substrate nodes are using the MAC split, which has very low fronthaul bandwidth requirement. While from the 5^{th} embedding, the fronthaul bandwidth utilization start increasing exponentially. This huge difference in the fronthaul bandwidth utilization is due to the significant difference in the fronthaul bandwidth requirements of the considered splits (see Table 1).

We will now examine the PRB utilization, the inter–cell interference level and the functional split at all DUs for all the algorithms/heuristics for a single iteration in order to better understand their relationship (see Fig. 74). It can be observed that the PRB utilization of individual substrate small cells varies due to different mapping decisions of ILP–DM, ILP–ST, HEU–DM and HEU–ST (see Fig. 74a, 74d, 74g

and 74j). However, the network–wide PRB utilization (the sum of the PRB utilization of the small cells) is the same for all the algorithms/heuristics during each embedding. This is justified by the fact that, although the individual virtual node mapping differs for all the algorithms/heuristics, all of them have identically accepted or rejected the virtual network request as shown in Fig. 73a.

Figures 74b, 74e, 74h and 74k display the inter–cell interference level in the cases of employing, respectively, ILP–DM, ILP–ST, HEU–DM and HEU–ST. It can be observed that both dynamic embedding algorithm/heuristic successfully embed up to the 4^{th} virtual network requests without creating inter–cell interference in the network. Whereas, from the 5^{th} virtual network requests the ILP–DM algorithm creates a little amount of inter–cell interference at two substrate small cells, while the HEU–DM heuristic creates inter–cell interference at three substrate small cells. Moreover, it can be seen that at two of them the level of inter–cell interference is much higher compared to the case of employing the ILP–DM algorithm. After all the embeddings, we can see that the total inter–cell interference in the network in the case of employing the ILP–DM algorithm is much lower than the one in the case of employing the HEU–DM heuristic (look at the difference in the scales). This again proves that the mapping efficiency of the ILP–DM algorithm is higher compared to the one of the HEU–DM heuristic.

