Université
Nice
Sophia Antipolis

LATENCY, COOPERATION, AND CLOUD IN RADIO ACCESS NETWORK

Navid Nikaein

Thesis submitted to
University of Nice
in partial fulfilment for the award of the degree of
Habilitation à Diriger des Recherches

19 March 2015

AUTHOR'S ADDRESS:
Navid Nikaein
Eurecom
Mobile Communication Department
Campus SophiaTech
450 Route des Chappes
06410 Biot Sophia Antipolis – France
Email: navid.nikaein@eurecom.fr
URL: http://www.eurecom.fr/~nikaeinn

Forward

This document is written as support to my French ”Habilitation à diriger des recherches” (HDR) degree. Unlike a Ph.D. thesis, a HR thesis does not aim at providing a detailed description of a particular research problematic, but rather describes the various facets of responsibilities that I experienced as researcher and assistant professor since I obtained my Ph.D.

My post-doctoral career started at Eurecom in January 2004, after having completed my Ph.D. entitled “Architecture and protocols for supporting routing and quality of service for mobile ad hoc networks” at École Polytechnique Fédérale de Lausanne EPFL, Switzerland in September 2003. After the postdoc, I joined the founding team of a startup company in Sophia-Antipolis, France, working towards a range of intelligent wireless backhaul routing products for private and public networks. Four years later, I deliberately chose to join Eurecom as an assistant professor because of the experimental platform OpenAirInterface, which allowed me to work on practical aspects of wireless access network. This means working on aspects that are sometimes overlooked or not found interesting or even old fashioned by other researchers, or collecting and analyzing data from experiments to prove a concept or a point. Working with this platform is a unique experience, since one has to deal with many practical problems that sometimes lead to very interesting research avenues. Furthermore the platform is ideal for teaching, giving students hands-on experience and showing them the problems of real implementations.

This document is composed of two parts. The first part describes the highlights of my research work I have been conducting after my Ph.D. The bulk of the text is based on existing and ongoing publications, but I have also added many new results, an introductory chapter summarizing all the works and putting them in context to each other, a chapter on the critical issues in cloud radio access network, and a conclusion chapter elaborating on my research methodology and giving directions for future work. The second part contains an extended CV, which includes teaching and supervising activities, project management and acquisition, as well as a selected list of my post-Ph.D. publications.

Contents

Appendices

Chapter 1

Introduction

The main volume of the work presented here considers protocol and algorithm design and validation through experimentation for radio access networks (RAN) in a cellular, mesh, and cloud settings.

Protocol and algorithm are the essential part of the wireless communication system allowing two or more nodes to communicate with each other according to a set of predefined communication messages and rules. It is generally divided into a control plane and a data-plane, whose signalling determines (a) the type of communication and the resulted network topology, and (b) the efficiency of communication on the top of what is achievable by the physical link. However new application scenarios found in namely machine-type communication, public safety networks, interactive online gaming, moving cells, and device-to-device communication, require more complex and dynamic protocols and algorithms. While in 4G systems, this represents an opportunity to limit the modifications in the physical layer and redesign the radio access protocols and algorithm to obtain the desired features, in 5G it rather calls for a new MAC/-PHY co-design that exploits new waveform, frame structure, and advanced (non-orthogonal) multiple access schemes among the others [4,5].

Experiment is also an integral part of wireless communications. An experiment is a procedure carried out with the goal of verifying, refuting, or establishing the validity of a hypothesis in a real-world.[1] One of the very first experiments in this field was carried out by Heinrich Hertz in 1887, proving the fact that electromagnetic waves can travel over free space. Another more recent example is the concept of cloud RAN that decouples the baseband unit (BBUs) from the radio units by locating a pool of BBUs at the high performance cloud infrastructure. The concept was first proposed by Y. Lin et al. IBM Research Division, China Research Laboratory in 2010 [6], and was partially proven (only centralization) by experiments some years later by Chih-Lin I et al. in 2014 [7]. Experiments can also be used to characterize the behaviour of the system and/or analyse a specific phenomenon not yet very well understood. For example, processing time measurement of RAN functions can be used to analyse the required computational power as a function bandwidth, modulation and coding scheme and assess the feasibility of RAN virtualization in the cloud infrastructure.

Because wireless communications systems have become very complex and comprise many

[1] en.wikipedia.org/wiki/Experiment

different fields, ranging from information and communication theory, signal processing to protocols and networking, performing a reliable experiment is becoming an expensive and time consuming task. There are three main approaches to perform an experiment [8]:

- **Simulation:** typically done in a controlled laboratory environment, where some or all parts of the system and protocol stack are modeled. In addition, the execution environment is not that of the real system and the interactions with external entities (e.g. real application) are limited. This makes simulation reproducible and highly scalable but not applicable as the use of (simplified) models may deviate results from the real system behaviour. Moreover, it can hide important issues when the software is implemented and deployed on a large scale in a real environment.
- **Emulation:** also performed in a controlled laboratory environment, where at least on element is modelled. The decision on which element is real or modelled depends on the use case and purpose of the experiments. However, emulation is often based on the real execution environment and opens up the interaction with external elements and their I/O streams. This makes emulation reproducible with high-level of applicability at the cost of less scalability.
- **Real testbeds:** typically performed in a quasi-controlled environment, where almost all the elements of the system are real. The execution environment is also real and open to external entities with their IO streams making this approach applicable. However, real testbed is often expensive and not scalable and the measurements produced on them are hard to predict and reproduce.

It is also possible to combine the benefits of each approach into one platform, which is exactly the intention of the Eurecom OpenAirInterface wireless technology platform. It is designed to provide a complete experiment life-cycle spanning from a unitary link-level simulation, system-level emulation, to real-time testbeds. The main differences between the methodology used with respect to existing open-source simulation/emulation/real testbed are that firstly it is built with a real-time framework in mind, secondly it is based on real software implementation of the protocol stack (L1/L2/L3), thirdly it is designed with a system approach and interoperability with the commercial equipments. The platform includes a radio transceiver card combining radio front-end and analogue/digital converters and an open-source software defined radio that runs in real-time on common x86 Linux machines.

This thesis gives some examples of how the OpenAirInterface emulation and testbed platform can be used to perform experimental research. The following section gives a summary of the thesis highlighting my key contributions and explorations to the field of experimental wireless communication protocols and networking.

1.1 Synopsis and Contributions

1.1.1 OpenAirInterface: An Open Cellular Ecosystem

Cellular systems are among one of the last industries expected to converge from a slow-moving proprietary and expensive HW/SW platforms towards an open SW platforms leveraging commodity hardware. This is required to build an open cellular ecosystem and foster innovations in the wireless world as already produced in OpenStack for cloud services and Android for mobile OS. Currently, the only open cellular ecosystem is OpenBTS, which provides an open development kit for 2G systems [9]. However, due to the lack of open cellular ecosystem for the subsequent wireless communication systems, applied research in this field is limited within the boundaries of vendor and operator R&D groups. Furthermore, several new approaches and technologies are being considered as potential elements making up such a future mobile network, namely cloudification and programmability of radio network, dynamic meshing among the base stations, and native support of machine-type communication. Research on these technologies requires realistic and flexible experimentation platforms that offer a wide range of experimentation modes from real-world experimentation to controlled and scalable evaluations while at the same time retaining backward compatibility with current generation systems.

Chapter 2 presents some results of LOLA[2] and FLEX[3] projects among the others, in which we develop the OpenAirInterface (OAI) wireless technology platform as a suitably flexible cellular ecosystem and playground for 4G and 5G research. OAI software platform provides a standard-compliant implementation of a subset of Release 10 LTE for UE, eNB, MME, HSS, S-GW and P-GW on standard Linux-based computing equipment (Intel x86 PC architectures). It can be used in conjunction with standard RF laboratory equipment (i.e. National Instruments/Ettus USRP and PXIe platforms) in addition to custom RF hardware provided by EURECOM to implement these functions to a sufficient degree to allow for real-time interoperation with commercial devices.

One of the challenges was to build a reliable yet scalable emulation platform that can be rapidly used for the performance evaluation and at the same time as a validation tool during the design-phase of a new technology. This allows an experimenter to seamlessly transit between real experimentation and repeatable, scalable emulation. This has been achieved by the usage of real protocol stack (L1/L2/L3) for the emulation platform. However, to preserve reproducibility, scalability, and applicability properties, the emulation is built based on three main principles: (1) hybrid discrete event generator to respect the frame/subframe timing and handle both periodic and user-defined events in a timely manner, (2) protocol vectorization (or virtualization) to allow different instances of the protocol stack share the same operating system, and (3) PHY abstraction to predict the modem performance as in a real physical layer.

Another main challenge we faced when targeting LTE standard compatibility was the interoperability with the commercial equipments at both ends and across all the layers. This has been achieved for 4G following a series of interoperability verification strategy to guarantee that the OAI soft eNB and EPC are able to connect with the commercial equipments at both ends (i.e. UE and EPC). An example usage of the platform to deploy a low cost LTE network on the top of commodity hardware is also provided in the chapter.

More details about this work can be found in Chapter 2, which is based on the publications [1, 8, 10].

[2] www.ict-lola.eu/

[3] www.flex-project.eu/

1.1.2 LTE Unified Scheduler Framework

Mobile devices have evolved remarkably over the last decade and their capability to simultaneously run many applications has significantly transformed the traffic characteristics of mobile networks. Quality of service (QoS) is a fundamental component associated with these applications and network should be able to support multiple QoS requests from the same user at the same time. This requires complex buffer management and simultaneous dynamic scheduling of resources to multiple users with multiple services.

Chapter 3 presents a unified scheduler framework (USF), design and developed in the context of FLEX project for LTE system.[3]. It enables more efficient allocation of resources to users running multiple internet applications in parallel while satisfying the user-driven, service driven and operator-driven requirements. Analytical studies have shown that the framework significantly enhances the performance of existing scheduling algorithms by increasing the resolution of scheduling. The framework is applied to the OpenAirInterface uplink and downlink scheduler, and the resulted schedulers were tested and validated under real conditions.

More details about this work can be found in Chapter 3, which is based on the publications [11, 12].

1.1.3 Low Latency Contention-Based Access

One of the key issues for which network operators are seeking solutions is that of determining applications and subsequently defining services which require very low latency and short channel acquisition times. This will provide scenarios for LTE-Advanced and 5G system architecture development and study. However, the majority of consumer applications will not benefit much from lower latency than that offered by LTE, or at least users would likely not be willing to accept increased subscription rates for this feature. Important exceptions to this are interactive multi-player gaming applications which, from an operator’s perspective, represent a very strategic application area with respect to revenue potential. In addition, for machine-to-machine (M2M) applications, there will likely be cases for lower latency, namely sensing and actuation, alarm and event detection, vehicular communication, but they are not yet fully exploited.

Chapter 4 presents some results obtained in the context of LOLA[2] project, in which latency is studied both analytically and experimentally to assess the achievable end-to-end latency performance in the current LTE/LTE-A systems. The results have shown that the delay requirements for certain delay sensitive applications are not met. To lower the latency, Chapter 4 introduces a new contention based access (CBA) method for uplink synchronized devices. The main feature of CBA is that the eNB does not allocate resources for a specific UE. Instead, the resources allocated by the eNB are available to all or a group of UEs and that any UE which has data to transmit randomly uses resource blocks among the available resources within the group(s) it belongs to. As the resources are not UE specific, collision happens when multiple UEs within a group select the same resources. To address the problem of collision, UE identifiers are protected and transmitted along with the data on a randomly selected resources. This will enable eNBs to decode the control information if collision occurs (in most cases based on MU-MIMO techniques), and consequently to interpret such a collision as a scheduling request, which in turn triggers a scheduling grant for the collided UEs. Simulation results validate the rationality of the CBA method and demonstrate the achievable performance with respect to the random access method. To validate and assess the latency performance of CBA in a LTE/LTE-A under real conditions, the method has been implemented in the OpenAirInterface emulation platform for both eNB and UE, and an exhaustive set of experiments have been carried out confirming significant latency gain with the proposed method.

Although the development and validation have been done based on an LTE release 10 framework, they have a strong relevance for future cellular systems, where network-controlled device-to-device communication, and massive coordinated and uncoordinated access, grant-free and stateless uplink channel access are discussed.

More details about this work can be found in Chapter 4, which is based on the publications [2, 3, 13, 14], as well as newly generated experimental results.

1.1.4 Evolving UEs for Collaborative Wireless Backhauling

The proliferation of 4G deployments has been increasing worldwide, promising to offer to carriers the capabilities to keep up with the increasing traffic growth. Moreover, the requirements for 5G technologies render new communication trends for seamless connectivity, heterogeneous networking and interoperability highly attractive. This implicates an extensive use case span from static to moving cell scenarios applicable to public and private networks. In this perspective, two limiting factors have been identified. First, re-establishment of (non-ideal) mobile X2 backhauling when it cannot be effectively utilized to inter-connect eNBs or wired interfaces are infeasible or too costly to be deployed. Second, dual or multiple UE connectivities to secondary base stations when the required performance cannot be achieved by the serving base station (interference, outage or high mobility) or when the serving base station is overloaded.

Chapter 5 presents a new paradigm for wireless link virtualization through cooperative L2/MAC information (packet) forwarding enabled by a group of UEs, which is investigated in the context of CONECT and LOLA projects. To build a virtual link, legacy UEs are leveraged as active network elements being capable of operating simultaneously over multiple base stations (eNBs). Through cooperation, they enable reliable multi-hop operation for the over-the-air re-establishment of X2 interface. The benefit for the UEs is that they can increase the aggregated data rate by exploiting data connection to multiple eNBS at the expense of extra power consumption. Through extensive experimentations with OpenAirInterface emulation platform, the performance of the proposed architecture is evaluated under realistic conditions and its rationality validated.

The proposed wireless link virtualization have been developed based on the LTE release 8 framework. However, it has a strong relevance for future cellular systems, where non-ideal backhauling and dual UE connectivity are discussed.

More details about this work can be found in Chapter 5, which is based on the project deliverables [15, 16] and the recently submitted paper [17].

1.1.5 Closer to Cloud Radio Access Network: Study of Critical Issues

Cloud radio access network is a novel architecture which can address the cost, energy, and bit rate concerns the mobile operators are facing to support explosive growth in mobile data traffic. The main idea behind C-RAN is to perform the required base band and protocol processing on a centralized computing resources or a cloud infrastructure. This replaces traditional base stations with distributed (passive) radio elements with much smaller footprints than the traditional base station and a remote pool of base band units allowing for simpler network densification. Deploying C-RAN for mobile operators has the benefit of cost reduction due to fewer number of sites, easy software upgrade, performance improvement with coordinated multi-cell signal processing, and multiplexing gain by exploiting processing load variations to fewer resources in terms of computing, networking and storage.

Chapter 6 investigates three critical issues for the cloudification of the current LTE/LTE-A radio access network, as a part of work carried out in the context of MCN[4] project. Extensive experimentations have been performed based on the OpenAirInterface unitary simulators to characterise the base band processing time under different conditions. Based on the results, an accurate model is proposed to compute the total uplink and downlink processing load as a function of bandwidth, modulation and coding scheme, and virtualization platforms. The results also reveal the feasible virtualization approach to enable radio access network cloudification.

More details about this work can be found in Chapter 6, which is mainly based on the newly generated results.

1.1.6 Conclusion

Future directions on the selected topics of interest in the perspective of 5G research are given Chapter 7.

[4] http://www.mobile-cloud-networking.eu

Chapter 2

OpenAirInterface : An Open Cellular Ecosystem

This docuement presents OpenAirInterface (OAI) wireless technology platform as a suitably flexible platform towards an open LTE ecosystem. The platform offers an open-source software-based implementation of the LTE system spanning the full protocol stack of 3GPP standard both in E-UTRAN and EPC [18–20]. It can be used to build and customized an LTE base station and core network on a PC and connect a commercial UEs to test different configurations and network setups and monitor the network and mobile device in real-time. OAI is based on a PC hosted software radio frontend architecture. With OAI, the transceiver functionality is realized via a software radio front end connected to a host computer for processing. This approach is similar to other software-defined radio (SDR) prototyping platforms in the wireless networking research community such as SORA [21]. Other similar approaches combining PCs and FPGA-based processing make use of NI LabVIEW software [22] or using the WARP [23] architecture. To our best knowledge, OpenAirInterface is the only fully x86-based SDR solution in open-source, providing both UE, eNB, and core-network functionality. A similar closed-source development commercialized by Amarisoft (LTE 100) which targets several USRP platforms provides eNB and core-network functionality on standard Linux-based PCs [24]. OAI is written in standard C for several real-time Linux variants optimized for x86 and released as free software under the terms of version 3 of the GNU General Public License (GPLv3). OAI provides a rich development environment with a rang of build-in tools such as highly realistic emulation modes, soft monitoring and debugging tools, protocol analyzer, performance profiler, and configurable logging system for all layers and channels.

Towards building an open cellular ecosystem for flexible and low-cost 4G deployment and experimentations, OAI aims at the following objectives:

- Open and integrated development environment under the control of the experimenters;
- Fully software-based network functions offering flexibility to architect, instantiate, and reconfigure the network components (at the edge, core, or cloud using the same or different addressing space);
- Playground for commercial handsets as well as application, service, and content providers;
- Rapid prototyping of 3GPP compliant and non-compliant use-cases as well as new concepts towards 5G systems ranging from M2M/IoT and software-defined networking to

cloud-RAN and massive MIMO.

2.1 Software Platform

Currently, the OAI platform includes a full software implementation of 4th generation mobile cellular systems compliant with 3GPP LTE standards in C under real-time Linux optimized for x86. At the Physical layer, it provides the following features:

- LTE release 8.6 compliant, with a subset of release 10;
- FDD and TDD configurations in 5, 10, and 20 MHz bandwidth;
- Transmission mode: 1 (SISO), and 2, 4, 5, and 6 (MIMO 2x2);
- CQI/PMI reporting;
- All DL channels are supported: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, PDSCH, PMCH;
- All UL channels are supported: PRACH, PUSCH, PUCCH, SRS, DRS;
- HARQ support (UL and DL);
- Highly optimized base band processing (including turbo decoder). With AVX2 optimization, a full software solution would fit with an average of 1x86 core per eNB instance (64QAM in downlink, 16QAM in uplink, 20MHz, SISO).

For the E-UTRAN protocol stack, it provides:

- LTE release 8.6 compliant and a subset of release 10 features;
- Implements the MAC, RLC, PDCP and RRC layers;
- protocol service for all Rel8 Channels and Rel10 eMBMS (MCH, MCCH, MTCH) [25];
- Channel-aware proportional fair scheduling [11, 12];
- Fully reconfigurable protocol stack;
- Integrity check and encryption using the AES and Sonw3G algorithms;
- Support of RRC measurement with measurement gap;
- Standard S1AP and GTP-U interfaces to the Core Network;
- IPv4 and IPv6 support.

Evolved packet core network features:

- MME, SGW, PGW and HSS implementations. OAI reuses standards compliant stacks of GTPv1u and GTPv2c application protocols from the open-source software implementation of EPC called nwEPC [26];
- NAS integrity and encryption using the AES and Snow3G algorithms;
- UE procedures handling: attach, authentication, service access, radio bearer establishment;
- Transparent access to the IP network (no external Serving Gateway nor PDN Gateway are necessary). Configurable access point name, IP range, DNS and E-RAB QoS;
- IPv4 and IPv6 support.

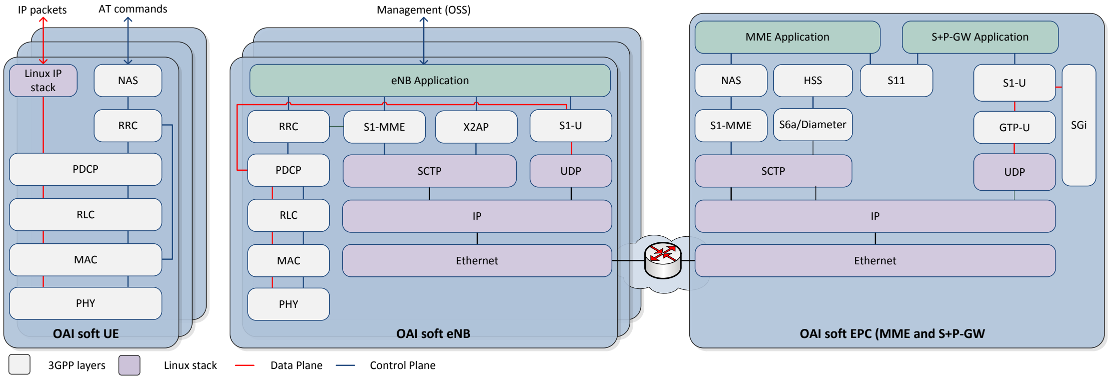


Figure 2.1: *OpenAirInterface LTE software stack.*

Figure 2.1 shows a schematic of the implemented LTE protocol stack in OAI. OAI can be used in the context of a rich software development environment including Aeroflex-Geisler LEON / GRLIB, RTOS either RTAI or RT-PREEMPT, Linux, GNU, Wireshark, control and monitoring tools, message and time analyser, low level loggin system, traffic generator, profiling tools and soft scope. It also provide tools for protocol validation, performance evaluation and pre-deployment system test. Several interoperability tests have been successfully performed with the commercial LTE-enabled mobile devices, namely Huawei E392, E398u-1, Bandrich 500 as well as with commercial 3rd party EPC prototypes. OAI platform can be used in several different configurations involving commercial components to varying degrees:

- OAI UE ↔ OAI eNB + OAI EPC
- OAI UE ↔ OAI eNB + Commercial EPC
- OAI UE ↔ Commercial eNB + OAI EPC
- OAI UE ↔ Commercial eNB + Commercial EPC
- Commercial UE ↔ Commercial eNB + OAI EPC
- Commercial UE ↔ OAI eNB + Commercial EPC
- Commercial UE ↔ OAI eNB + OAI EPC

2.2 Hardware Platform

For real-world experimentation and validation, the default software radio frontend for OAI is ExpressMIMO2 PCI Express (PCIe) board. This board features a LEON3 embedded system based on Spartan 6 LX150T FPGA as well as 4 high-quality RF chipsets from Lime Micro Systems (LMS6002), which are LTE-grade MIMO RF front-ends for small cell eNBs. It supports stand-alone operation at low-power levels (maximum 0 dBm transmit power per channel) simply by connecting an antenna to the board. External RF for high-power and TDD/FDD duplexing can be connected to ExpressMIMO2 depending on the deployment scenario. RF equipment can be configured for both TDD or FDD operation with channel bandwidths up to 20 MHz covering a very large part of the available RF spectrum (250 MHz-3.8 GHz) and a subset of LTE MIMO transmission modes. ExpressMIMO2 boards are reasonably-priced and completely open (GNU GPL), both at the hardware and software level. Figure 2.2 shows the ExpressMIMO2 hardware platform.

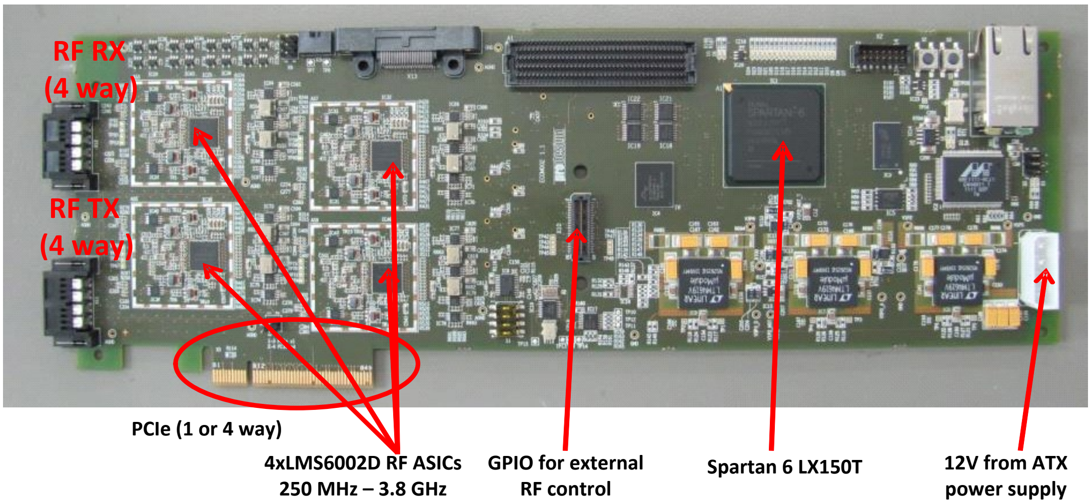


Figure 2.2: *OAI ExpressMIMO2 hardware platform.*

The embedded software for the FPGA is booted via the PC or can reside entirely in the boot ROM which is part of the FPGA design (c.f. Fig. 2.3). In the current design, the embedded software is booted by PCIexpress dynamically under control of the PC device driver. The basic design does not include any on-FPGA signal processing and consumes approximately 10-15% of the FPGA resources. There is significant room left for additional processing on the FPGA, for instance Xilinx FFT processors to offload some processing from the host PC if required.

To enhance the current design in the FPGA, every newly added block should have an AHB bus interface. The LEON3 processor is easily configured in order to interact with every block connected to the AHB bus. The PCI express controller block has been optimized in order to support the transfer of samples between the ExpressMIMO2 card and the PC at a rate that can goes up to 30.72 Msamples/s in both directions, while using only a one-lane PCI Express interface.

As shown in Fig. 2.3, there is a direct bus between the ADC/DAC and AHB PCIe interfaces to avoid the AHB protocol delay for DMA and as a results improve the transfer rate between the two blocks without passing through AHB and Leon3. All the DMA transfers are done through a shared allocated memory in the PC between the card and the user application. This shared memory has the size of one LTE frame (10 ms). There are also some shared allocated structures which allows to easily configure the card to the desired mode. The interface between the user application and the card is a command based interface. The Leon3 processor reads the commands written by the driver in a dedicated register in the PCI Express controller and executes the corresponding instructions. Although the role of the integrated Leon3 processor is very limited and consists only on configuring the RF chips and executing the DMA transfers between the card and the PC, it is easily reconfigured to add new functionality because the firmware that runs on top of it is dynamically uploaded from the PC.

Besides ExpressMIMO2, OAI now supports the UHD interface on recent USRP PC-hosted

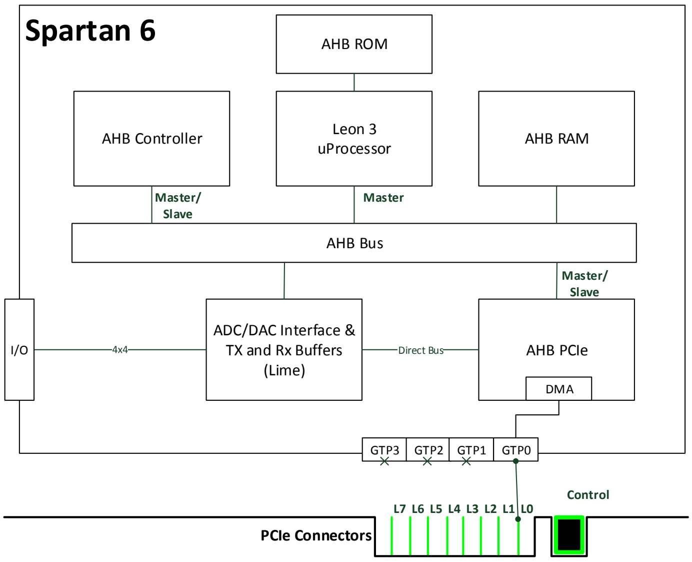


Figure 2.3: *ExpressMIMO2 FPGA design.*

software radio platforms which are widely used in the research community. Specifically, Agilent China has recently succeeded in interconnecting the OpenAirInterface softmodem software with a USRP B210 platform [27, 28]. This development is now delivered as part of the publicly-available software package from the OAI website and SVN server [18]. EURECOM will continue to maintain this development and extend to X300 (USRP-Rio) family products. This achievement illustrates the validity of the standard PC plus generic SDR frontend approach taken in OAI since the code has been independently ported successfully on a totally different hardware target.

2.3 Build-in Emulation Platform

Apart from real-time operation of the software modem on the hardware targets described above, the full protocol stack can be run in a controlled laboratory environment for realistic system validation and performance evaluation (see Fig. 2.4) [29]. The platform is designed to represent the behavior of the wireless access technology in a real network setting, while obeying the temporal frame parameters of the air-interface. The behavior of the wireless medium is obtained (a) using a PHY abstraction unit which simulates the error events in the channel decoder, and (b) using (real) channel convolution with the PHY signal in real-time. The platform can be run either with the full PHY layer or with PHY abstraction. The remainder of the protocol stack for each node instance uses the same implementation, as would be in the full system. Each node has its own IP interface that can be connected either to a real application or a traffic generator. The emulator also implements the 3GPP channel models comprising three components, path loss, shadow fading and stochastic small scale fading, and interacts with the mobility generator to perform different channel realization over time with interference. The platform targets large-scale repeatable experimentation in a controlled laboratory environment with various realistic test-cases and can be used for integration, performance evaluation and testing.

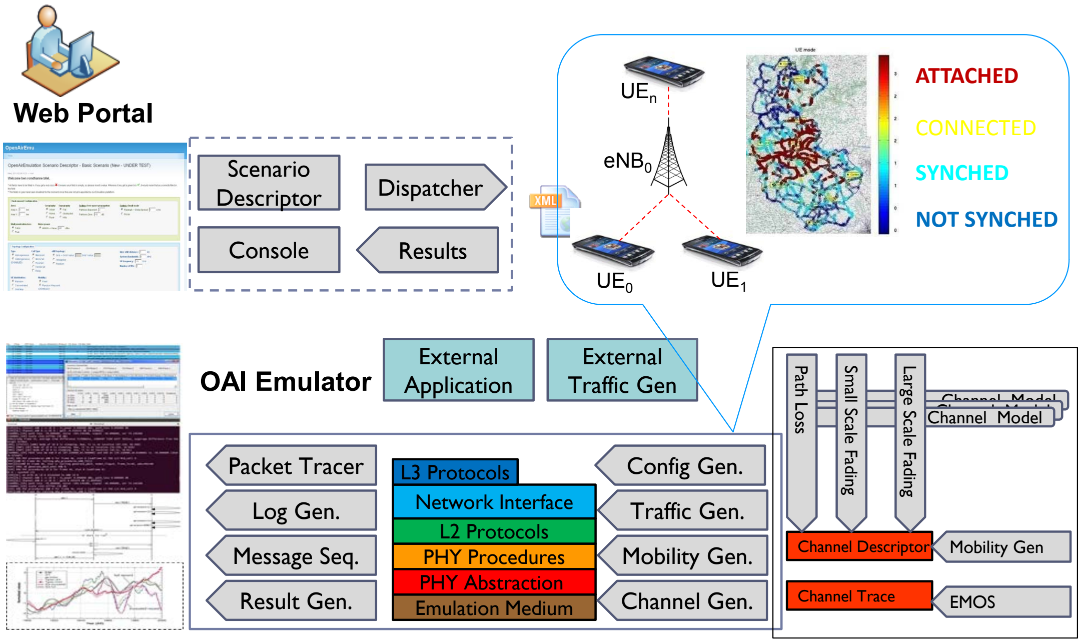


Figure 2.4: *Built-in Emulation Platform.*

The OAI emulation platform is based on a hybrid discrete event generator in that it handles periodic and user-defined events in a timely manner, preserving reproducibility, scalability, and applicability properties. It supports large number of users/connections per cell with PHY abstraction module and up to few users with full PHY under the same addressing space. Channel is described using a channel model, path loss, and node positions, and realized with the a predefined granularity (e.g. slot, subframe, frame). The supported channel models are AWGN, SCM_A, SCM_B, SCM_C, SCM_D,EPA, EVA, ETU, Rayleigh with 1 to 8 delay taps, Rice with 1 to 8 delay taps. Node positions are provided by the mobility generator module, which supports different mobility models, including STATIC, RWP, RWALK, Trace-based, steady-state RWP, and connection domain mobility model. Different traffic patterns are available for each node including voice, video, (background) data, chat/messaging, online gaming traffics as well as machine-type communication ranging from a simple sensing device (e.g. smoke, temperature sensor) to complex real-time application scenarios (e.g. embedded automatic driving system in a vehicle) [30]. Traffic pattern can be also customized in terms of packet size, packet inter-arrival time, traffic state, and session duration. Both packet size and inter-arrival time processes are modelled as i.i.d. series of random variables that can follow different distributions (e.g. uniform, Poisson, Parteo). Each node can generate multiple traffic patterns and each generated traffic type can be mapped into a logical channel allowing a specific treatment and QoS to be applied. Different key performance indicators (KPI) can be measured, namely (i) one-way delay, (ii) jitter of one-way-delay, (ii) Delay variation, (iv) loss rate, and (v) number of connected users.

The following subsections presents the OAI top-level experiment workflow, and provides further details about the three main design principles of the emulation platform.

2.3.1 Experiment Design Workflow

A sequential experiment workflow, where the output of each step will be the input of the next, is employed to allow an experiment to be reproduced. Five consecutive steps are defined: *scenario description, configuration, execution, monitoring, analysis*, where each step is split into several sub-steps as explained in the following subsections (see Figure 2.5.

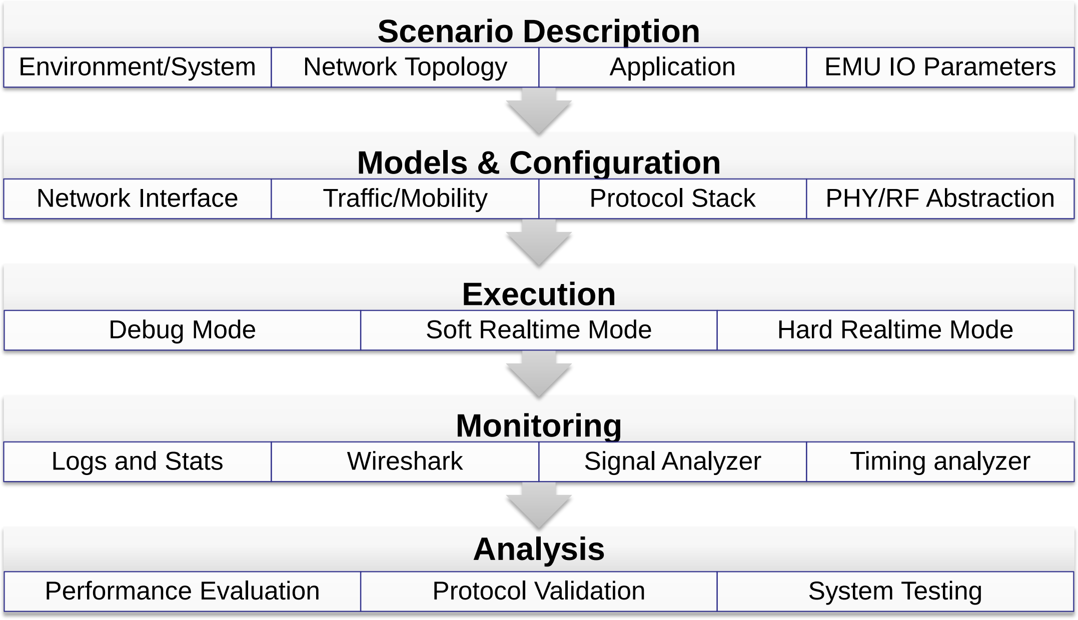


Figure 2.5: *Experiment design workflow*

Scenario Description

A scenario has four main elements described in xml format, namely (1) system/environment, where system (e.g. bandwidth, frequency, antenna) and environment (e.g. pathloss and channel models) parameters are defined; (2) network topology, where network area, network topology (i.e. cellular, mesh), nodes' type, initial distribution, and mobility model (e.g. static, random way point, random walk, trace-based, steady-state random way point, and connected domain mobilities) are set; (3) application where real application and/or emulated traffic pattern in terms of packet size and inter-departure time are defined; (4) EMU IO Parameters, where supervised parameters (e.g. emulation time, seed, performance metrics) and analysis method (e.g. protocol PDUs and operation) are set.

Configuration

This step defines a sequence of components' initialization based on the scenario description. It includes four sub-steps: (1) network interface, where the OAI IP interface is configured, (2) traffic and mobility, where traffic pattern and mobility model parameters are set, (3) protocol stack, where protocols are configured given the network topology and PHY abstraction, where a channel model predicting the modem performance is configure.

Execution

This step defines the execution environment for the emulator in order to synchronize the emulated nodes and run the experimentation. It includes four execution modes: (1) debug mode, where the emulation is executed in user space without any Linux IP connectivity, (2) soft real-time mode, where the emulation has the IP connectivity and calibrated to respect the layer 2 frame timing on average, and (3) hard real-time mode, where the emulation is executed in real-time/low-latency kernel with Linux IP protocol stack respecting layer 2 frame timing strictly.

Monitoring

This step defines how the experiment is monitored (passive and/or active). It includes: (1) logs and stats, where experiment traces and logs are collected, labelled and archived for post-processing, and online control and data place status are available to the experimenter (2) packet traces, where protocol control- and data-place signalling is captured and stored during the experiment.

Analysis

This step processes raw data and produces results and statistics. It includes three -non exclusive-analysis objectives: (1) performances evaluation, where the key performance indicators are measured and evaluated, (2) protocol validation, where the protocol control- and user-plane signalling are validated versus protocol specification, and (3) system testing, where the system as a whole is analysed and tested.

2.3.2 Discrete Event Generator

The discrete event generator (DEG) is one of the main building block of simulation/emulations allowing high-level control over a user experiments. This is important to meet different experiment requirements and use cases, in particular when targeting large scale scenarios. They allow simulation/emulation configuration and monitoring as well as scheduling user-defined events over time and space. A typical DEG consists of an event producer, an event list, a scheduler, and an event consumer. A simplified DEG implementation in OAI is shown in Listing 2.1.

```

static void my_function (Model *model) {
//
}

int main () {
  mod model;
  ev event;
  initialize(&mod,&ev);
  configure(&mod,&ev);
  schedule_event(ev_op, ev_type, time, &my_func, &mod);
  ...
  schedule_end_simu (ev);
  run();
  release();
  return 0;
}

void run() {
  while (! end_of_simulation()) {
    time=get_timestamp();
    // user-defined events
    while ((ev = event_list_get_head (&global_event_list)) != NULL) {
      if (ev.time == time )
        execute(ev);
    }
    // execute periodic events
    process_subframe(time);
  }
}
```

Listing 2.1: *Example of discrete event generator*

In OAI, the emulation events are divided into two types. Periodic events are those events occurring at each period and must be executed in the current timestamp (e.g. MAC/PHY subframe processing). These events are not queued and does not interfere with the simulation logic and user-defined events. The second type is the user-defined events, which are happening in a specific emulation time and mainly relates to the modifications of the model, reconfiguration, or changes in the state of a simulation/emulation entity. These events are stored in a timely-ordered event list.