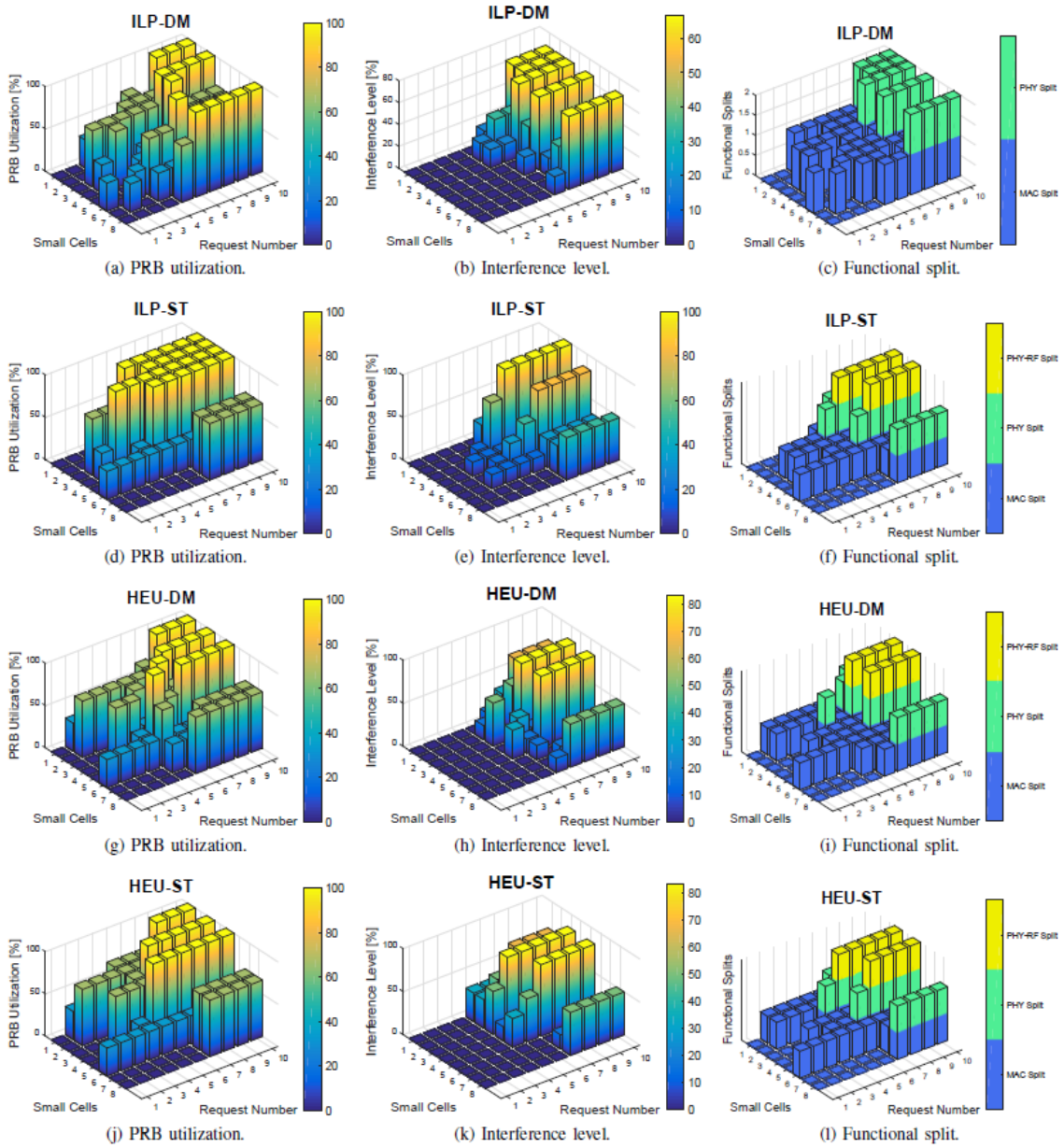


Figure 74 - PRB utilization, interference level and functional splits of the ILP–based algorithms and of the heuristics.

The picture is different in the cases employing static algorithms/heuristics. Initially, the HEU–ST heuristic is as efficient as both of the dynamic embedding algorithms, since like them, HEU–ST successfully embeds up to the 4^{th} virtual network requests without creating inter–cell interference in the network. Nevertheless, after it starts introducing inter–cell interference at some of the substrate small cells. Whereas, the ILP–ST algorithm starts creating inter–cell interference from the 3^{th} embedding, and ultimately, results in a higher network–wide inter–cell interference compared to the HEU–ST heuristic. As it has been already mentioned, in this case, this is a consequence of the HEU–ST heuristic being slightly more efficient than the ILP–ST algorithm.

Finally, let us analyze how the functional split change at the substrate small cells as a function of changing inter–cell interference. Figures 74c, 74f, 74i and 74l show the functional splits per substrate small cell for a single iteration (10 embeddings) for, respectively, ILP–DM, ILP–ST, HEU–DM and HEU–ST. In general, the lower is the inter–cell interference level, the lower–layer split is selected, leading to a more efficient fronthaul bandwidth utilization. Among all the algorithms/heuristics, the superiority of the ILP–DM algorithm is obvious since, thanks to fact that it is always able to find the optimal mapping, the substrate small cells only employ either the MAC split or the PHY split.

Thus, being able to keep the level of inter–cell interference low from the threshold starting from which the PHY–RF split must be employed, no substrate small cell employs the PHY–RF split. It is also obvious that the HEU–DM heuristic exhibits the second best performance. Whereas, the performance of ILP–ST resembles the performance of HEU–ST. The reason for this is twofold. First, if a substrate small cell has been used for mapping then by default, in the considered scenario, it employs the MAC split even if there is no inter–cell interference. We can see that up so the 4^{th} embedding there is no inter–cell interference in the network in the case of employing the HEU–ST heuristic (see Fig. 74h). However, as it can be seen in Fig. 74i, the substrate nodes that have been used to map the virtual network requests up to the 4^{th} embedding employ the MAC split. Second, although the level of inter–cell interference at some substrate nodes are higher in the case of ILP–ST algorithm compared to the one of HEU–ST heuristic, they use the same functional split option. This is because for each of the splits, there is a defined inter–cell interference range[7], and if the level of inter–cell interference at those node is within a same range then regardless of the difference in the level of their inter–cell interference, the same corresponding functional split must be employed by those substrate nodes.

Discussion: The proposed algorithms/heuristics provide MNOs with a various options to select between promptness mapping optimality and scalability. We have seen that the ILP–DM algorithm, thanks to its dynamic embedding strategy, achieves the optimal mapping for all the virtual network requests. However, it comes at the expense of significant embedding time (see Fig. 72d), which makes this algorithm not applicable to dense mobile networks with a few tens of small cells in the substrate/virtual networks. The HEU–DM heuristic, although less efficient, approximates the optimal mapping solutions found by the ILP–DM algorithm. Moreover, it is significantly faster in embedding virtual network requests, which makes it more scalable compared to the ILP–DM one. It is, however, not application to ultra–dense mobile networks with a few hundreds of small cells in the substrate/virtual networks. If it is decided to employ a dynamic embedding algorithm/heuristic, one must also take into account the possible downsides (e.g., service interruption of the users of the MVNOs) of the re–embedding of the virtual network requests.