Fig. 2.6 illustrate main components of DEG in OAI. It can be seen that there is top level emulation scheduler that controls the emulation progress and coordinates the execution of all the components. DEG interacts with the emulation scheduler through two components, event scheduler (or producer) used to add the user-defined events to the list, and event handler (or consumer) to retrieve the event corresponding to the current time and remove it from the list. Then, the emulator scheduler proceeds with the execution of the user-defined events followed by the periodic events.

2.3.3 Protocol Vectorization and Emulation Data Transport

OAI provides vectorization (or virtualization) of the entire protocol stack within the same physical machine to increase the scalability of the emulation platform (c.f. Fig. 2.7). Protocol vectorization consists of sharing the same operating system instance and Linux IP protocol stack for independent emulated node instances. It allows networks nodes to coexist in the same execution environment. Note that, protocol virtualization offers the same functional properties (i.e. services) and the same non functional properties (i.e. performances) than that of a real protocol.

To further increase the platform scalability and allow complex network experimentation, two or more emulated data flows may coexist between emulated nodes as shown in Fig. 2.7. Either nodes communicate via direct memory transfer (shared memory) or via IP multicast (over Ethernet) depending on whether they are part of the same physical machine or not. From the point

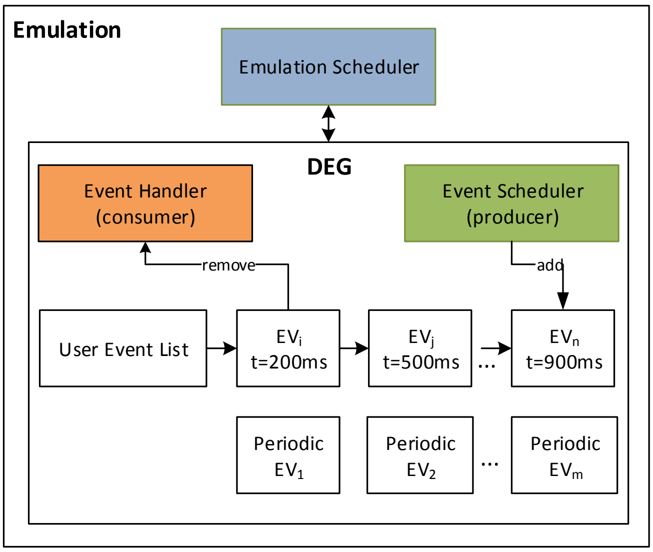


Figure 2.6: *Discrete event generator of OAI emulation.*

of view of the protocol stack, the data flow is transparent and that the network connectivity is independent from the node distribution on a physical machine.

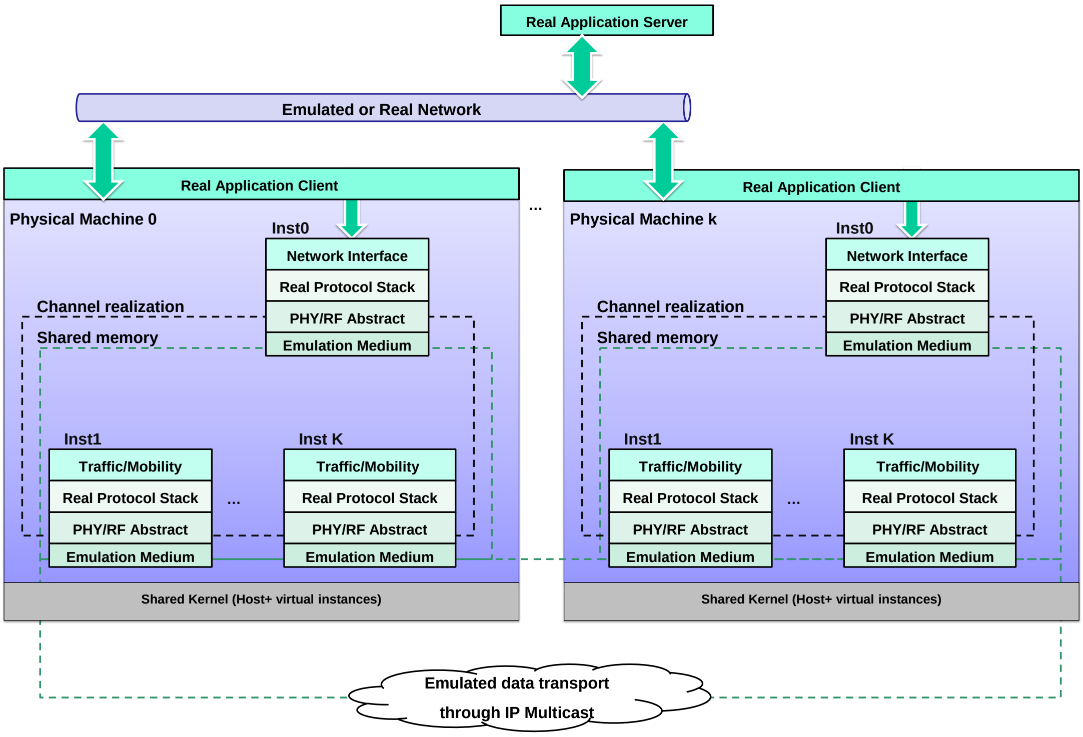


Figure 2.7: *Protocol vectorization and complex network experimentation based on the OAI emulation-platform.*

2.3.4 PHY Abstraction [1]

It was found with the help of profiling in OAI system-level emulator that for any kind of emulation more than 75% of the total simulation time and resources were spent only on the modulation, demodulation, coding, decoding and the convolution of the channel with the signal at the physical layer. This is a huge overhead in terms of complexity and time duration of system level evaluations. Therefore, to reduce the complexity and duration of system level evaluations a new interface is needed to replace the actual physical layer computations and provides the higher layers with necessary and accurate link quality metric, i.e., block error rate (BLER) or packet error rate (PER).

The use of PHY abstraction in system evaluations should provide three main benefits as follows:

1. **Low complexity and speed** by replacing the complete physical layer processing with a rather simple calculations using table look ups,
2. **Scalability** by making it possible to evaluate huge systems with hundreds of nodes,
3. **Realism** by providing the true link quality metric as it would have obtained with full PHY processing.

PHY abstraction, also referred to as link-to-system mapping and link prediction, provides such an interface between system level simulators and link level simulators for the large scale system simulations. This interface is normally a metric representing the quality of an instantaneous physical link (channel) between the eNodeB and the connected UEs by taking into account other important parameters of the system. These parameters as shown in Fig. 2.8 may include the knowledge about power and resource allocation to the specific UE, number of spatial layers, modulation and coding scheme (MCS), and mainly channel characteristics such as path loss, shadowing, fading, and interference.

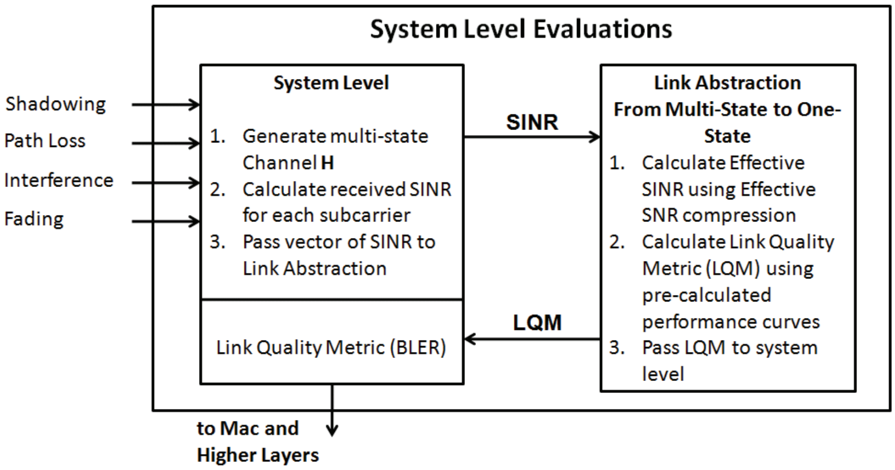

System Level Evaluations
Shadowing
Path Loss
Interference
Fading
System Level
1. Generate multi-state Channel H
2. Calculate received SINR for each subcarrier
3. Pass vector of SINR to Link Abstraction
Link Quality Metric (BLER)
SINR
LQM
Link Abstraction From Multi-State to One-State
1. Calculate Effective SINR using Effective SNR compression
2. Calculate Link Quality Metric (LQM) using pre-calculated performance curves
3. Pass LQM to system level
to Mac and Higher Layers

Figure 2.8: *Link Abstraction in System Performance Evaluation.*

PHY abstraction is rather trivial for the frequency flat channels as the simple averaging of channel qualities is sufficient for link quality mapping but for highly frequency selective channels the performance evaluation is not that trivial. This is mainly because of the smaller coherence bandwidth than that of the signal bandwidth giving rise to the multi-state channel at the receiver. However to address this issue many link abstraction techniques have been proposed in the literature for these multi-state channels [1]. OpenAirInterface apply the methodology of expected effective SINR mapping (EESM) based PHY abstraction [31, 32].

In OpenAirInterface the required parameters for large scale system simulations are highly dynamic and can be generated either by the specialized tools already included in the emulator, such as openair traffic generator (OTG) and openair mobility generator (OMG), or these parameters can be specified explicitly in great details for the specific scenarios and evaluations. The use of PHY abstraction in OpenAirInterface system-level emulator is explained in Fig. 2.9.

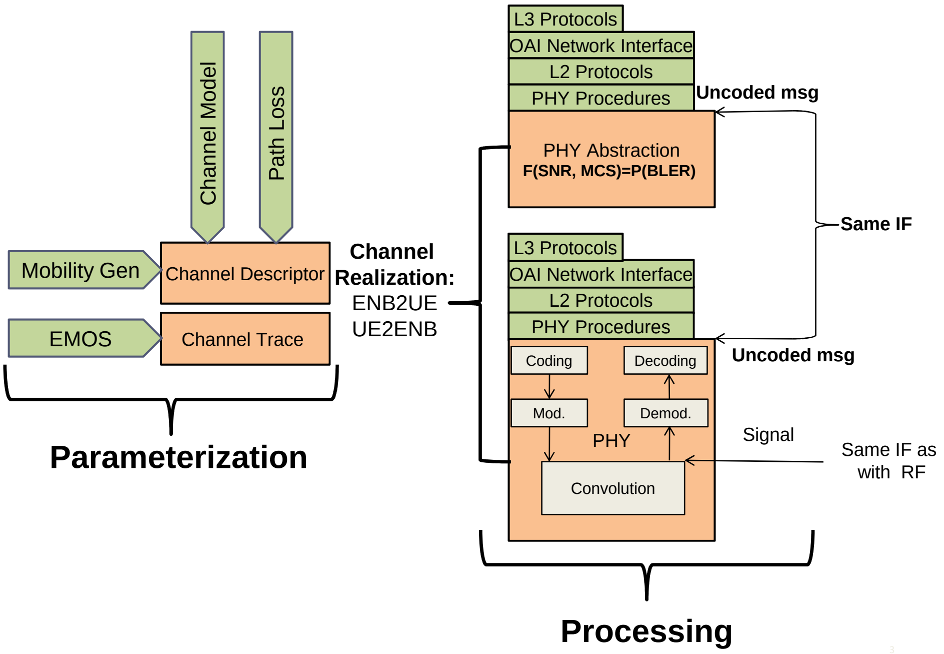


Figure 2.9: *PHY Abstraction in System Performance Evaluation in OpenAirInterface*

It can be seen from the Fig. 2.9 that there are two important steps in any evaluation using OpenAirInterface, parameterization and processing. It is important to note that parameterization is independent of the knowledge about the PHY abstraction. The output (channel realizations) from parameterization step is given to the processing where the comparison between using the full PHY and PHY abstraction is shown. It can be seen that in the case of PHY abstraction there is no coding, decoding or other complex operations involved from the transceiver chain at the physical layer (L1) only. The main purpose of the physical layer is to inform the higher layers about the status of the decodability of data packet. If the decoding is successful then the higher layers are notified about it. However in the case of using PHY abstraction this is achieved by predicting a link quality metric in terms of block error probability from the instantaneous channel realizations across all of the subcarriers. After the BLER is calculated using PHY abstraction, a random number between 0 and 1 is generated which is compared with this BLER for taking the decision on the successful or unsuccessful transmission. Then the outcome of this probabilistic experiment is passed to the higher layers which perform their tasks

Table 2.1: SIMULATION TIMES DIFFERENT TRANSMISSION MODES

		Time in minutes and seconds	
		Full PHY	PHY Abstraction
TM 1	Total time	2m26.602s	0m6.794s
	user CPU time	2m25.633s	0m6.480s
	system CPU time	0m0.924s	0m0.328s
TM 2	Total time	4m1.607s	0m9.085s
	user CPU time	3m59.079s	0m8.753s
	system CPU time	0m1.940s	0m0.364s
TM 6	Total time	2m19.320s	0m7.027s
	user CPU time	2m18.473s	0m6.752s
	system CPU time	0m0.824s	0m0.300s

independent of the knowledge about the PHY abstraction.

To illustrate the speedup factor, system level simulations for different transmission modes both with full PHY and PHY abstraction. The underlying scenario consists of a system with one eNodeB and two UEs, 500 frames, and full buffer in downlink. In the comparison, only downlink abstraction is considered.

During the simulations the accumulated averaged throughput of system over given number of frames and also the execution time for the simulation are calculated. To show that using PHY abstraction is less complex and it speeds up the evaluation process, execution time for system simulations of same scenarios with full PHY and PHY abstraction are measured. It is found that simulations with abstraction took extremely short time than that of with full PHY. The calculated speedup factor for PHY abstraction was found to be around 30 when compared to the time for full PHY. Table 2.1 shows the execution time for the simulation and it is clear from the results that PHY abstraction speeds up the process very drastically.

The next important thing to demonstrate is the realism of abstraction in system level emulation. Here realism means that the emulations with PHY abstraction should produce the results similar to the simulations with full PHY. This is shown by plotting the accumulated average throughput of the system over a given number of frames in Fig. 2.10 for transmission mode 1, 2 and 6 respectively. It is very clear that performance of both full PHY and PHY abstraction is very much close to each other and provide the same system throughput. Another important aspect to note is that although the adjustment factors are calibrated with Rayleigh channel model but to show the applicability of PHY abstraction in diverse channel models, different channel models are used for the simulations of these transmission modes. For example simulation for the TM 1 was performed with 8-tap Ricean channel, simulation for TM 2 with 8-tap Rayleigh channel and simulation for TM 6 with single tap Ricean channel. It is clear that the calibrated factors for Rayleigh channel are also applicable to other channel models thus giving rise to its applicability. Further it can be straight forwardly inferred that in the case of more UEs in the system, the gains achieved from PHY abstraction will be even significant while maintaining the realism of evaluations.

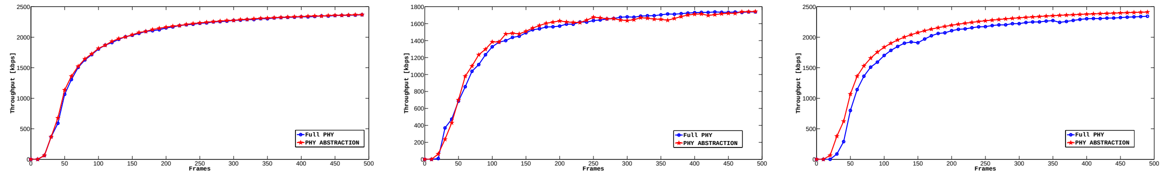


Figure 2.10: *Accumulated average system throughput over given number of frames: LTE Transmission Mode 1 (left), Transmission Mode 2 (middle), and Transmission Mode 6 (right).*

2.4 Comparison of OAI with Other Platforms and Approaches

Wireless testbeds are essential for furthering the evolution of wireless communication networks. It enables precise evaluation and validation of new algorithms, protocols and techniques for researchers. Testbeds can be built using commercial base stations and some allow open access for experiments. The CREW (Cognitive Radio Experimentation World) project [33], like Berlin LTE-advanced testbed [34] has implemented an LTE testbed that operates in the 2.1 GHz band with both indoor and outdoor configurations. Such testbeds are few in number and expensive to deploy.

Wireless technology platforms are generally classified as software oriented and hybrid platforms.

- **Software platforms** such as GNU Radio [35] are used with low cost external programmable hardware (FPGA and DSP cores) that performs most of the signal processing and includes an RF front end. Testbeds build using these platforms usually generate their signal offline using Matlab or Octave and use the nodes for transmission. Some programmable hardware that are commonly used for build such SDRs are Ettus USRP [27], Nuand bladeRF [36], and hackRF [37]. Hydra [38] is one such wireless testbed that is implemented using GNU Radio and Ettus USRP [27]. It uses a 16 bit ADC and supports upto 400 MSps and can transmit between 70 Mhz to 60 GHz;

- **Hybrid platforms** such as OAI and PicoSDR provide both the hardware and the software in a complete package to enable rapid deployment of high performance wireless testbeds. These platforms are developed from the scratch to operate with dedicated and/or commonly used hardware and software platforms.

LTEENB [39] is a base station software that simulates an LTE core network with all the processing performed using a standard PC and uses an Ettus USRP. It implements 3GPP release 8 and runs in real-time. It can achieve about 60 Mb/s in downlink and about 25 Mbps in uplink. The gr-lte [40] is a GNU Radio LTE receiver that can receive, synchronize and decode LTE signals that runs on the user equipment. OPEN-LTE is another open-source implementation of the 3GPP LTE specifications that has been successfully run on USRP and hackRF. It requires the signal to be processed offline using Octave.

Nutaq PicoSDR [41] is a MIMO waveform development platform that is capable of establishing 4X4 MIMO communication between 300 MHz and 3.8 Ghz. Each PicoSDR board contains one or two Perseus cards, each of which contains an FPGA which is connected to the RF front-end. Currently it supports 3GPP Release 9 and is configured to work with commercial LTE UEs but it does not scale well and can only support low data rates. Recently, the PicoSDR has been used at INRIA [41] to develop an wireless technology platform that has been shown to achieve upto 250 kbps Tx rate at the PHY layer.

SORA is a hybrid SDR platform that was developed by Microsoft Asia. It consists of a radio control board connected to a RF front end and is capable of supporting very high data rates with commodity PCs. It achieves high system throughput using a PCIe-based interface card (upto 16Gbps) and uses several software and hardware techniques to achieve high performance such as drastically reducing computation requirement by using look-up tables for PHY processing. SORA supports upto 8X8 MIMO communication with 802.11 a/b/g/n at full data rates. It provides a modular data flow composition system called Brick for programming various layers. It currently does not have support for standard compliant LTE.

Wireless open-Access Research Platform (WARP) is a programmable SDR platform that consists of FPGAs for DSP-optimised development connected to an RF front end. It uses programmable blocks to enable creation of highly flexible and tailored architectures. These programmable blocks offer high computation capability for moderate energy consumption. WARP can support upto 65 MSps with a 14 bit ADC and upto 4X4 MIMO configuration. It has reference designs for 802.11 a/g PHY and MAC and is designed to interoperate with commercial WiFi devices. But its RF range is currently limited to the 2.4 GHz and 5 GHz bands.