In order to address the scalability problems of the exact dynamic placement algorithm and the dynamic placement heuristic, an exact static embedding algorithm (ILP–ST) and a static embedding heuristic (HEU–ST) are also proposed. We have seen that the static embedding algorithm/heuristic are less efficient in embedding virtual network requests and are, therefore, less efficient in employing the network resources compared to their dynamic counterparts. We have also seen that the ILP–ST and HEU–ST are not comparable since their embedding decisions depend on their previous embedding and

[7] The inter–cell interference value $I(m)$ for each functional split defined in Table 4 puts un upper bound to the acceptable inter–cell interference range since those values are used in the inequality constraint (16) in the ILP formulation.

they both use different strategies to embed the virtual network requests. Thus, one must consider the availability of the network resources and the time requirement in order to select the embedding algorithm/heuristic that would lead to the most optimal solution.

10.6 Conclusions

10.6.1 Technical/Scientific Impact

Small cells are rapidly emerging as a cost–efficient solution to provide additional capacity in current and future mobile networks, while the C–RAN architecture has come to the fore as a promising way to use the precious baseband resources elastically and efficiently based on the actual need and, through better inter–cell coordination, overcome all the possible performance degradation that may be entailed by network densification. By leveraging the C–RAN architecture over densely deployed small cells, MNOs are expected to meet the ever–increasing coverage and capacity demands, and at the same time, make sure that their precious frequency resources are used efficiently. There are, however, many challenges associated with network densification and the C–RAN architecture yet to be addressed.

In this work, we studied the problem of flexible changing the functional split option at small cell based on some KPIs. The flexible functional split in the RAN provides the possibility of exploiting complex CoMP algorithms designed to reduce/cancel the inter–cell interference. However, depending upon the level of inter–cell interference, a particular functional split would be more efficient to be used. We have seen that the processing requirements of the DUs and the CU pool, and fronthaul bandwidth requirement change substantially, depending upon the selected functional split option. This means that significant benefits can be reaped by employing the *right* functional split option for each small cell. Although, in our scenario the functional splits change from the higher–layer splits (e.g., the MAC split) toward the lower–layer splits (e.g., the PHY split, the PHY–RF split), the functional splits can also be changed towards the reverse direction, for example, considering daylight vs. night traffic variation and users distribution.

10.6.2 Feedback toward development

As a future work, we plan to extend the problem of the flexible functional split selection to real scenarios. In particular, we want to consider an operational LTE–A mobile network, and based on the available transmission links as well as the spatially and temporarily fluctuating traffic demand at eNBs, we want to study the possibility of flexibly changing the functional split options at eNBs. Finally, with the goal of conducting complex experiments in order to better understand the trade–offs between the functional splits, we also want to implement the PHY split and the MAC split in the 5G-EmPOWER testbed deployed at CREATE-NET premises.

10.6.3 Expected business impact

The study on the function split in RAN has been proposed in 3GPP and the CU and DU separation will be standardized in coming 3GPP release. In general, the flexible functional split will be a necessary feature to enable different implementations of 5G network and thus has substantial business impact to mobile operators. However, the gain of the proposed solution in terms of CAPEX of OPEX will need further investigation.

11. Conclusions and Impact

11.1 Technical Impact

The results in this deliverable depict the merits of COHERENT architecture in terms of exploiting the programmable RAN to implement our devised algorithms. For instance, the distributed control plane placement problem of CAP trade-off in the software-defined RANs incur significant bandwidth demand to keep them synchronized. Such observation draws the tradeoff between bandwidth and reliability. Moreover, the problem of allocation of primary carrier, Physical cell ID, and Beam ID utilize the centralized measurement at C3, and the genetic algorithms as the candidate that can be implemented in C3 leveraging the programmability at RAN. We also utilize the hierarchical control framework of COHERENT to address the network management problems related to mmWave & sub-6GHz multi-connectivity.

Moreover, several RAN slicing and user association works are explored based on the programmable RAN characteristics for multi-service execution environment. For example, the Traffic Rule abstraction is introduced, and four resource abstraction types are defined to flexibly serve the needs of different services. Then, the MAC scheduler can exploit these abstractions to be evolved into inter-slice and intra-slice processing, separately. Further, a joint user association and channel assignment as well as the traffic-aware user association are surveyed to provide a clear picture on the impact of the dynamicity of user association. These abstractions and user association algorithms provide the foundation for the multi-RAT and multi-tenant deployment in heterogeneous programmable RANs. Last but not least, the deliverable also examines some extensions to better cover the evolution of communication links, like two-hop D2D relaying. These studies comply with the COHERENT architecture and also exploit the programmability of the software-defined RAN concept to serve the needs of the intended use case.