Cognitive baseband radio (COBRA) [42] is a wireless platform architecture for future mobile handsets and battery operated devices as well as base stations for small cells. The architecture can be adapted for 802.11 a/c/n, LTE and TVWS networks. This wireless platform is standard compliant and can interoperate with commercial devices. This platform is mostly suitable for battery operated devices and does not scale to macro base stations.

Small Form Factor SDR (SFF SDR) [43] radio platform is an open development platform developed by Texas Instruments that is made up of three modules, digital processing, data conversion and RF module which offers high flexibility for development. It is portable and can support a wide range of technologies from RFID to WiMAX and can communicate in the 360 MHz to 960 MHz range. As an additional feature, it provides embedded power monitoring to further improve energy efficiency and accurately estimate battery life. Currently it does not support standard compliant LTE.

Table 2.3 provides a comparison among the commonly used platforms.

2.5 Validation

To demonstrate the validity and operation of OpenAirInterface LTE eNB and EPC, and the usage of commodity hardware to run LTE network in a PC, a sample indoor experiment is presented in this section. The considered setup is depicted in Figure 2.11, and consists of a static laptop equipped with a USB LTE dongle (Bandrich C500), 1 OAI soft eNB and 1 OAI soft EPC running on the top of Intel-based PC(s). Different setups are possible ranging from an all-in-one PC to all in a physically separated entities, which are deployment-specific. For the experiment, we used two different physical machines to run eNB and EPC so as to take into account the delay of S1 interface. In such a configuration, eNB is running on the host PC under real-time/low latency Linux, MME and S+P-GW running on regular Linux, and HSS in

Table 2.2: COMPARISON BETWEEN SDRS

Feature	**SORA**	**WARP**	**GNU Radio + USRP (OpenLTE)**	**LTEENB**	**Nutaq PicoSDR**	**OAI**
Open-Source	Partial (for academic purposes only)	Yes	Yes	No	No	Yes
Work with commercial LTE UEs ?	No	No	Yes	No, Yes for the commercial version	Yes	Yes
Emulation Capability	Yes	Yes	Yes	Yes	Yes	Yes
Cost	$8000 ; Multi-core PC: $1000 ,4x4 MIMO Kit: $7000	$ 6000	$2000	$2000	$ 11000	$3200 4x4 MIMO, $ 1000 NI/Ettus B210 without host PC
Performance	Upto 300 Mbps	54 Mbps	12 Mbps	Downlink 60 Mbps ; Uplink 25 Mbps	Upto 300 Mbps	Downlink 20 Mbps ; Uplink 2 Mbps [1]
Features	Supports 802.11 a/b/g/n in full datarate with 4x4 MIMO	Supports 802.11	Supports some features of LTE	Implements LTE release 8 with FDD and a core network emulation	Supports LTE PHY only	Supports 802.11n, LTE E-UTRAN (Rel. 8 and 10) and EPC (Rel. 9 and 10) in both FDD and TDD
Project Active	Yes	Yes	Currently in development	Yes (Commercialised by Amarisoft)	Commercial	Yes
Community Size	50 academic institutions and several ongoing projects	Very popular with several active projects (headed by Mango Communictions)	Small community, avg 2 new threads per week in the forum (low traffic)	No community	No Communicty	30 academic and industrial institutions and several active projects
Power Consumption	450 W	200 W (180+20)	USRP's	USRP's	220 W	30W EXMIMO card for 4x4 MIMO, USRP's

another PC. The eNB is configured for in FDD band 7, with 5MHz BW and transmission mode 1 (SISO). The OS is a low latency Linux kernel and the hardware platform is the EXMIMO 2.

Table 2.3: Comparison between RF Frontend

Feature	HackRF	BLADERF-FX40	BLADERF-FX15	USRP-B210	USRP-X300	EXMIMO2
Radio Spectrum	30Mhz - 6GHz	300 MHz – 3.8 GHz	300 MHz – 3.8 GHz	50MHz – 6 GHz	50MHz – 6 GHz	300 MHz – 6 GHz
Bandwidth	20MHz	28MHz	28MHz	30.72 MHz	120 MHz	20MHz ?
Duplex	half	full 1x1	full 1x1	full 2x2	full 2x2	full 4x4
ADC/DAC Chip	MAXIntegrated	LIME	LIME	AD	AD	LIME
Sample Size (ADC/DAC)	8 bits	12 bits	12 bits	12 bits	14 bits	16 bits
Sample Rate (ADC/DAC)	20MS/s	40MS/s	40MS/s	61Ms/s	200 MS/S	?
Interface	USB2	USB3	USB3	USB3	SFP,ETH,PCIe x4	PCIe x 1
FPGA Logic Elements	Yes CPLD	40K	115K	150K	325K - 400K	30K ?
Opensource	all	all	all	host code	host code	all?
Cost	300$	400$	650$	1100$	4000$	4000$

Fig. 2.12 shows the two screen-shots of the connection manager of the dongle. It can be observed a successful attached procedure (left figure) and downlink data rate of 7Mbps obtained (right figure).[2]

A set of experiments is carried out to characterize the performance in terms of round trip time (RTT) considering only one logical channel. In particular, the RTT is evaluated through different traffic patterns generated by D-ITG [44], namely 32, 125, 512, 1408 packet sizes (PS), and 1, 0.5, 0.25, and 0.1 inter-departure time (IDT).

Fig. 3.6 demonstrates the measured RTT as a function of PS and IDT at 5 meters (setup 1) and 20 meters (setup 2) of distance from the eNB. As expected, higher RTT values are observed when PS, IDT, and distance increase. However, low RTT variations are observed, which is due to the fact that (a) the experiment is completely isolated, i.e. no external traffic, and no user mobility, and (b) resource allocation is performed as a function of traffic load in way to maximize the user spectral efficiency.

2.6 Conclusion

This document presents the OpenAirInterface as a suitably flexible platform for an open cellular ecosystem for 4G and future 5G research. The platform offers an open-source reference software implementation of 3GPP-compliant LTE system and a subset of LTE-A features for real-time indoor/outdoor experimentation and demonstration as well as large scale system emulation. An example usage of the platform to deploy a low cost LTE network on the top of commodity hardware is shown.

[2] Video can be found at https://plus.google.com/+OpenairinterfaceOrg

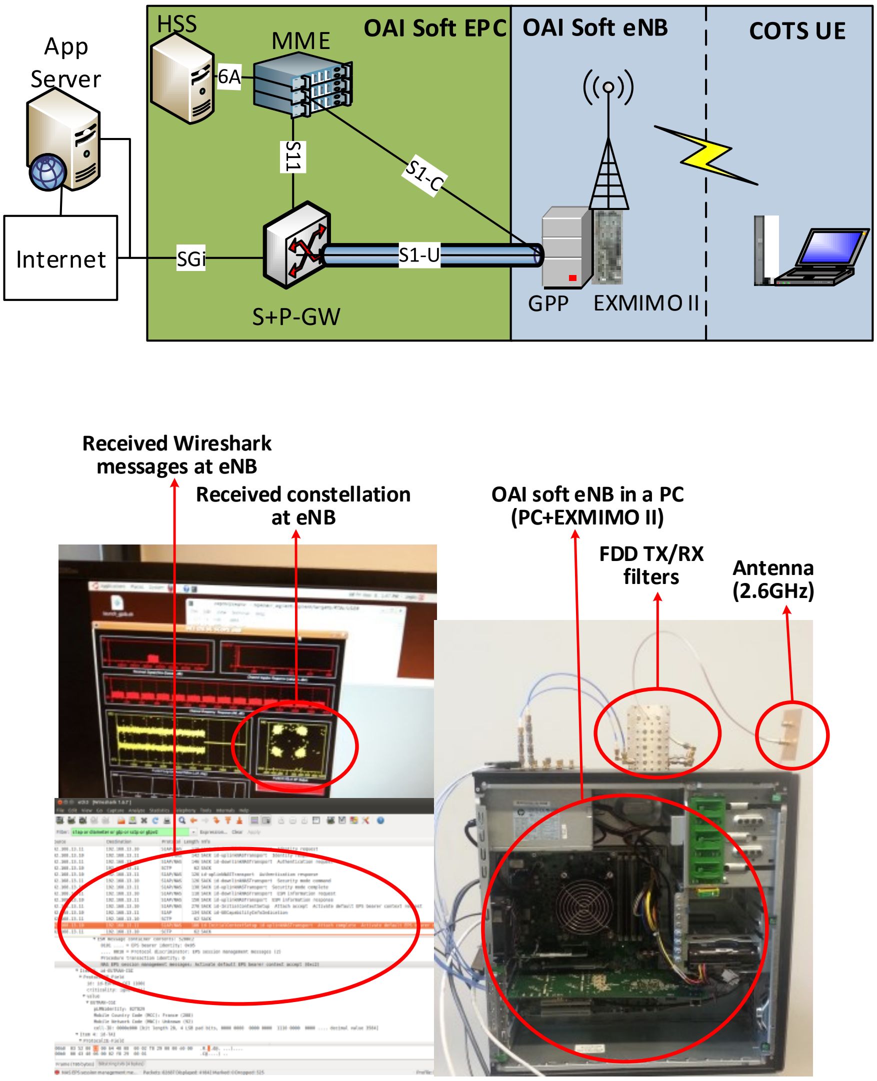
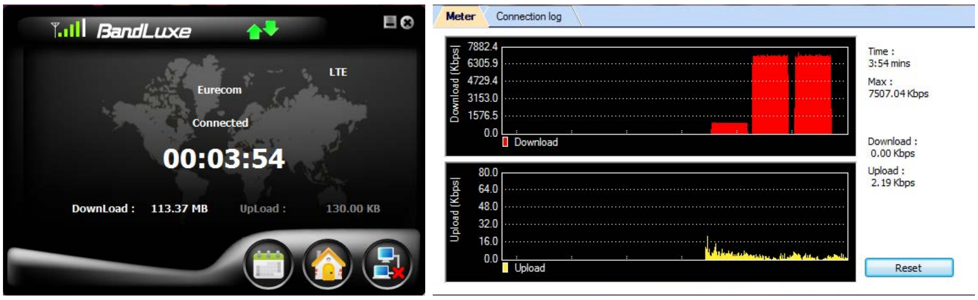


Figure 2.11: *Experiment setup and eNB hardware components and tools.*



Eurecom

Connected

00:03:54

DownLoad : 113.37 MB UpLoad : 130.00 KB

Time : 3:54 mins

Max : 7507.04 Kbps

Download : 0.00 Kbps

Upload : 2.19 Kbps

Figure 2.12: *Validation Results*

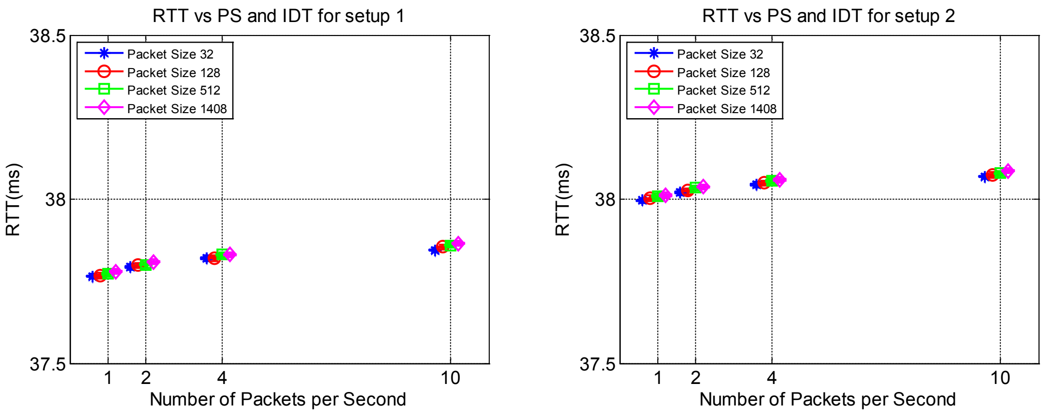


Figure 2.13: *Experiment 1*

Chapter 3

Unified Scheduler Framework for LTE/LTE-A

Modern mobile devices are capable of simultaneously running multiple internet applications. Consider for example a user having video call on the mobile device. At the same instant of time, there is a possibility of running several other applications such as video buffering, online gaming, voice recognition, and email services. Each of this services has a respective QoS requirement and therefore a dedicated radio bearer is established for each data flow and mapped to a corresponding logical channel. As a result, a single user has buffer queued in several logical channels in the same transmission time interval (c.f Fig. 3.1). Furthermore, the queued buffer will expand into two dimensions ($N \times K$) when there are multiple active users in the system requesting for several services. Here N and K refer to total number of active users and total number of logical channels for each user, respectively. Each buffer element (n, k) has a specific QoS requirement and is characterized by several parameters with specific values that are described in Section 3.1. All these parameters are crucial for buffer management and scheduling algorithm.

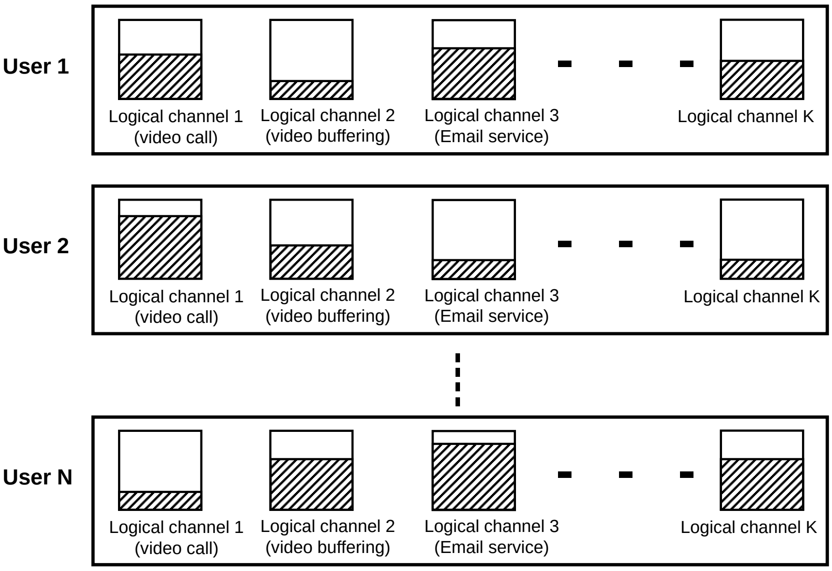

User 1
Logical channel 1 (video call)
Logical channel 2 (video buffering)
Logical channel 3 (Email service)
Logical channel K
User 2
Logical channel 1 (video call)
Logical channel 2 (video buffering)
Logical channel 3 (Email service)
Logical channel K
User N
Logical channel 1 (video call)
Logical channel 2 (video buffering)
Logical channel 3 (Email service)
Logical channel K

Figure 3.1: *Example of buffer queue for Multi-service offering*

Most of the traditional scheduling frameworks have been designed to deal with users having single service offering, meaning that they classify a user belonging to only one specific QoS class and therefore apply the scheduling algorithm at the user level [45–47]. Such framework would lead to inefficient buffer management and resource allocation, and achieve a sub-optimal system performance in multiple-service offering scenario. In the traditional MAC, the scheduling algorithm optimally allocates resources to users based on the assumption of user belonging to a single QoS. The proposed scheduler considers a multi-service users and as a result improve the performance of MAC schedulers.

In this section, a scheduling framework that deals with the shortcomings of traditional static MAC-layer scheduling is proposed to provide a more dynamic approach to satisfy the operator-driven, technology driven and service-driven requirements through reconfigurable APIs. for the analysis, the framework is applied to downlink resource allocation, but it can be easily extended to uplink.

3.1 Scheduling Parameters

The scheduler is designed for all-IP packet switched networks and applied to the 3GPP LTE eNB scheduler in downlink. In such networks, the buffer of a logical channel consists of packets and the scheduling algorithm is applied to N users having K logical channels. Every logical channel is associated with a service belonging to specific QoS class. Depending up on the service requests, there is a buffer queue in the respective logical channel of a user. In the analysis, each user requests multiple services at the same time. Therefore, there can be multiple buffer queues waiting to be scheduled. Each buffer queue is identified by a set of parameters that are utilized for designing an optimal scheduling algorithm. The parameters are categorized in four different groups depending up on their association.

3.1.1 Packet-Level Parameters

Packets are the data units that constitute the logical channels. Every packet has a set of predefined parameters as follows:

- **Average packet size in byte (APS):** Packet size is dependent up on the type of service being served such as ftp packets, browsing. For the analysis, average packet size for each kind of service is considered.

- **Packets inter-arrival time in ms (PIT):** This gives the frequency of packets arrival in the buffer queue which is also dependent up on the type of service.

- **Packet arrival time in ms (PAT):** It is the time-stamp of when the packet arrived in the buffer queue.

- **Packet maximum-allowable delay in ms (PMD):** This is a predefined value for each service type. It is also referred to as the latency constraint.

- **Packet remaining time in ms (PRT):** This is the time remaining before the packet will be dropped from the buffer queue. It is dependent up on the packet arrival time, packet maximum-allowable delay and current time.

3.1.2 Logical Channel-Level Parameters

These parameters characterizes the logical channel/service of a user.

- **Number of protocol data units (NPDU):** Every logical channel is composed of a number of packets referred to as protocol data units (PDU).
- **Head-of-line delay in ms (HOD):** It is the time in buffer for the first packet in the buffer queue of a logical channel.
- **Buffer size of a logical channel in bytes (BS):** This is the sum of the all the packet's size in a logical channel.
- **Guaranteed bit-rate in Kbps (GBR):** This is the minimum required bit-rate for each service type. There are few services that have non-guaranteed bit-rate.
- **Traffic load (TL):** This is factor for traffic modeling of a given service. It varies from 0 to 1 where 0 means there is no traffic and 1 indicates continuous flow of traffic.

3.1.3 User-level parameters

The parameters associated with user are:

- **Total buffer size in bytes (TBS):** The sum of the buffer sizes for all logical channels of a user.
- **Number of logical channels (NLC):** It is the total number of active logical channels or services of a user. Please note that NLC could also be represented by the number of logical channel groupd (NLCG).
- **Channel quality indicator (CQI):** This is the channel quality feedback received from physical layer. It is an important factor for all channel aware schedulers.
- **Subscription type (ST):** When a user subscribed to a network operator, it has the option of selecting one of the many subscriptions. Usually, the better the subscription type, better is the promised data-rate. Three subscription types are considered in this analysis: basic, silver and gold.

3.1.4 System-level parameters

These are the global parameters of a system.

- **Number of active users (NU):** It indicates the number of users that are active for transmission/reception in that particular transmission time.
- **Maximum allowed scheduled users (MAU):** Every system has a limitation on the number of user that can be actually served. This number is usually dependent up on the system bandwidth.
- **Frame configuration type (FCT):** It represents the frame structure, and can be time division duplex (TDD) or the frequency division duplex (FDD). In this work, FDD configuration type is considered.
- **Transmission time interval in ms (TTI):** This is the smallest unit of transmission time. For example, in LTE, subframe is the TTI.
- **Minimum resource allocation unit (MRU):** This is the minimum unit of resources in frequency domain that can be allocated for scheduling.
- **Service scheduling policy (SSP):** This is scheduling policy applied to each service configured by the system or 3rd party through the API.

3.2 Scheduling Framework

We model a framework for scheduling users and their logical channels based on the parameters defined earlier. The primary task of this framework is two-dimensional buffer management at per-user per-service level which results in higher resolution for scheduling algorithms. In addition, the purpose of the framework is to develop a modular approach:

- Every module has a well-defined specific functionality that will contribute to performance enhancement at the system level;
- Modular approach adds flexibility to the scheduling framework and it can be integrated conveniently into different standards. Every module can be individually altered depending up on the constraints and system requirements. Note that this framework provides a generic solution to enhance the performance of existing scheduling algorithms for different wireless standards. The following sections describes the framework structure in terms of its interfaces and modules of MAC-layer as shown in Fig. 3.2.

3.2.1 MAC-layer Interfaces

In order to manage buffers, select scheduling algorithm and allocate resources to users, the MAC-layer requires the knowledge of all the parameters defined in Section 3.1. These parameters are received by the MAC-layer from system and interfaces towards other layers, namely physical (PHY) and radio link control (RLC) layers.

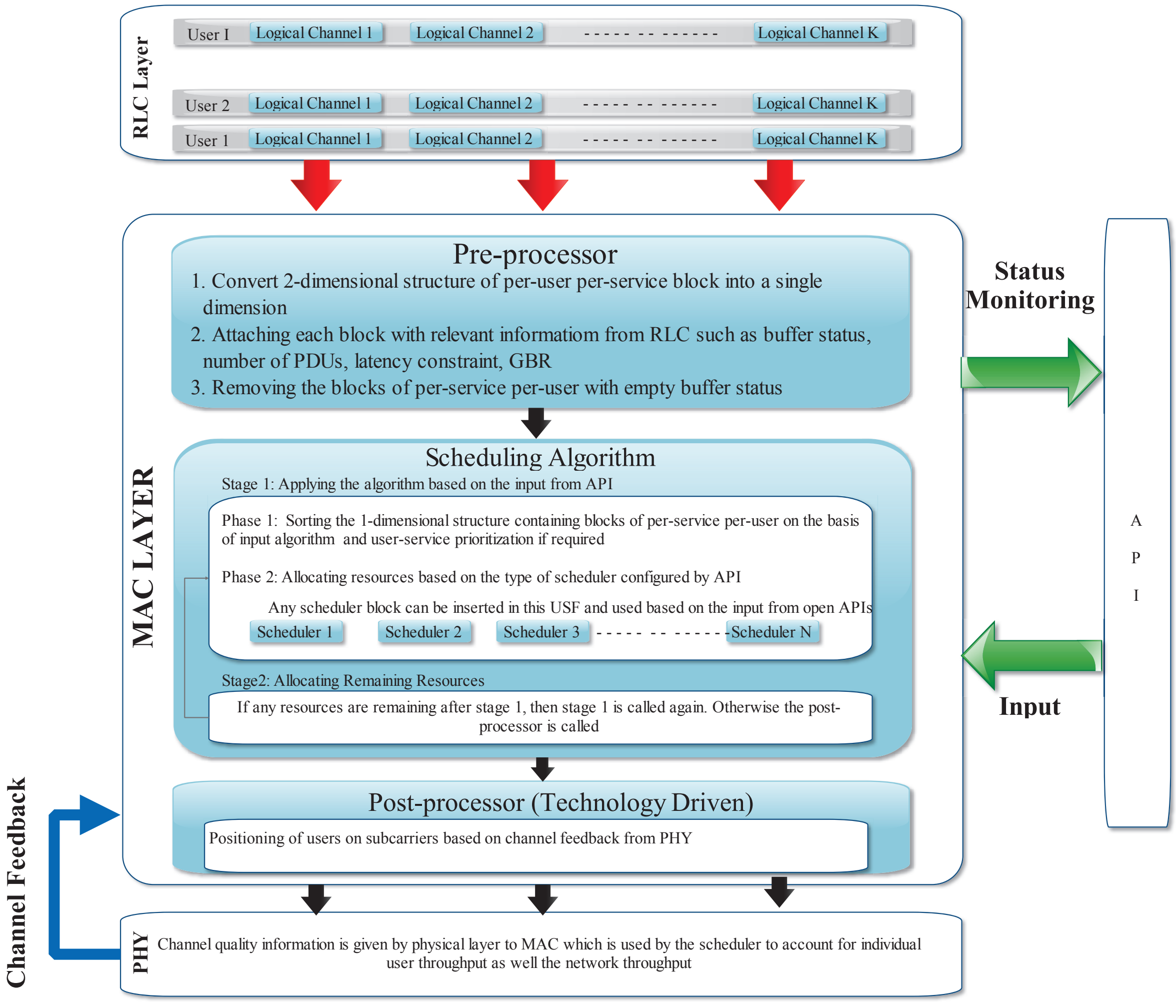

RLC Layer
User 1
Logical Channel 1
Logical Channel 2
----- -- ------
Logical Channel K
User 2
Logical Channel 1
Logical Channel 2
----- -- ------
Logical Channel K
User 1
Logical Channel 1
Logical Channel 2
----- -- ------
Logical Channel K
MAC LAYER
Pre-processor
1. Convert 2-dimensional structure of per-user per-service block into a single dimension
2. Attaching each block with relevant informatiom from RLC such as buffer status, number of PDUs, latency constraint, GBR
3. Removing the blocks of per-service per-user with empty buffer status
Scheduling Algorithm
Stage 1: Applying the algorithm based on the input from API
Phase 1: Sorting the 1-dimensional structure containing blocks of per-service per-user on the basis of input algorithm and user-service prioritization if required
Phase 2: Allocating resources based on the type of scheduler configured by API
Any scheduler block can be inserted in this USF and used based on the input from open APIs
Scheduler 1
Scheduler 2
Scheduler 3
----- -- ------
Scheduler N
Stage2: Allocating Remaining Resources
If any resources are remaining after stage 1, then stage 1 is called again. Otherwise the post-processor is called
Post-processor (Technology Driven)
Positioning of users on subcarriers based on channel feedback from PHY
Status Monitoring
Input
A
P
I
Channel Feedback
PHY
Channel quality information is given by physical layer to MAC which is used by the scheduler to account for individual user throughput as well the network throughput

Figure 3.2: *MAC-layer Scheduling Framework*

MAC-RLC

This interface shares the packet information of users and logical channels with the MAC-layer. Each logical channel represents a service with specific values to the parameters defined in Section 3.1. Through this interface, a two dimensional data structure of users and logical channels along with their associated parameters is created. These parameters informs the scheduler about traffic characteristics of every service in given transmission time interval.

MAC-PHY

Most of the modern scheduling algorithms are channel-aware, therefore it is crucial for MAC-layer to have a knowledge of the channel quality information of all the active users in the system. Once the channel estimation is done at the user terminals, the result is sent to the base station via feedback channel. The physical layer receives this information and forward through MAC-PHY interface to MAC. Channel quality information is used while calculating the expected throughput for a user and in turn for the entire system.

3.2.2 MAC-layer Modules

The proposed framework includes three main functional modules, namely pre-processor, scheduler, and post-processor. Once the MAC has a knowledge of all the necessary parameters through its interfaces, then the following three modules proceed with their defined tasks.

Pre-processor

This module represents a novel extension to the traditional scheduling framework. Main function of the pre-processor is to convert the two-dimensional buffer of users × logical channels into a single dimension as shown in Fig. 3.3. In the following each element is denoted by term block. For each block there holds $Buffer[n][k] > 0$, where the $n \in N$ and $k \in K$.

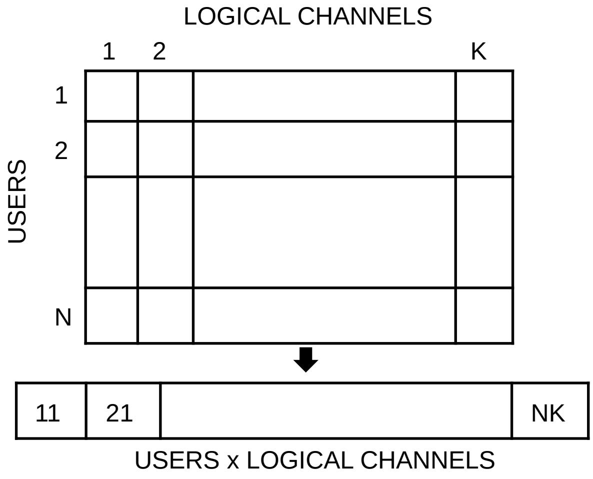


Figure 3.3: *Transform from 2-dimensional to 1-dimensional system*

The pre-processor's algorithm is given by Algorithm 1. As a result of this pre-processor, the scheduler module receives input in terms of block and its associated parameters. The traditional one-dimensional framework can easily be converted to this two-dimensional framework with the addition of this module and existing scheduling algorithms does not require to change in order to be implemented with this framework.

Scheduler

Once the conversion to single dimension is done in the pre-processor, the first task of scheduler module is to deal with sorting of blocks based on scheduling requirements. For example, in case of proportional-fair (PF)scheduling algorithm, blocks are sorted in decreasing order with respect to channel quality information. In round robin case, the blocks remain in order of their index. After block sorting, the actual allocation of resources to the sorted blocks is done in two stages:

1. **Service driven:** Service-level scheduling is performed based on logical channel priority across all the users. In this stage, the logical channels are processed vertically to compute the total number of resources for every services.

Algorithm 1: Convert *users* × *logical channels* into *blocks*

```
Input :
        let $n$ be the index of users $\in N$..
        Let $k$ be the index for logical channels $\in K$.
        Let $b$ be the counter of blocks.
Output:
        Let $block$ be the results Blocks (c.f. Fig. 3.3
$n = 1$;
$b = 1$;
while $k \leq$ NLC do
    $n = 1$;
    while $n \leq$ NU do
        if Buffer[n][k] $> 0$ then
            $block[b] =$ Buffer[n][k];
            $b = b + 1$;
        end
        $n = n + 1$;
    end
    $k = k + 1$;
end
```

2. **Operator-driven:** User-level scheduling is performed based on the allocation policy reconfigured through API, e.g. maximum throughput, proportional fair scheduling. At this stage, all the users are processed horizontally to distribute the resourcecs among users for a given service.

As can be seen, one of the major difference between a traditional framework and our proposed framework is the resolution of scheduling and flexibility in applying different scheduling policy per user and per service. Since the input to the scheduling module is blocks, the sorting and allocation is done at block level rather than user level.

Post-processor

The post-processor implementation depends on the wireless standard. The mapping of users to resource elements in the frequency domain is applied based on the specifications of the system. For example, in 3GPP LTE, the selection of resource block groups (RBGs) for a particular service of a user is done by post-processor. Number of RBGs depends on the system bandwidth. In LTE, RBGs are the smallest unit of frequency resources that can scheduled for a user.

3.3 Comparative Analysis

This section studies the gain due to two-dimensional buffer management within the proposed framework. In addition, an example of simple scheduling algorithm is given to explain the implementation differences with the framework. Then, simulation results are presented for few traditional scheduling algorithms. The main motive is to show performance gain of scheduling algorithms in terms of system throughput, fairness-index and percentage of satisfied users and services (those achieving GBR). The primary reason for comparing these traditional scheduling algorithms is to demonstrate the effectiveness of the proposed framework even with simple algorithms. However it will be proven that the performance improvement will happen for most of the scheduling algorithms and therefore it would be applicable for even more complex algorithms. To characterise the gain obtained at the MAC layer, perfect decoding is assumed in the physical layer. Therefore all the scheduled packets are assumed to be successfully received.

For any scheduling algorithm, sorting of users on the basis of a pre-defined performance metric is a primary step for efficient allocation of users. Sorting of users and allocating resources in the corresponding order contribute to the performance gain of scheduler. For example, suppose there are four users in a system with channel quality (2,1,8,4) and they are required to be sorted in decreasing order of channel quality. Then the sorted list will be (8,4,2,1) and naturally it will provide the best possible performance. It can be seen that as a result of sorting, four users changed their position and each contributed to improved performance. If only one user was sorted, then performance would not be optimal but better than unsorted list. Therefore with every sorting step, the performance increases till it reaches the best scenario. In order to generalize this explanation, let us define a variable G_1, which is the gain due to the change in position of single user and when x is the number of users that changed position due to sorting, then the total gain would be $x \times G_1$. However, the total gain also depends on the number of users that can be actually scheduled, for that Equation 3.1 gives the conditional total gain TG applicable to any traditional scheduling framework.

$$\mathrm{TG_t} = \begin{cases} x \times G_1, & \text{if } x \leq N' \\ N' \times G_1, & \text{if } x > N' \end{cases} \tag{3.1}$$

where N' is the actual number of scheduled users and N is the number of total active users. N' is a subset of N and it can be deduced from Equation 3.1 that $\mathrm{TG_t} = [0, N \times G_1]$.

With the proposed framework, the scheduler gets an input of blocks from the pre-processor and sorting is done for blocks as depicted in Fig. 3.3. Consequently the conditional total gain is

$$\mathrm{TG_p} = \begin{cases} y \times G_2, & \text{if } y \leq B' \\ B' \times G_2, & \text{if } y > B' \end{cases} \tag{3.2}$$

where G_2 is the gain due to the change in position of single block, y is the number of sorted blocks and B' is the number of scheduled blocks. B' is a subset of $N \times K$ and it can be deduced from Equation 3.2 that $\mathrm{TG_p} = [0, N \times K \times G_2]$.

By comparing Equation 3.1 and 3.2 in a similar scenario, it can be concluded that $\mathrm{TG_p} \geq \mathrm{TG_t}$ since the parameters x, G_1 and N' in Equation 3.1 always fit inside y, G_2 and B' in Equation 3.2 respectively.

For the example case, the round-robin algorithm is considered. All the parameters defined in the algorithm are self-explanatory. First, the pseudo code of round-robin algorithm without the framework is shown. As can be seen from Algorithm 2, scheduling is done only at the user levels. After this step, the resources allocated to each user are used for logical channel in order of their increasing index i.e $alloc\_resources\_per\_lc\_per\_user\,[n]\,[k]$. Then the implementation of round-robin algorithm with the framework is shown in Algorithm 3. Note that scheduling is done for blocks that have been shown in Fig. 3.3. This means that the scheduler has a higher resolution of scheduling and is able to consider the constraints and requirements on per-logical channel per-user basis. This leads to more efficient buffer management and optimal allocation of resources.

Algorithm 2: Calculate $alloc\_resources\_to\_user$ (Round-robin without framework)

```
while $number\_of\_resources\_remaining > 0$ do
    while $number\_of\_users\_scheduled \leq max\_allowed\_sched\_user$ do
        $alloc\_resources\_to\_user\,[n] =$
        $min\_resource\_alloc\_unit + alloc\_resources\_to\_user\,[n]$;
        if $user\_scheduled\,[n] \neq 1$ then
            $user\_scheduled\,[n] = 1$;
        end
        $number\_of\_users\_scheduled = number\_of\_users\_scheduled + 1$;
        $number\_of\_resources\_remaining =$
        $number\_of\_resources\_remaining\ + min\_resource\_alloc\_unit$;
    end
end
```

Algorithm 3: Calculate $alloc\_resources\_to\_block$ (Round-robin with proposed framework)

```
while $number\_of\_resources\_remaining > 0$ do
    while $number\_of\_blocks\_scheduled \leq max\_allowed\_blocks$ do
        if $number\_of\_users\_scheduled \leq max\_allowed\_sched\_user$ then
            $allocated\_resources\_to\_block\,[j] =$
            $min\_resource\_alloc\_unit + allocated\_resources\_to\_block\,[j]$;
            if $user\_scheduled\,[n] \neq 1$ then
                $user\_scheduled\,[n] = 1$;
            end
            if $block\_scheduled\,[j] \neq 1$ then
                $block\_scheduled\,[j] = 1$;
            end
            $number\_of\_users\_scheduled = number\_of\_users\_scheduled + 1$;
            $number\_of\_blocks\_scheduled = number\_of\_blocks\_scheduled + 1$;
            $number\_of\_resources\_remaining =$
            $number\_of\_resources\_remaining\ + min\_resource\_alloc\_unit$;
        end
    end
end
```

In terms of complexity, it can be seen that both the frameworks have the two iteration loops, the only difference is the number of iteration steps with the proposed framework. The number of iteration steps for traditional algorithm is the number of users, while for the proposed algorithms, it the number of blocks which is greater. Thus, the complexity is $O(nk)$, where n is the total number of users and k is the resources. Therefore, without any large increase in complexity, significant gains can be achieved for the existing algorithms with the proposed framework. It should also be noted, that this comparison of complexity is between the frameworks and is therefore applicable to any scheduling algorithm.

3.3.1 Simulation Setup

For the simulations, 3GPP LTE setting with its system parameters are used [48]. The simulations are done for a system bandwidth of 5MHz with 25 resource blocks and 1000 frames with 10 subframes each. The duration of a frame is 10ms. In addition to the system parameters, standard quality of service class defined in LTE are implemented using 9 logical channels for different services with varying QoS [49]. The traffic is generated based on the average packet size and average inter-arrival time for each service [50, 51]. The actual inter-arrival time between two packets for a given service of a user (block) is affected by TL (ranging on $[0, 1]$) and given as:

$$\mathrm{APIT} = \begin{cases} \frac{\mathrm{PIT}}{\mathrm{TL}}, & \text{if TL} \neq 0 \\ \text{no traffic}, & \text{otherwise} \end{cases} \tag{3.3}$$

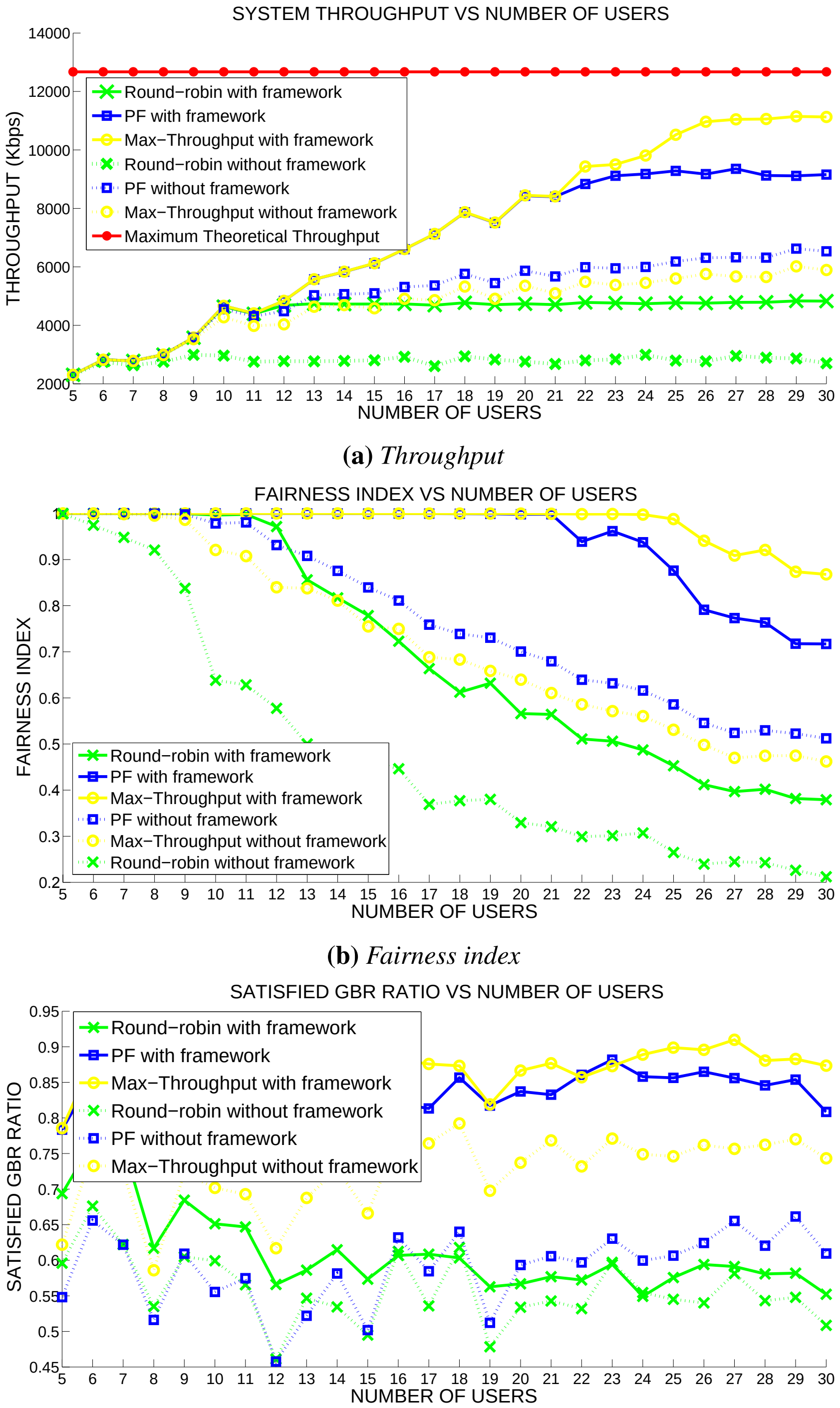
where TL is randomly generated.