11.2 Feedback Toward Development

All our devised algorithms can be implemented on top of the platform introduced in [COHERENT D2.4] leveraging the programmable RAN concept and COHERENT SDK. However, some modifications are necessary to provide suitable network abstraction parameters, represent the network graph, and enable flexible MAC scheduling control logics. Further, the development shall also be awareness to the slicing over-the-top and provide sufficient service level agreement (SLA) enforcement among the others.
Moreover, the end-to-end (E2E) consideration shall be further considered in the future (e.g., core network, transport network that beyond the radio access network) as the E2E network virtualized network can be viewed on the top of the physical one in future service-orientated vision as different slices. In that sense, surveys of heterogeneous backhaul considerations (e.g., wired/wireless) can be further conducted to better depict the network performances.

11.3 Expected Business Impact

The system architecture for resilient network nodes proposed in Chapter 2 takes into account all practical aspects of deployment, including reliability concerns and bandwidth overheads. The network operators can define the reliability target and then calculate bandwidth overhead to archive the required reliability level. This decision will affect operator business because reliability trades with additional resources and a proper balance between reliability and bandwidth overhead can be found with help of the proposed algorithms for placement and routing of distributed control plane functions.

BeamID optimization, as discussed in Chapter 3, is an important component of future mmWave 5GNR networks. It improves handover reliability and thus is essential for the overall LTE radio network performance. Robust functioning of handover increases operators' revenue and enhances user experience.

End-to-end Wi-Fi RAN slicing in Chapter 4 contributed an important element for efficient traffic offloading solution from cellular networks. The proposed framework will allow operators to partition specific services and applications in their own slice of resources and to guarantee isolation between

concurrent slices. The offering Wi-Fi slicing adds flexibility to operators to enable the deployment of more services and is a cost-effective way for fulfilling customer requirements on high data rates.

Chapter 5 solves a traffic–aware user association problem aiming at optimizing resource utilization in a heterogeneous cellular network. Mobile operators can benefit from this feature by offloading best effort traffic to the Wi-Fi network while keeping demand-attentive traffic on the mobile network.

Wi–Balance, a joint channel selection and user association scheme for Wi–Fi–based WLANs presented in Chapter 5, provides an effective collision domain isolation and channel assignment strategy. This scheme demonstrated an improvement of up to 25% and 30% in terms of network throughput and channel utilization compared with a standard RSSI–based user association mechanism and thus can result in a significant optimization of costs for Wi-Fi operator.

The offloading traffic technique introduced in chapter 6, for transferring traffic from mobile to Wi-Fi networks can be very effective and promising for alleviating congested 5G networks in the future. The 5G networks are demanded to carry a high bit rate traffic, which is going to lead to the case where investment in mobile network infrastructure may be unaffordable, therefore, it will be advantageous for MNOs that can offload traffic through Wi-Fi networks. This can lead to CAPEX decreases for operators and increase the competition among them.

Chapter 7 has verified that dynamic resource slicing can be achieved and this can be proven the independence of different actors as the operator, the service provider, etc. This can bring new players to the field and provide value chain in the network infrastructure and improve the income of the actors/players.

Chapter 8 proposes the solution to take advantage of the daily traffic pattern and delay tolerance of different traffic to optimize the radio resource usage of multiple VNOs over same MNO's infrastructure. It show the hierarchical control framework proposed by COHERENT can help operators provide RAN sharing services more efficiently.

Chapter 9, provides the means for traffic steering techniques. These techniques can increase the user experience consistency by implementing these techniques, which is a fundamental KPI in the 5G network Operators. In such a case the end users can enjoy higher QoE, make them happy with the provision of the services by the operator, who at the end can save in CAPEX and OPEX.

Chapter 10 discusses the possible technical benefits of implementing C-RAN and distributed D-RAN in mobile networks. This technique can lower the CAPEX for an MNO since the BBUs are moved to the cloud through virtualization.

Bibliography

[3GPP TR 23.711] TR 23.711 Architecture enhancements for dedicated core networks; Stage 2 (Release 14), 3GPP, Sep. 2016.

[3GPP TR 23.799] TR 23.799 Study on Architecture for Next Generation System (Release 14), 3GPP, Dec. 2016.

[3GPP TR 25.996] 3GPP TR 25.996, Technical Specification Group Radio Access Network; Spatial channel model for Multiple Input Multiple Output (MIMO) simulations

[3GPP TR 36.912] 3GPP, “Feasibility study for further advancements of E-UTRA (LTE-Advanced),” Tech. Rep. TR 36.912, 2009.