3.3.2 Results

Performance analysis is done for three traditional scheduling algorithms: round-robin, proportional-fair and maximum-throughput [45]. In order to compare the performance of these scheduling algorithms with and without the proposed framework, system throughput, fairness index and satisfied GBR percentage (percentage of users that are satisfied) are plotted. All the performance metrics only refer to MAC-layer scheduling performance since it is assumed that there is a perfect decoding at the physical layer. In Fig. 3.4(a), the achievable throughput against number of active users in the system are compared. It can be observed that the performance of three considered scheduling policy is improved by a factor of 2 when the framework is applied. To set a benchmark, the theoretical system throughput is plotted.

In order to give a more clear picture, the fairness index and the satisfied GBR percentage are plotted in Fig. 3.4(b) and 3.4(c) respectively. The satisfied GBR percentage refers to the percentage of the logical channels (services) that are served with a data rate greater than or equal to GBR. It can be observed a similar pattern as in Fig. 3.4(a) that indicates performance enhancement when the framework is applied. It shows that the framework not only increases the throughput but also results in improved fairness index and provides better quality of service to users with higher percentage of satisfied GBR.

In Fig. 3.5 throughput is shown when using different traffic patterns. While TL is randomly generated in case of Fig. 3.4, in Fig. 3.5 it is selected such that the traffic pattern is similar to



(a) *Throughput*

(b) *Fairness index*

(c) *Satisfied GBR ratio*

Figure 3.4: *Comparison*

that of a residential area in evening when the services such as streaming, conversational video calls, are requested by large number of users.

The main purpose of this plot is to demonstrate the performance variation of scheduling algorithms with the change in traffic pattern. It can be concluded that there is no single scheduling algorithm optimal for every scenario. The traffic is continuously evolving in modern wireless

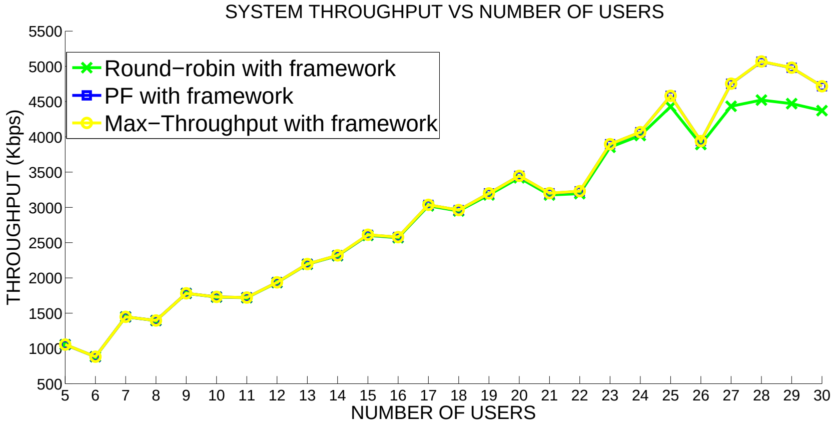


Figure 3.5: *Throughput comparison of scheduling algorithms in USF for traffic scenario with higher demand for conversational and streaming services*

networks and therefore static scheduling frameworks are no more an optimal solution. There is a need for more dynamic and adaptive approach. The modular nature of our framework enable us to append any additional module without disrupting the entire framework. Henceforth, an additional module consisting of intelligent APIs could be appended that would act as an interface between MAC-layer and external factors such as operator requirements and network constraints. These intelligent APIs would be capable of dynamically configuring scheduling algorithms depending up on traffic patterns and several other factors.

The results show a clear performance gain in terms of throughput, fairness-index and percentage of satisfied GBR users through our framework and also pointed out that there is necessity for a dynamic reconfigurability of scheduling algorithm.

3.4 Initial Experimentation and Validation

The scheduler framework discussed above is developed in the MAC layer of the OpenAirInteface LTE eNB, and its complete implementation is available online [52]. A set of experiments is carried out with the same set up as in the previous experiment (c.f. Section 2.5) to validate the operation of the scheduling framework with the maximum spectral-efficiency policy, and characterize its performance in terms of round trip time (RTT) considering only one logical channel. In particular, the RTT is evaluated through different traffic patterns generated by D-ITG [44], namely 32, 125, 512, 1408 packet sizes (PS), and 1, 0.5, 0.25, and 0.1 inter-departure time (IDT).

Fig. 3.6 demonstrates the measured RTT as a function of PS and IDT at 5 meters (setup 1) and 20 meters (setup 2) of distance from the eNB. As expected, higher RTT values are observed when PS, IDT, and distance increase. However, low RTT variations are observed, which is due to the fact that (a) the experiment is completely isolated, i.e. no external traffic, and no user mobility, and (b) resource allocation is performed as a function of traffic load in way to maximize the user spectral efficiency.

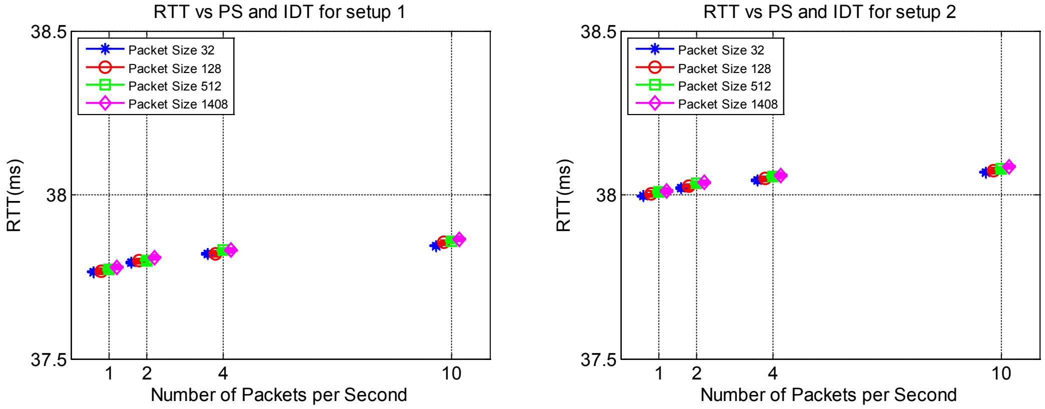


Figure 3.6: *Experiment 1*

3.5 Conclusion

This chapter presents a unified scheduler framework is also presented and applied to the LTE radio access network that enables more efficient allocation of resources to users running multiple internet applications in parallel. Comparative analysis have shown a clear performance gain in terms of throughput, fairness-index and percentage of satisfied GBR users and pointed out that there is necessity for a dynamic re-configurability of scheduling algorithm which will be addressed in our succeeding work. The resulted uplink and downlink scheduler have been implemented in the OpenAirInterface, and were tested and evaluated under real conditions.

Chapter 4

Low Latency Contention-Based Access

Low-latency protocols and access methods are becoming crucial, in view of emerging application scenarios found in machine-to-machine communication, online interactive gaming, social networking and instant messaging, to improve the spectral efficiency and to lower the energy consumption in end-devices. This pertains to multiple-access methods, scheduling, and connection management procedure. Particular to sensing devices, minimizing protocol and access latency during upload minimizes energy consumption in DSP and embedded processors.

However, the majority of wireless systems, including LTE/LTE-A, are designed to support a small number of connected devices with a continuous flow of information, at least in terms of the time-scales needed to send several IP packets such that the induced signaling overhead is manageable. While these systems are intended mostly for downlink-dominant and bursty traffic, emerging application scenarios are generally of different characteristics [53,54], namely:

- Uplink dominant packets (more packets in uplink than downlink);
- Small and low duty cycle packets (long period between two consecutive data transmissions);
- Periodic and infrequent event-driven packets;
- Raw and aggregated packets (combining traffic of multiple sources into a single packet, relevant for gateway nodes);
- Coordinated and uncoordinated packets (quasi synchronize access attempts on small time scale from many devices reacting to the same/similar/global events).

In addition, cellular networks are generally stateful and reactive. This is because the perceived quality of experience (QoE) can fall into different states depending on the user/application session and traffic pattern. Changes between the states not only depends on the current but also on the history of the session and traffic, which often results in different levels of performance, e.g. delay and data rate. Thus, the network is reactive to the user traffic [55]. More generally, the state is a function of spatio-temporal wireless network variability, application traffic patterns and requirements, user subscription type, resource allocation policy, and network configuration across different operators. This implies that the level of performance depends on the quadruple (operator, user/application, time, space). Fig. 4.1 sketches the behaviour of reactive network to application/user states, namely history, start, operation, and optimum. It can be seen the overall data rate and delay experienced by the application are fluctuating over time depending on the current user state.

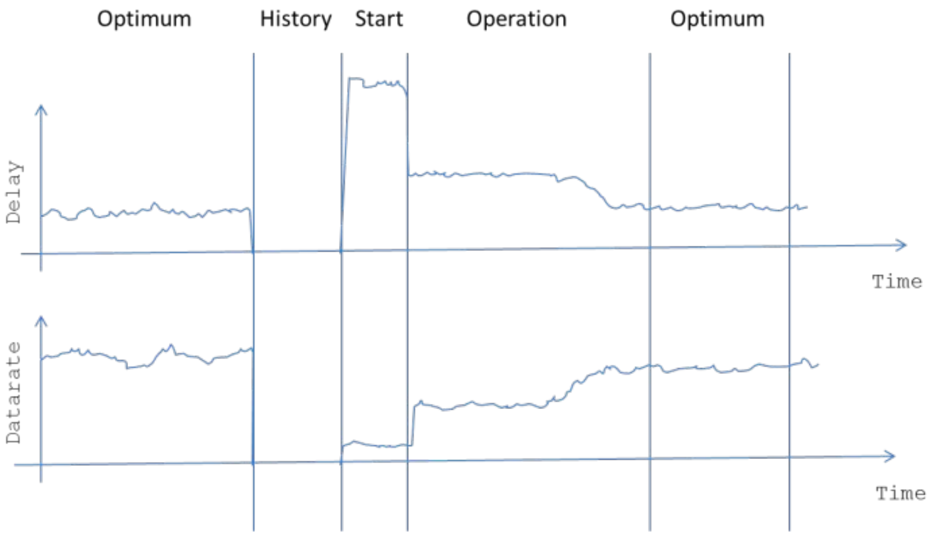


Figure 4.1: *Behaviour of reactive networks*

In view of the above characteristics, the evolving system requires an efficient channel access in terms of access latency and reliability in particular for small to very-small packets. In the literature, a lot of attention has been given to address uplink channel access using the random access as the main performance optimization. Nevertheless, questions related to alternative channel access methods and protocols, low complexity efficient scheduling and resource allocation algorithm, and low-resolution resource allocation have remained largely unaddressed. Furthermore, supporting coordinated traffic pattern, small packets, massive number of connected devices, and a number of other key challenges remain still open.

This chapter focuses on the design of an efficient uplink channel access method and a resource allocation policy applicable to current and next generation wireless systems capable of achieving a low latency communication in radio access networks and remain efficient for small packets (down to few tens of bytes).

The reminder of the chapter is organized as following. Section 4.1 elaborates on the notion of the latency in wireless communication systems followed by the latency analysis and measurement of the current LTE network. Section 4.2 introduces the general idea for the proposed contention based access (CBA) method. In Section 4.3, detailed low layer signalling enhancement to enable CBA technique in the current LTE specification (Rel. 11) is presented. Section 4.4 describes a resource allocation scheme for CBA. Section 4.5 and 4.6 provide extensive simulation and experimental results and analysis. Concluding remarks are presented in Section 4.8.

4.1 Latency in LTE

4.1.1 Definitions

In queueing networks delay has four components [56]: (1) processing delay, (2) queueing delay, (3) transmission delay and (4) propagation delay. While this is often true for the wired network (i.e. stateless behavior), in wireless networks the queuing and transmission delays also depend on the channel conditions and the history of the traffic pattern (including the inter-departure time as it bears information about the last packet), which in turn affect the scheduling decision, and consequently user and application state.

One of the important design objectives of LTE/LTE-A (and to some extent HSPA) has been to reduce the network latency which consists of c-plane and u-plane latency. The c-plane latency can be defined as the time taken by the first packet to successfully reach the receiver reference point. In the LTE/LTE-A, the c-plane latency is defined as a transition time between two states, from IDLE or INACTIVE to ACTIVE (c.f. Fig. 4.2). Typically, in the LTE/LTE-A the transition time from the IDLE to the ACTIVE state should be less than 100ms, and from the INACTIVE to the ACTIVE state depends on the DRX/DTX cycle [57].

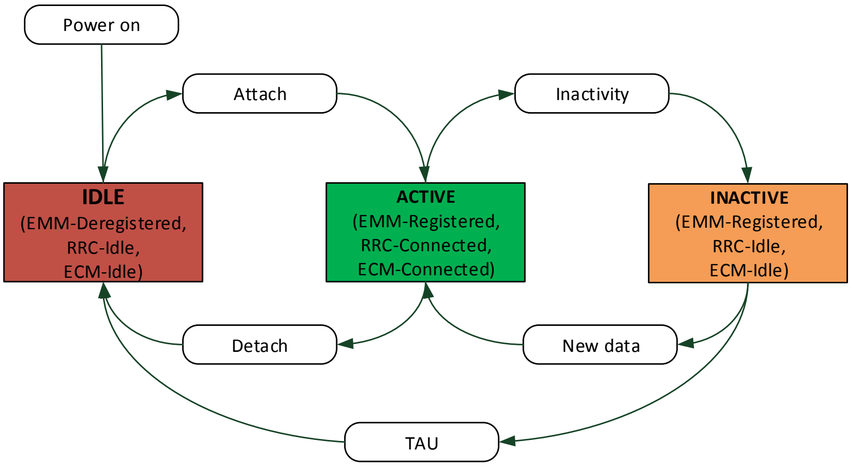


Figure 4.2: *LTE/LTE-A states*

The user plane latency, also known as transport delay, is defined as the one-way transit time between a packet being available at the IP layer of the sender and the availability of this packet at the IP layer of the receiver. In the LTE/LTE-A, this latency is defined between the UE and EPC edge nodes. The LTE/LTE-A specifications target the user-plane latency of less than 5ms in unloaded condition (a single user with a single data stream) for a small IP packet. As shown in Fig. 4.3, the 3GPP definition [58] leaves room to some ambiguity in three cases [3]:

- Packet unit: IP datagram or IP fragments;
- Availability: start or end;
- Intermediate interfaces: no IP.

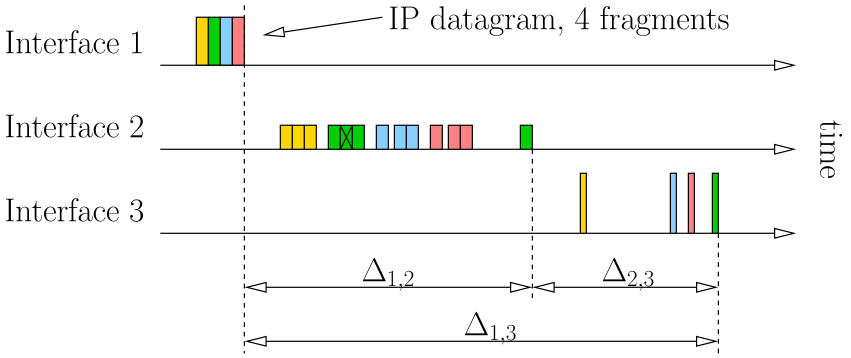


Figure 4.3: *Latency Definition*

In certain application scenarios, packets are small and (extremely) low duty-cycle, which from a system throughput perspective represents a vanishing data rate. In such low spectral-efficiency applications (seen by the application not the aggregate spectral efficiency), low latency protocols are crucial to achieve low-energy consumption in end-points

- minimizing c-plane signaling overhead, required prior to sending/receiving the first packet;
- minimizing u-plane protocol overhead, required for each sending/receiving packet.

Both overheads translates directly into energy- and performance-loss. Therefore, latency can also be defined in terms of the efficiency of very low-layer procedures allowing for time/frequency synchronization, identification/authentication, channel access time, channel interleaving, and channel code block length.

4.1.2 Analysis [2]

Table 4.1 highlights the main components in E-UTRAN and EPC contributing to the total access latency assuming the LTE/LTE-A FDD frame structure. The c-plane and u-plane establishment delays depend on the actual state of the UE, e.g. IDLE or ACTIVE. For the u-plane, the delay also depends on the traffic load and channel state, which is not considered in the analysis. This delay is the highest during the UE power-up when a UE transits from the IDLE state to the ACTIVE state and is zero in the ACTIVE state. In the ACTIVE, discontinuous reception mode (DRX) can be enabled. This mode is introduced in LTE to improve UE battery life time by reducing the transceiver duty cycle in the active operation. The DRX offers significant improvement with respect to the resource utilization, particularly for applications characterized by the ON-OFF periods or extended OFF periods. However, the cost associated with enabling the DRX mode is that there will be extended latency when a UE needs to wake up and transmits/receives data (see Table 4.2).

The u-plane latency depends mainly on scheduling policy, buffering and processing, TTI and frame alignment, number of retransmissions, and IP access delays. The delay associated with scheduling can be improved if there are pre-allocated resources (semi-persistent scheduling). The processing and queueing delay are assumed to be 3 ms for UE/(H/D)eNB/RN, and 1-3ms for/S+P-GW. The retransmission takes at the best 5ms (HARQ RRT time), and here it is assumed that the transmission error rate varies from 30% to 50% to estimate the retransmission latency. The latency for the (H/D)eNB and RN also depends on the cell load.