[3GPP TR 38.801] TR 38.801 Study on new radio access technology: Radio access architecture and interfaces (Release 14), 3GPP, Mar. 2017.

[3GPP TR R3-080376] Nokia Siemens Networks and Nokia, “SON use case: cell Phy ID automated configuration,” Tech. Rep. R3-080376, 3GPP, 2008.

[3GPP TS 36.814] 3GPP, “Further advancements for E-UTRA physical layer aspects (Release 9),” Tech. Rep. TS 36.814; v9.0.0, 2010.

[3GPP TS 36.872] 3GPP, “Small cell enhancements for E-UTRA and E-UTRAN - physical layer aspects (Release 12),” Tech. Rep. TS 36.872 v12.1.0, 2013.

[3GPP TS 38.211] TS 38.211 NR; Physical channels and modulation (Release 15), 3GPP, Dec. 2017.

[Agarwal2013] S. Agarwal, M. Kodialam, and T. Lakshman, “Traffic engineering in software defined networks,” in In Proc. INFOCOM. IEEE, 2013, pp. 2211–2219.

[Ahmed 2010] F. Ahmed, O. Tirkkonen, M. Peltomäki, J. Koljonen, C. Yu, and M. Alava, “Distributed Graph Coloring for Self-Organization in LTE Networks,” Journal of Electrical and Computer Engineering, vol. 2010, Article ID 402831, 10 pages, 2010.

[Ahmed2016] F. Ahmed, J. Deng and O. Tirkkonen, ``Self-organizing Networks for 5G: Directional Cell Search in mmW Networks," IEEE PIMRC Workshops, pp. 1-5, Sep 2016.

[Bandh 2009] T. Bandh, G. Carle, and H. Sanneck, “Graph coloring based physical-cell-ID assignment for LTE networks,” in Proceedings of the ACM InternationalWireless Communications andMobile Computing Conference (IWCMC ’09), pp. 116–120, June 2009.

[Banerjee2015] A. Banerjee, R. Mahindra, K. Sundaresan S. Kasera and J. Van der Merwe and Sampath Rangarajan, “Scaling the LTE Control-Plane for Future Mobile Access,” in Proc. of IEEE CoNEXT, Heidelberg, Germany, 2015.

[Basta2017] A. Basta, A. Blenk, K. Hoffmann, H. J. Morper, M. Hoffmann and W. Kellerer, “Towards a Cost Optimal Design for a 5G Mobile Core Network Based on SDN and NFV,” IEEE Transactions on Network & Service Management, vol. 14, no. 4, pp. 1061–1075, 2017.

[Belt 2014] J. Van De Belt et al., “A dynamic embedding algorithm for wireless network virtualization,” in IEEE VTC Fall, 2014, pp. 1–6.

[Berde2014] Berde, Pankaj, et al. "ONOS: towards an open, distributed SDN OS." Proceedings of the third workshop on Hot topics in software defined networking. ACM, 2014.

[Bianci2000] G. Bianchi, “Performance analysis of the ieee 802.11 distributed coordination function,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, pp. 535–547, March 2000.

[Chen2017] X. Chen, h. Zhang, Z. Han, “Delay-tolerant resource scheduling in large-scale virtualized radio access networks,” IEEE ICC 2017, Paris, France, 2017

[Chieochan2010] S. Chieochan, E. Hossain and J. Diamond, “Channel assignment schemes for infrastructure-based 802.11 WLANs: A survey,” IEEE Communications Surveys Tutorials, vol. 12, no. 1, pp. 124–136, 2010

[Cisco2017] “Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021,” February 2017

[CoherentD2.2] Deliverable D2.2 of the Coherent project. http://www.ict-coherent.eu/coherent/wp-content/uploads/2016/10/COHERENT_D2_2_PU_v1.0.pdf

[CoherentD5.1] Deliverable D5.1 of the Coherent project. http://www.ict-coherent.eu/coherent/wp-content/uploads/2017/03/COHERENT_D5_1_v6.0.pdf

[Coronado2017] E. Coronado, R. Riggio, J. Villalon, and A. Garrido, “Programming Abstractions for Wireless Multicasting in Software-Defined Enterprise WLANs,” in Proc. of IEEE IM, Lisbon, Portugal, 2017

[Coronado2018a] Estefania Coronado, Roberto Riggio, Jose Villalo, and Antonio Garrido, "Wi–Balance: Channel–Aware User Association in Software–Defined Wi–Fi Networks", in Proc. of IEEE NOMS 2018, Taipei, Taiwan

[Coronado2018b] Estefana Coronado, Davit Harutyunyan, Roberto Riggio, Jose Villalon and Antonio Garrido, "Wi–Not: Exploiting Radio Diversity in Software–Defined 802.11–based WLANs", in Proc. of IEEE NOMS 2018 (Poster), Taipei, Taiwan

[Coronado2018c] Estefana Coronado, Davit Harutyunyan, Roberto Riggio, Jose Villalon and Antonio Garrido, "Lasagna: Programming Abstractions for End–to–End Slicing in Software–Defined WLANs", submitted to IEEE WoWMoM 2018.

[cpri] “Common Public Radio Interface, Interface Specification V6.0,” CPRI, Tech. Rep., August 2013.

[cran] “C-RAN, The Road Towards Green RAN,” China Mobile, Tech. Rep., 2011.

[Deng2017ICC] J. Deng, O. Tirkkonen, T. Chen, “D2D Relay Management in Multi-cell Networks”, IEEE ICC 2017 Wireless Communications Symposium.

[Deng2017mag] J. Deng, O. Tirkkonen, R. Freij-Hollanti, T. Chen and N. Nikaein, “Resource Allocation and Interference Management for Opportunistic Relaying in Integrated mmWave/sub-6 GHz 5G Networks,” in IEEE Communications Magazine, vol. 55, no. 6, pp. 94-101, 2017.

[Du2016] J. Du, L. Zhao, J. Feng, J. Xin, and Y. Wang, “Enhanced PSO based energy-efficient resource allocation and CQI based MCS selection in LTE-A heterogeneous system,” China Communications, vol. 13, no. 11, pp. 197–204, 2016.

[Elmallah1992] E. S. Elmallah, “Algorithms for k-terminal reliability problems with node failures,” Networks, vol. 22, no. 4, pp. 369–384, 1992.

[Empower2018] “5G-EmPOWER.” [Online]. Available: http://empower.create-net.org/

[Fan2011] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, “MCS selection for throughput improvement in downlink LTE systems,” in Proc. of IEEE ICCCN, 2011

[Foukas2016] X. Foukas et al., “FlexRAN: A Flexible and Programmable Platform for Software-Defined Radio Access Networks.” in ACM CoNEXT, 2016, pp. 427–441.

[Foukas2017] X. Foukas et al., “Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture,” in ACM MobiCom, 2017.

[Foukas2017] X. Foukas et al., “Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture,” in ACM MobiCom, 2017.

[Garcia 2009] L. G. U. Garcia, K. I. Pedersen, and P. E. Mogensen, “Autonomous component carrier selection: interference management in local area environments for LTE-advanced,” IEEE Communications Magazine, vol. 47, no. 9, pp. 110–116, 2009.

[George2008] George Karakostas. 2008. Faster approximation schemes for fractional multi-commodity -ow problems. Transactions on Algorithms (TALG) 4

[Glover1993] F. Glover & al., “A user’s guide to tabu search,” Annals of Operations Research, vol. 41, no. 1, pp. 3–28, Mar. 1993.

[Gudipati2013] A. Gudipati et al., “SoftRAN: Software defined radio access network,” in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013, pp. 25–30.

[Gwee 1993] B. H. Gwee, M. H. Lim and J. S. Ho, "Solving four-colouring map problem using genetic algorithm," Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, 1993, pp. 332-333.

[harutyunyan16] D. Harutyunyan and R. Riggio. "Functional Decomposition in 5G Networks." in Proc. of IFIP AIMS, Munich, Germany, 2016.

[harutyunyan17] D. Harutyunyan and R. Riggio, “Flexible Functional Split in 5G Networks,”

[harutyunyan18flex] D. Harutyunyan and R. Riggio. "Flex5G: Flexible Functional Split in 5G Networks ", in IEEE Transactions on Network and Service Management (to appear)

[harutyunyan18mig] D. Harutyunyan and R. Riggio. "How to Migrate from Operational LTE Networks to C–RAN with Minimal Investment ?", in IEEE Transactions on Network and Service Management (under review)

[harutyunyan18mmw] D. Harutyunyan and R. Riggio. "Functional Splits over a Reconfigurable Wireless Fronthaul in 5G Networks", in Wiley Journal of Network and Service Management [under review].

[Harutyunyan2018] Davit Harutyunyan, Supreeth Herle, Dimitri Maradin, George Agapiu, and Roberto Riggio, "Traffic–Aware User Association in Heterogeneous LTE/Wi-Fi Radio Access Networks", in Proc. of IEEE NOMS 2018, Taipei, Taiwan

[Hertz1987] A. Hertz and D. Werra, “Using tabu search techniques for graph coloring,” Computing, vol. 39, no. 4, pp. 345–351, 1987.

[Heusse2003] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11b,” in Proc. of IEEE INFOCOM, San Francisco, California, USA, 2003.