In the 3GPP LTE-A, relaying at RN can be classified according to which layer is used to forward the data, namely L0, L1, L2, and L3. The L0 relaying is the over-the-air amplify and forward the received signal and does not induce any delay. The L1 relaying is a digital buffer and forward the received signal inducing minimal buffering and processing delay compared to the L0 relay. The L2 relaying requires additional processing to decode and forward the received frame inducing at least one sub-frame delay and possibly scheduling delay. The L3 relaying, also known as self-backhauling, induces an additional delay as the RN terminates the layer 3 for the UE interface before forwarding the packet.

The core network interface delay through the S+P-GW is calculated for a propagation speed in copper cables of 200000 km/s, and distance between 2 nodes of 200-400 km.

Table 4.1: LATENCY BUDGET IN LTE/LTE-A

Latency Estimates	Latency Components
0 – 77.5ms	C-plane establishment delay - LTE idle to LTE active (47.5ms +2Ts1c*) - RRC idle to LTE active (37.5ms+2Ts1c*) - RRC connected to LTE active (13.5ms) - LTE active (0ms) *Ts1c (2-15ms) is the delay on S1 c-plane interface
0 – 28.5 ms	U-plane establishment delay - LTE idle to LTE active (13.5ms+Ts1u*) - RRC idle to LTE active (3.5ms+Ts1u*) - RRC connected to LTE active (3.5ms) - LTE active (0ms) *Ts1u (1-15ms) is the delay on S1 u-plane interface
5ms / 8 – 17ms	DL / UL data transmission (with 3ms processing delay) - DL U-plane Scheduling delay (1ms) - UL U-plane Scheduling delay (4ms) - UL U-plane Scheduling delay (request and grant) (9ms) - U-plane data transmission and reception (4ms)
4 – 7 ms	additional delay - U-plane (H/D)eNB/RN processing delay (1–3ms) - U-plane TTI and Frame alignments (1.5ms) - U-plane Retransmission 30%-50% for 5ms HARQ RRT (1.5–2.5ms) - U-Plane S/P-GW processing delay (1 – 3ms)

As Table 4.1 indicates, the c-plane establishment delay is dominant when comparing to the u-plane latency.

Table 4.2 shows the handover related latency for both data forwarding and radio processing for contention-free access [57]. The estimate may vary depending on the procedures. Six handover scenarios are possible among HeNBs, (D)eNBs, and RNs. Depending on the scenario, three main handover types are possible: intra-eNB, S1, and X2, with transitions to the same or different carrier frequency. For example, the data forwarding for the (D)eNB/RN handover scenarios is done through the X2 interface inducing lower latency than that of HeNB/eNB handover where the S1 handover is generally done through the IP backbone, thus adding variable delay. Please note that in a typical HeNB deployment, the HeNB is connected with the EPC via HeNB GW or directly over a fixed-line broadband access and the Internet.

3GPP has set requirements for the length of the detach time observed by the UE [59]. The maximum limit for handover delay is defined as

$$D_{handover} = T_{Search} + T_{Sync} + T_{RRC} + T_{Margin}$$

where T_{Search} represents the time required to identify the unknown target cell identity applicable only to the network-triggered handover (e.g. load-balancing). Otherwise, it is 0. T_{Sync} is the time to acquire the contention-free random access and get an uplink resource grant, and it is set to 30 ms. T_{RRC} is the RRCConnectionReconfiguration transaction processing time (happens before and after T_{Sync}) of up to 10ms. T_{Margin} is the implementation-dependent margin time upper-bounded to 20ms. Thus, the maximum detach time for known target cell must remain below 65ms [59–61].

Table 4.2: LATENCY BUDGET IN LTE/LTE-A

Latency Estimates	Description
10-512ms	Length of DRX cycle -Short DRX cycle (2-640ms) -Long DRX cycle (10-2560ms)
5-150ms	handover latency from source to target eNB: -intra-eNB handover (5-30ms) -X2 handover ((D)eNB-RN or eNB-eNB) (5-50ms) -S1 handover (HeNB-eNB or eNB-eNB) (15-150ms)

Fig. 4.4 presents the baseline end-to-end latency budget for each segment of the network without considering handover, DRX, and service delay. In the analysis, we assume that the interconnection between the devices/gateways and servers/users may be done either through the IP backbone or the IP Packet eXchange (IPX). The delay for the IP backbone (Internet) depends on the region as well as the number of nodes in the network, and processing delays in the nodes. For example, in Europe it could vary from 15ms up to 150ms [62]. This is also true for the IPX; in Europe the delay for interactive traffic class is in the range of 42 – 122ms. The delay can be increased depending on the service delay (i.e. application, 3rd party, and specific cellular services).

We note that the main latency bottleneck resides in the radio access network although the latency of the core network is non-negligible and depends on the middleboxes and the distance to the first IP access point. The bottlenecks vary depending on the applications, in particular DRX delay, handover and HARQ delays especially for mobility based scenarios, scheduling delay for different traffic profile, operator-specific cell configuration (e.g. scheduling request periodicity, random access availability per frame), and access and processing delays for high density scenarios. The overall delay also depends on the actual cell traffic load, outage probability, and radio propagation conditions. Access delay depends also on the proximity of the server/service

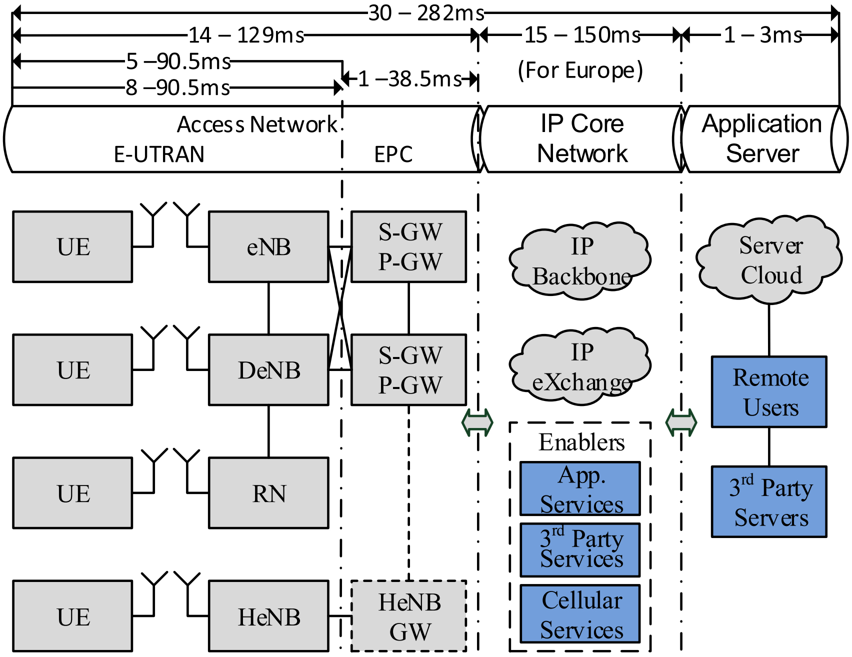


Figure 4.4: *End-to-end latency budget based on the above analysis.*

with respect to the terminal, and may be improved by placing the content and service closer to the user for instance in the operators' domain [63].

4.1.3 Measurements [3]

As stated earlier, 3G/4G mobile networks are stateful systems and the delay experienced by each packet depends heavily on the past history, namely, on the timing and size of the preceding packets transmitted and received by the same device. Therefore, different traffic patterns will inevitably result in different delay distributions. Moreover, while in fixed-capacity wired networks one can expect that increasing the data rate will result in larger packet delays (due to queuing), this is not a priori true in mobile networks, wherein radio resources are dynamically assigned to each terminal, depending on its actual traffic demand and channel state information. Therefore, *higher traffic rates could lead to a smaller delay in specific cases*. This is a fundamental differences between wired and mobile networks which invalidates the applicability of many bandwidth estimation methods to mobile networks. Based on such considerations, it should be clear that delay statistics obtained with a specific traffic pattern must be interpreted with caution and cannot be considered as exactly representative of the delay experienced by an application with a different traffic pattern.

A high precision latency measurements in operational LTE and HSPA networks are conducted to allow separately assess the one-way delay contributions of the radio access network and the core network for both technologies. In the considered benchmarking scenarios, packet timings are randomized as proposed in RCF 2330 [64, 65] to avoid synchronization effects

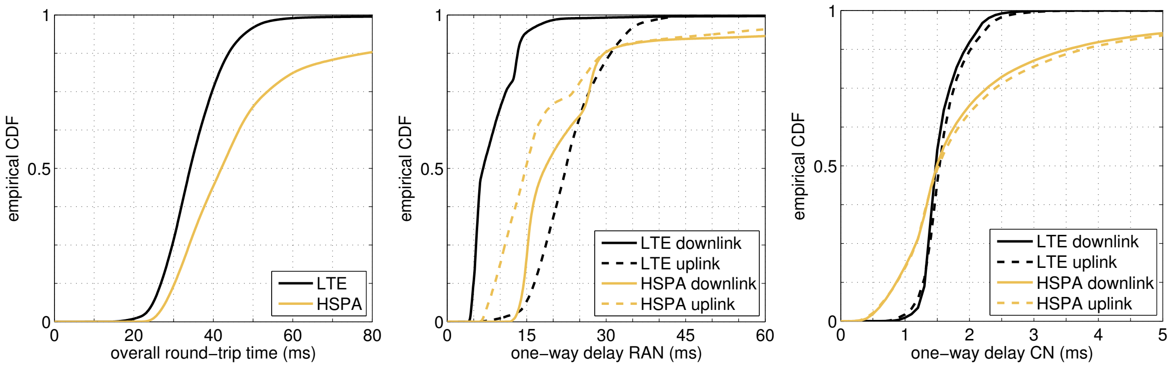


Figure 4.5: *Latency of different technologies for a broad range of packet sizes and data rates: (left) CDFs of the overall round-trip time for HSPA and LTE, (center) CDFs of the one-way delay caused by the radio access network, (right) CDFs of the one-way delay caused by the core network.*

(LTE and HSPA networks are synchronous) and guarantee the PASTA (Poisson Arrivals See Time Averages) property [66]. Probe packets are generated with random size and inter-arrival time.

The traffic generator deployed for the measurements is a custom user-space application. It produces two independent UDP streams for uplink and downlink, both at port 3000. Since we aim to investigate the influence of the data rate on the one-way-delay, the probe stream is organized into chunks of fixed data rate R. Each chunk has a duration of 150s, followed by an idle period of 10s, that forces the scheduling processes at the NodeB to return to a common state. Within each chunk the packet (datagram) sizes s_i at UDP layer are random, uniformly distributed between 10 and 5000 Bytes. The high upper limit shall enable to observe IP fragmentation effects. The packet inter-arrival times d_i are calculated for each packet i as $s_i = R \cdot d_i$. Consequently, the distribution of inter-arrival times is also uniform, with limits depending on the data rate R. Additionally, for all chunks we impose a lower limit of 1ms and an upper limit of 1s.

Round-trip Time and Individual One-way Delays

Fig. 4.5 compares the Cumulative Distribution Functions (CDFs) of LTE and HSPA Rel. 8 latencies.

RTT: In the leftmost graph the overall RTTs are compared. Since no RTT measurements are done, the respective values are synthesized from the uplink and downlink OWDs taken as independent random variables, namely, from the convolution of the respective Probability Density Functions (PDFs). It can be observed that the latency reduction brought by LTE is small, namely, 36ms compared to 42ms (or 14%) by consulting the median. However, comparing the minimum delay, it can be noted that LTE is able to perform much better than HSPA (11.9ms vs. 18.3ms or 35% reduction). We refrain from consulting mean or higher percentiles, because of the highly different load within both networks that causes a long-tail in the HSPA latency distribution.

OWD in RAN: The delay contribution of the RAN is presented in Fig. 4.5 (center). It is found that the downlink OWD in LTE is significantly lower than in HSPA. For the uplink the

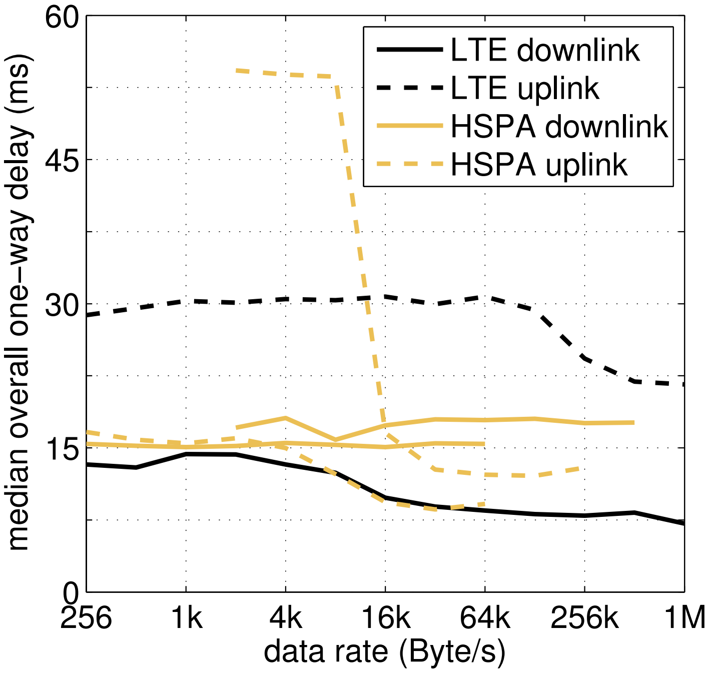


Figure 4.6: *Median one-way delay over data rate for different technologies. LTE: all packets smaller than 1500 Bytes. HSPA: two curves are shown per each direction, one for packets smaller 250Bytes (lower curves), the other for packet sizes between 1250 and 1500 Bytes (upper curves). The reason is the observed correlation of packet size and delay in HSPA. Generally OWD decreases with increasing data rate. HSPA outperforms LTE for small packets and/or low rates.*

situation is reversed. The main underlying reason is the difference in resource allocation and in cell configuration.

OWD in SAE/CN: The delay contributions through the SAE-GW (for LTE traffic) and the GGSN (for HSPA traffic) are shown in Fig. 4.5 (right). All different OWDs are minor compared to the overall RTT (note the different scale on abscissa). It can be seen that the CDFs for uplink and downlink OWD match very well for both technologies. Moreover the median OWD corresponds exactly to the values indicated by 3GPP [67].

One-way Delay and Data Rate

Fig. 4.6 depicts the median OWD obtained at different rates to further analyze the impact of data rate. The median allows for a more fair comparison even if the two networks operate at different loads. The focus here is on packet size below 1500 Bytes. For HSPA, two distinct curves per direction are plotted, one for small packets (size below 250 Bytes, lower curves) and one for large packets (size between 1250 and 1500 Bytes, upper curve).

The general trend in Fig. 4.6 shows that the delay is decreasing for an increasing data rate. In the uplink direction this effect is more distinct. For small data rates and low variations of the packet sizes, HSPA may show lower uplink OWD than LTE. The reason being the different uplink scheduling policies and cell configuration in both technologies described above. This explains the small gain in RTT of LTE compared to HSPA, which we observed in Fig. 4.5.

Discussion

In the following three delay sensitive applications are considered for latency comparison. Note that applications like file-downloads, web-browsing and messaging are not considered here, since the user-experience for those cases is only affected by RTTs bigger than 1s [68]. The considered traffic types are:

- **Online Gaming:** characterized by small constant sized packets with random inter-arrival times in uplink, and variable sized packets with constant arrival time in downlink. The data rate is between 1 kByte/s and 5 kByte/s for most cases. The delay required for good Quality of Service (QoS) lies below 50ms [69].
- **Machine-to-machine communication:** characterized by short and small in number packets with low duty-cycle, which from a system throughput perspective represents a vanishing data rate [53, 68]. In addition, such packets are more uplink-dominant and follow two traffic patterns: (i) periodic non-realtime packets such as keep alive or update messages and, (ii) event-driven realtime packets such as alarm notifications. The data rate is considered to be below 1 kByte/s. The delay requirements for proper functionality of future applications is debatable, realtime M2M applications may appear with RTTs below 25ms [69].
- **VoIP:** is characterized by constant small-sized packets with a constant inter-arrival time, hence, a constant data rate. The vast amount of different VoIP applications makes the firm characterization of traffic of this type rather difficult. Data rates may reach from 4 kByte/s up to 200 kByte/s. Further, RTTs of up to 200ms are sufficient for excellent user experience.

When examining the results presented above, we can conclude that

- The interactive traffic of Online Games can benefit from the lower delay in the downlink of LTE. A drawback is the increased uplink delay;
- The sporadic nature of M2M traffic patterns and the main traffic flowing in uplink (most M2M nodes are assumed to be sensors), leads to the conclusion that the latency performance of HSPA will be better in this case;
- VoIP delay requirements of 100ms can be achieved by both technologies. The higher delay of LTE in certain situations favors HSPA for some applications.

Table 4.3 summarizes the results. The green color indicates high user satisfaction, whereas red indicates that the QoS may be impaired.

4.2 General Idea of Contention Based Access

To provide a low latency uplink channel access, a new resource allocation method, called contention based access (CBA), is proposed. The main feature of CBA is that the eNB does not

Table 4.3: DELAY REQUIREMENT PER APPLICATION TYPE

Application	LTE (up/down)	HSPA (up/down)
Online Gaming	(31 / 13) ms	(12 / 17) ms
M2M	(30 / 10) ms	(10 / 16) ms
VoIP	(30 / 15) ms	(35 / 16) ms

allocate resources for a specific UE. Instead, the resources allocated by the eNB is applicable to all or a group of UEs and any UE which has data to transmit randomly uses resource blocks among the available allocations within the group(s) it belong to. As the resources are not UE specific, collision happens when multiple UEs within the same group use the same resource. To detect the collided UEs, their identities (RNTI in LTE/LTE-A) will be further protected and transmitted alongside with the data (i.e. different coding rate for control and data information is applied). This will enables eNBs to decode the control information despite the collision (based on MU-MIMO techniques), and consequently interpret such a collision as a scheduling request, which in turn triggers a scheduling grant for the collided UEs. The proposed methods can also be seen as a grant-free channel access, where signaling overhead caused by dedicated channel access, namely random access, SR, BSR, are bypassed.

To improve the resource efficiency and provide better QoS, multiple CBA groups could coexist in a cell, and a given UE could belong to multiple groups. Various grouping methods can be applied, namely grouping based on UE identity/location/proximity, QoS requirement, and traffic pattern. Fig. 4.7(left) illustrates an example of CBA with three groups, namely uplink CBA (group1), direct CBA (group 2), and aggregated CBA for gateway nodes (group 3).