[IEEE2005] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Amendment 7: Medium Access Control (MAC) Quality of Service (QoS), ANSI/IEEE Std 802.11e, LAN/MAN Standards Committee of the IEEE Computer Society Std., 2005

[ITU2015] ITU-R, “IMT Vision - Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond,” Sep. 2015.

[Jin2013] X. Jin and E. Li, L. Vanbever and J. Rexford, “SoftCell: Scalable and Flexible Cellular Core Network Architecture,” in Proc. of IEEE CoNEXT, Santa Barbara, California, USA, 2013

[Kandula2007] S. Kandula, D. Katabi, S. Sinha, and A. Berger, “Dynamic load balancing without packet reordering,” SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 51–62, 2007.

[Karupongsiri2017] C. Karupongsiri, “Negative impact of CSMA/CD on WLAN IEEE 802.11,” in Proc. of IEEE TENCON, Penang, Malaysia, 2017.

[Kckeown2008] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” ACM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008

[Kohler2000] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, “The Click Modular Router,” ACM Transactions on Computer Systems, vol. 18, no. 3, pp. 263–297, 2000.

[kokku12] Kokku, Ravi and Mahindra, Rajesh and Zhang, Honghai and Rangarajan, Sampath, “NVS: a substrate for virtualizing wireless resources in cellular networks,” IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1333–1346, 2012.

[Kokku2012] R. Kokku et al., “Nvs: A substrate for virtualizing wireless resources in cellular networks,” IEEE/ACM Transactions on Networking (TON), vol. 20, no. 5, pp. 1333–1346, 2012.

[Koponen2010] Koponen, Teemu, et al. "Onix: A distributed control platform for large-scale production networks." OSDI. Vol. 10. 2010.

[Kozel 2015] D. Kozel, “Optimization of digital modulation schemes using evolutionary algorithms.,” GNU Radio Conference, 2015.

[Ksentini 2017] A. Ksentini and N. Nikaein, “Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction,” IEEE Communications Magazine, vol. 55, no. 6, pp. 102–108, 2017.

[Ksentini2017] A. Ksentini and N. Nikaein, “Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction,” IEEE Communications Magazine, vol. 55, no. 6, pp. 102–108, 2017.

[Lembo2017] S. Lembo, J. Deng, R. Freij-Hollanti, O. Tirkkonen, and T. Chen. “Hierarchical Network Abstraction for HetNet Coordination”. In IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, Oct. 2017.

[Lembo2017b] S. Lembo, “Functions and Abstractions for Radio Access Network Softwarization”, Ph.D. thesis, Aalto University, Nov 2017.

[Levin2011] Dan Levin, Andreas Wundsam, Anja Feldmann, Srini Seethamaran, Masayoshi Kobayashi, and Guru Parulkar. 2011. A first look at OpenFlow Control Plane Behavior from a Test Deployment. Technical Report. http://www.eecs.tu-berlin.de/menue/forschung/forschungsberichte /2011

[ltebook] S. Sesia, I. Toufik, and M. Baker, LTE - the UMTS long term evolution: from theory to practice. Wiley, 2011.

[Marsch2016] P. Marsch et al., “5G radio access network architecture: design guidelines and key considerations,” IEEE Communications Magazine, vol. 54, no. 11, pp. 24–32, 2016.

[Mckeown2008] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” ACM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008

[Naveen2007] Naveen Garg and Jochen Koenemann. 2007. Faster and simpler algorithms for multicommodity ‐ow and other fractional packing problems. SIAM J. Comput. 37, 2 (2007), 630–652.

[nokia] NOKIA, “Ultra Dense Network (UDN),” 2016. [Online]. Available: http://tools.ext.nokia.com/asset/200295

[OAI] “OpenAirInterface.” [Online]. Available: http://www.openairinterface.org/

[Qazi2017] Z. A. Qazi et al., “A High Performance Packet Core for Next Generation Cellular Networks,” in ACM SIGCOMM, 2017, pp. 348–361.

[Richart2016] M. Richart et al., “Resource slicing in virtual wireless networks: A survey,” IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 462–476, 2016.

[riggio16] R. Riggio, D. Harutyunyan, A. Bradai, S. Kuklinski, and T. Ahmed, “SWAN: BaseBand Units Placement over Reconfigurable Wireless FrontHauls,” in Proc. of IEEE CNSM, Montreal, Quebec, Canada, 2016.

[Riggio2015] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski and T. Rasheed, “Programming Abstractions for Software-Defined Wireless Networks,” IEEE Transactions on Network and Service Management, vol. 12, no. 2, pp. 146–162, 2015

[Riggio2016] R. Riggio, I. G. B. Yahia, S. Latr, and T. Rasheed, “Scylla: A language for virtual network functions orchestration in enterprise wlans,” in Proc. of NOMS, Instabul, Turkey, 2016

[Ros2016] F. J. Ros and P. M. Ruiz, “On reliable controller placements in software defined networks,” Computer Communications, vol. 77, pp. 41–51, 2016.

[Ros2016] Ros, Francisco J., and Pedro M. Ruiz. "On reliable controller placements in software-defined networks." Computer Communications 77 (2016): 41-51.

[Rost2016] P. Rost et al., “Mobile network architecture evolution toward 5G,” IEEE Communications Magazine, vol. 54, no. 5, pp. 84–91, 2016.

[Rost2017] P. Rost et al., “Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks,” IEEE Communications Magazine, vol. 55, no. 5, pp. 72–79, 2017.

[Rupe2003] J. W. Rupe, “Reliability of computer systems and networks fault tolerance, analysis, and design,” IIE Transactions, vol. 35, no. 6, pp. 586–587, 2003

[Ryu2018] “Ryu SDN Framework.” [Online]. Available: https://osrg.github.io/ryu/

[Samdanis2016] K. Samdanis et al., “From Network Sharing to Multi-tenancy: The 5G Network Slice Broker”, IEEE Communications Magazine, Vol. 54, Issue 7, July 2016.

[Santos2016] R. Santos and A. Kassler, “A SDN controller architecture for Small Cell Wireless Backhaul using a LTE Control Channel,” in Proc. of IEEE WoWMoM, Coimbra, Portugal, 2016

[scf] “Small cell virtualization functional splits and use cases,” Small Cell Forum, Tech. Rep., 2015.

[Shaoteng2018] Shaoteng Liu, Rebecca Steinert, Dejan Kostic, “Flexible distributed control plane deployment”, IEEE/IFIP Network Operations and Management Symposium 2018 (accepted).

[Shi2014] H. SHI, R. V. Prasad, E. Onur and I. Niemegeers, “Fairness in Wireless Networks:Issues, Measures and Challenges,” IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 5–24, 2014.

[SRS2018] “Software Radio Systems,” http://www.softwareradiosystems.com/.

[Suresh2012] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, “Towards Programmable Enterprise WLANS with Odin,” in Proc. Of ACM HotNets, New York, NY, USA, 2012.

[Taleb2015] T. Taleb et al., “EASE: EPC as a service to ease mobile core network deployment over cloud,” IEEE Network, vol. 29, no. 2, pp. 78–88, 2015.

[Tauber2013] M. Tauber and S. N. Bhatti, “Low rssi in wlans: Impact on application level performance,” in Proc. of IEEE ICNC, 2013.

[Topologyzoo2011] (2011) The internet topology zoo. [Online]. Available: http://www.topology-zoo.org/

[Turner2010] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage, “California fault lines: understanding the causes and impact of network failures,” in SIGCOMM Computer Communication Review, vol. 40, no. 4. ACM, 2010, pp. 315–326.

[Winner] IST-4-027756 WINNER II, D1.1.2 V1.2, WINNER II Channel Models, Part I Channel Models.

[WRT2018] “OpenWRT,” https://openwrt.org/.

[Xia2013] D. Xia, J. Hart, and Q. Fu, “Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs,” in Proc. of IEEE International Conference on Communications, Budapest, Hungary, 2013.

[Yang2012] M. Yang et al., “Karnaugh-map like online embedding algorithm of wireless virtualization,” in IEEE WPMC, 2012, pp. 594–598.

[Yuan2009] C. Yuan, Z. Yuan and G. Xiqi, “Collision and Corruption Awareness Enhancement to IEEE 802.11 DCF: Scheme and Performance Analysis,” in Proc. of IEEE CMC, Yunnan, China, 2009.

[zaki10] Zaki, Yasir and Zhao, Liang and Goerg, Carmelita and Timm-Giel, Andreas, “LTE wireless virtualization and spectrum management,” in Proc. of IEEE WCNC, Budapest, Hungary, 2010.

[zaki11] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, “Lte mobile network virtualization,” Mobile Networks and Applications, vol. 16, no. 4, pp. 424–432, 2011.

[Zhang2010] M. Zhang, C. Yi, B. Liu, and B. Zhang, “Greente: Power-aware traffic engineering,” in in Proc. ICNP. IEEE, 2010, pp. 21–30.

[Zhou2016] X. Zhou et al., “Network Slicing as a Service: Enabling Enterprises’ Own Software-Defined Cellular Networks”, IEEE Communications Magazine, Vol. 54, Issue. 7, July 2016.in Proc. of IEEE CNSM, Tokyo, Japan, 2017.