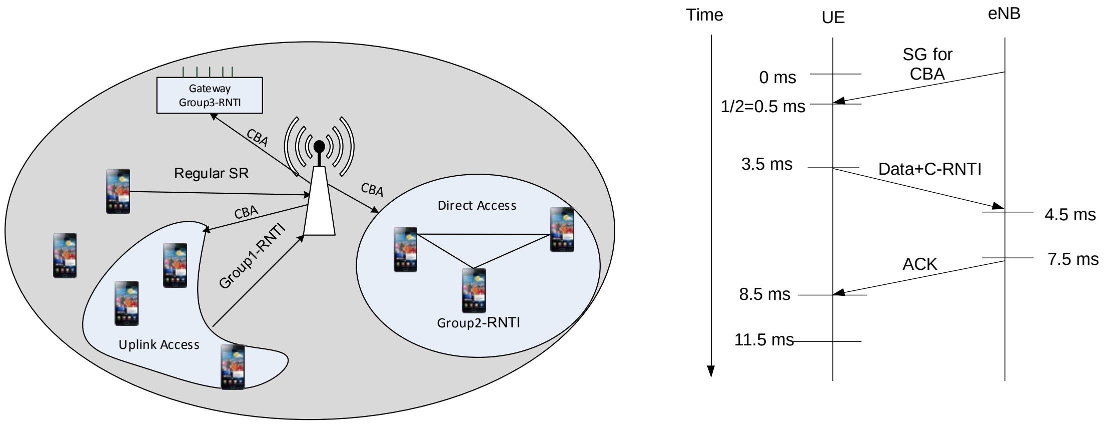


Figure 4.7: *Channel access with CBA*

The procedure for contention based uplink access is shown in Fig. 4.7 (right) and described below. Firstly, the UE receives the resource allocation information which indicates the resource allocated for CBA. Assuming the CBA resource is available in each subframe, a UE wait for 0.5 ms to receive the scheduling grant (SG) information for CBA. Then, after decoding the resource allocation information which costs 3 ms, the UE sends the frame on the randomly selected resources. The latency for this whole procedure is 7.5 ms in the best case, which is much smaller than that of the random access (27 ms). Here we assume that the UE is uplink synchronized, which does not necessarily mean the RRC connected state.

As the CBA resources are not UE specific but rather group-specific, collisions may happen when multiple UEs select the same resource. In a network with sporadic traffic, the collision probability is very low, which means most transmissions are free of collision and therefore CBA method can potentially outperforms the regular scheduling method in view of latency. However, in a high traffic load scenarios the collision probability is very high, which means a lot of retransmission are needed and hence the latency is increased. For example supposing the total available resource block in a subframe is 50, the collision probability is 0.06 if 3 UEs transmit in the subframe, while the collision probability increases to 0.99 if 20 UEs transmit in the subframe.

To solve the above problem, the following method is used. Each UE sends its identifier, C-radio network temporary identifier (C-RNTI), along with the data on the randomly selected resource. Since the C-RNTI is of very small size, therefore it can be transmitted with the most robust modulation and channel coding scheme (MCS) without introducing huge overhead. By the use of MU-MIMO detection, these highly protected CRNTIs might be successfully decoded even if they are sent on the same time-frequency resource. Upon the successfully decoding for the collided C-RNTIs, the eNB triggers regular scheduling for the corresponding UEs as shown in Fig.4.8(left). Therefore, a UE can retransmit the packet using the regular HARQ procedure. The overall latency for CBA with collision is 15.5 ms, which is still less than that of the regular scheduling.

For the collided UEs whose C-RNTIs are not decoded, neither dedicated resource (SG) nor ACK information is received; those UEs have to retransmit the packets as shown in Fig.4.8(right). It has to noted that the retransmissions is still based on CBA, which is referred as HARQ for CBA as it is different from the regular HARQ procedure (In regular HARQ, dedicated resource is allocated for a UE with retransmission).

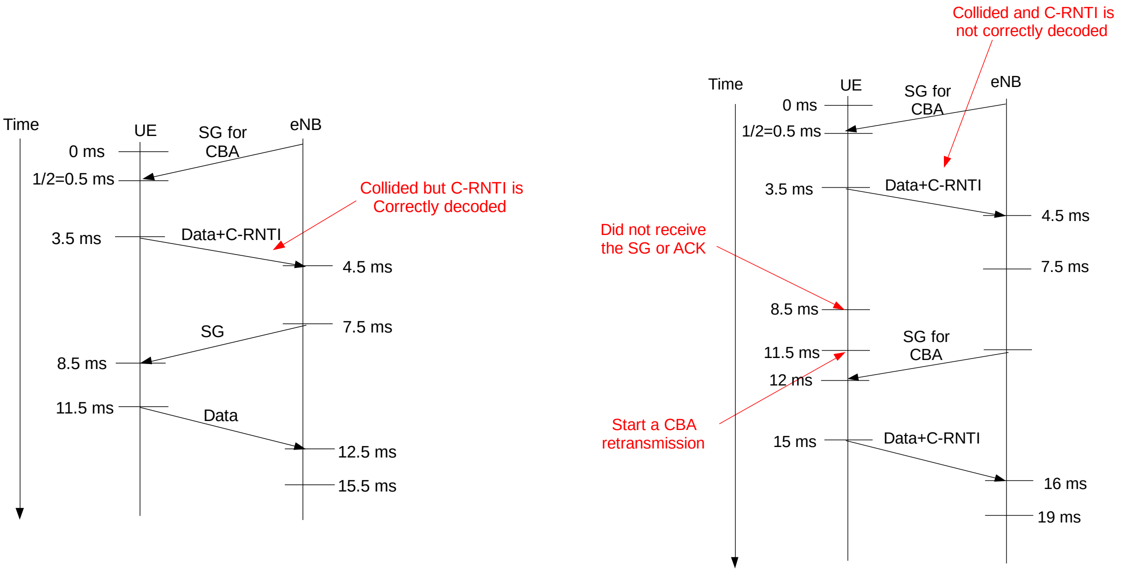


Figure 4.8: *Contention based access with collision detection(left) and with retransmission (right)*

4.3 Implementation of Contention Based Access in LTE

In order to implement the proposed contention based access method in LTE, some modifications are needed. Before presenting the detailed modifications to LTE, we first introduce the regular uplink scheduling method in LTE. Fig.4.9 demonstrates a regular uplink scheduling procedure in LTE:

1. a UE sends the SR information on the physical uplink control channel (PUCCH) to request resource from eNB.
2. a eNB decodes the SR packet and perform an initial (quasi-static) resource allocation for that UE.
3. a UE sends its buffer state report (BSR) on the allocated resource.
4. a eNB allocates suitable amount of resource for the UE according to its BSR information following the applicable resource allocation policy. The resource allocation information is sent with DCI 0 [1]. The MCS index used for uplink transmission and the cyclic shift used for reference signal are also specified in DCI 0. This DCI 0 information is attached by a 16-bit CRC, where the CRC parity bits are scrambled with the C-RNTI such that UEs can identify the UE-specific resource.
5. A UE uses its C-RNTI to identify its DCI 0 information. With this DCI 0 information, a UE finds the resource allocation information, the uplink MCS index and cyclic shift. Then it sends the packet on the allocated resource with the specified MCS and cyclic shift.

4.3.1 Enhancements to the RRC Signaling Related to 3GPP TS 36.331

Signaling to Inform UEs about the CBA-RNTI

The CBA-RNTI, which is used by a UE to decode the resource allocated information for CBA, is allocated by eNB during the RRC connection reconfiguration procedure for the data radio bearer (DRB) establishment. To implement this procedure, the CBA-RNTI has be added to the RadioResourceConfigDedicated information element as defined in [60]. It should be mentioned that the CBA-RNTI is not UE specific. Instead, all UEs or a group of UEs have a common CBA-RNTI configured by RRC signaling.

4.3.2 Enhancement to the PHY Signaling Related to 3GPP TS 36.212

Signaling to Inform UEs about the CBA Resource Allocation

To adapt to the resource allocation for CBA, a new DCI format, DCI format 0A, is defined. The DCI format 0A is used to inform UEs about the resource allocated to CBA. The content of

[1] DCI format 0 is used for uplink resource allocation, while DCI format 1/1A/1B/1C/1D and 2/2A is used for downlink.

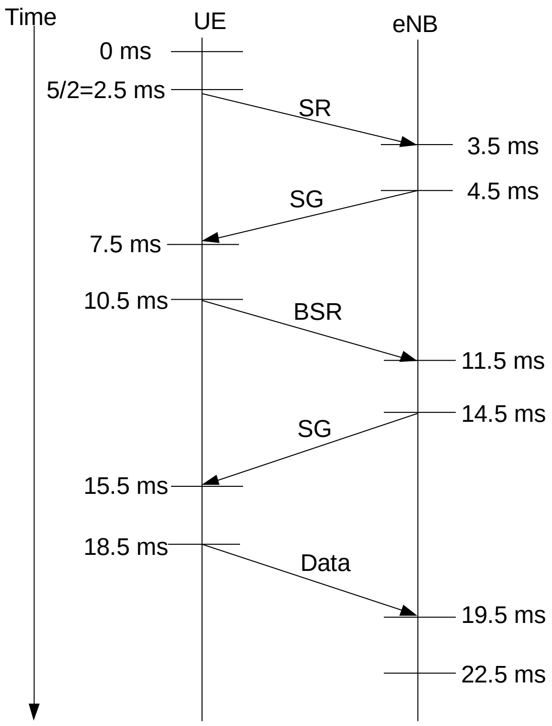


Figure 4.9: *Uplink packet scheduling in LTE*

DCI format 0A is shown in Tab.4.4, where N_{RB}^{UL} is number of resource block in the uplink. The CRC for DCI format 0A is scrambled with a new defined radio network temporary identifier CBA-RNTI. With the CBA-RNTI, the UE decodes the DCI format 0A to locate the resource allocated for CBA. As the resource allocation is not UE specific, multiple UEs may select the same resource, which causes collisions. Please note that DCI format 0 can also be used to convey the CBA resource allocation.

Table 4.4: FIELD OF DCI FORMAT 0A

Information	Type Number of Bits	Purpose
Hopping flag	1	Indicates whether PUSCH frequency hopping is performed
Resource block assignment	$\log_2 N_{RB}^{UL}(N_{RB}^{UL}+1)$	Indicates assigned resource blocks

Signaling to Inform eNB about the Selected MCS

In CBA, the MCS used for uplink transmission is not indicated by eNB. Instead, UEs determine the MCS independently [2]. Therefore, the UE should inform the eNB about the selected MCS index so that the uplink frame can be properly decoded. To inform the eNB about the selected MCS index, the following method is proposed.

A new type of uplink channel information (UCI) is defined as shown in Fig. 4.10, which includes the uplink MCS index as well as C-RNTI and transmitted along with the data. It has to be noted that the MCS for CBA data and control information are usually different. To achieve that, the CBA data and control information are treated independently: high MCS is used for CBA data while low MCS is employed for CBA control information. The resulted bit streams (one for CBA data and other for CBA control information) are then assembled for further processing, e.g., resource mapping, which is common in LTE. For example: the uplink channel quality indicator (CQI) can be multiplexed with data, and sent on PUSCH.

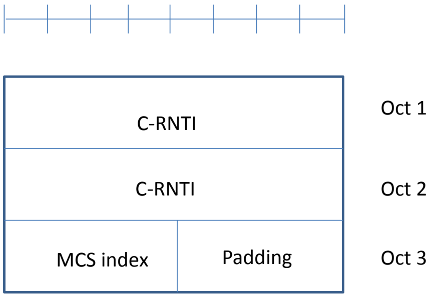


Figure 4.10: *Uplink channel information conveying MCS and C-RNTI for a CBA transmission*

4.3.3 Enhancement to the PHY Procedures related to 3GPP TS 36.213

UE Procedure to Locate the Resource Allocated for CBA

With the allocated CBA-RNTI which is obtained during the RRC connection reconfiguration procedure procedures, a UE can locate the resource allocated for CBA by decoding the DCI format 0A information.

UE Hybrid ARQ (HARQ) Procedure for CBA

UEs should find the ACK/NACK information after sending the data frame such that a new transmission or a retransmission can be properly performed. To adapt to CBA, the method to locate the ACK/NACK information is described as follows.

The ACK information is sent by eNB for a correct received CBA packet, which is the same as the one specified in [70]. The physical HARQ indicator channel (PHICH) index is implicitly

[2] In CBA, cyclic shift for uplink reference signal is used for channel estimation. The cyclic shift is random selected by UEs. The eNB identifies a UE's cyclic shift by trying all the possible values. The one with the highest peak value is considered as the used cyclic shift.

associated with the index of the lowest uplink resource block and the cyclic shift used for corresponding contention-based access. Therefore, UEs which successfully send frames can find the corresponding ACK information without extra signaling. The details for this method can found in [70]. If two UEs select the same resource and use the same cyclic shift, we assume that none of the two packets can be decoded by eNB as the eNB cannot correctly estimate channel information for these two UEs. As a result, no ACK is sent [3]. On the other hand, if two UEs select the same resource and use different cyclic shifts, it is possible that one of the two collided packet is correctly decoded. In this case, an ACK is sent. The UE with the correctly received packet can locate the ACK, the the other UE cannot find the ACK since it uses a different cyclic shift.

For those UEs whose C-RNTI is correct detected but data is corrupted (c.f. Section 4.2), the eNB triggers regular scheduling by sending the DCI 0 information (not DCI 0A). In the DCI 0 information, the new data indicator (NDI) field is set to 0 to represent the NACK information. Hence once a UE receives the DCI 0 information with NDI 0, it infers that the last transmissions is unsuccessful. And then, this UE starts a retransmission on the dedicated resource indicated by DCI 0 information. For the FDD system, the UE starts the retransmission three subframes later after receiving the DCI with format 0 as shown in Fig.4.8.

There are also some UEs whose C-RNTIs cannot be successfully decoded. For these UEs, they cannot receive any information at the expected time. As a result, new retransmissions with CBA are performed as shown in Fig. 4.8(right).

CBA Data Reception

The reception of a CBA data is different from the reception of regular data in that an eNB should first decode the UCI before decoding the CBA data. This is because an eNB first extracts the MCS and RNTI from the UCI in order to decode the ULSCH CBA payload. With this method, the eNB can try to decode the C-RNTIs for the collided UEs.

Resource Allocation for Correctly Received C-RNTI

For an erroneous packet with correctly decoded C-RNTI, the eNB allocates dedicated resource for the corresponding UE and informs the UE through the DCI 0 information. With this allocated resource, a UE can send the data packet without collision.

4.3.4 CBA Operation

With the above modifications to the LTE standard, we can implement the contention based access method using the flow presented in Fig.4.11(right) for the UE and in Fig.4.11(left) for eNB.

The steps for the UE operation is presented below.

[3] In some case, even if two UEs use the same cyclic shift, one of packets can be correctly decoded by eNB. As a result, an ACK can be received by both UEs, and thus one of the packet will be lost.

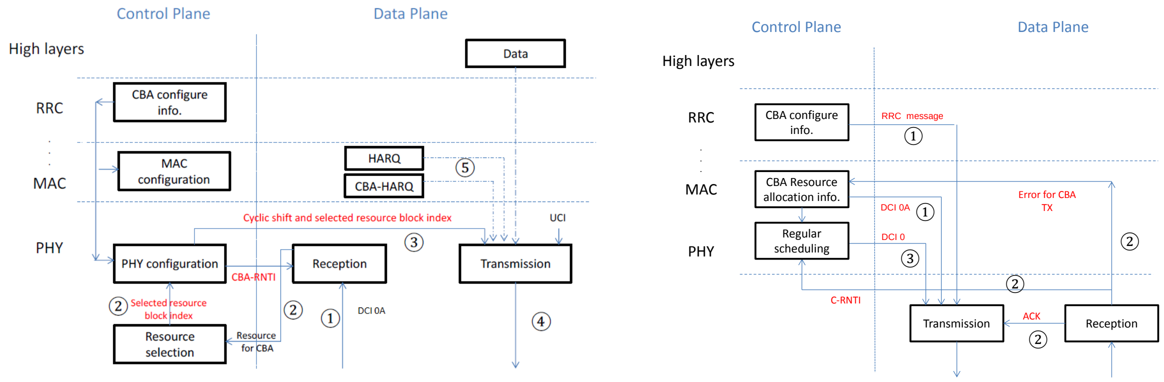


Figure 4.11: *CBA operation for UE (left) and eNB (right)*

1. **Reception:** Upon reception of RRC message, a UE configures its MAC and PHY layers. With the CBA RNTI, the UE decodes DCI 0A, and is able to locate the resource allocated for CBA transmission.

2. **Resource selection:** Selects randomly among allocated resources and inform physical layer about the resources.

3. **Configuration:** Configures a transmission using the randomly selected cyclic shift and the results from the resource selection module.

4. **Transmission:** Sends the data and the CBA control information following the instructed configuration.

5. **HARQ:** Retransmit the packet either through the regular HARQ procedure if a scheduling grant is received, or through CBA-HARQ procedure if nothing is received.

The steps for the eNB operation is presented below.

1. **Configuration and Resource Allocation:** Apply grouping policy and sends the CBA configuration information to UEs through the RRC message. Then, perform a resource allocation for CBA and send the DCI, where the CRC parity bits of the DCI is scrambled with the allocated CBA RNTI.

2. **Successful Reception:** Send an ACK for the correctly received CBA packets such that it can be sent on the PHICH channel.

3. **Unsuccessful Reception:** Decoded C-RNTIs of the collided UEs, and trigger a scheduling grant so that specific resources can be allocated for those UEs.

4.4 Resource Allocation Scheme for Contention Based Access

The main target for resource allocation is to assign the proper amount of resource such that the latency constraints are satisfied and the allocated resources are efficiently used. Accurate resource allocation for CBA is very important as it is directly connected to latency experienced by the application traffic.

Let us denote the total number of resource elements allocated for one CBA transmission as N_{RACH}. This contains the amount of resource elements used for control information transmission, denoted as N_{ctrl} in addition to those reserved for data N_{data}, i.e.,

$$N_{RACH} = N_{ctrl} + N_{data}. \tag{4.1}$$

Therefore, The spectral efficiency of the control information is

$$R_c = 24/N_{ctrl}(\text{bits/RE}) \tag{4.2}$$

under the assumption that the control information comprises 24 bits (16 bits for C-RNTI, 4 bits for MCS and 4 padding bits). Similarly, the spectral efficiency of the data is

$$R_d = M_{data}/N_{data}(\text{bits/RE}) \tag{4.3}$$

where M_{data} is the bit of data payload.

For each contention based access transmission, we have the following events:

1. neither the control information nor the data are detected, which is denoted as E_1;
2. the control information is not detected but the data is detected, which is denoted as E_2;
3. the control information is detected but the data is not detected, which is denoted as E_3;
4. both the control information and data are detected, which is denoted as E_4.

In order determine the probability of each event we take a an approach based on instantaneous mutual information. This asymptotic measure yields a lower bound on the above probabilities for perfect channel state information at the receiver. To this end, the received signal of *m*th antenna at resource element *k* is

$$y_m[k] = \sum_{u=0}^{N_u-1} H_{m,u}[k]x_u[k] + Z_m[k], m = 0, \cdots, N_{\text{RX}} - 1 \tag{4.4}$$

where $H_{m,n}[k]$ is the channel gain for user u at antenna m, $x_u[k]$ is the transmitted signal, $Z_m[k]$ is the noise, and N_u is the random number of active users transmitted on this resource block.

The normalized sum-rate for N_u contending users based on mutual information for both data and control portions is computed as

$$I_{\text{X}} = \frac{1}{N_u N_{\text{X}}} \sum_{k=0}^{N_{\text{X}}-1} \log_2 \det \left(\mathbf{I} + \sum_{u=0}^{N_u-1} \gamma_u \mathbf{H}_u[k] \mathbf{H}_u^*[k] \right) \tag{4.5}$$

where X is represents either control or data, $\gamma_n, n = 0, \cdots, N_u - 1$, is the received signal-to-noise ratio (SNR) and $\mathbf{H}_i[k] = \begin{pmatrix} H_{0,n}[k] & H_{1,n}[k] & \cdots & H_{N_{\mathrm{RX}}-1,n}[k] \end{pmatrix}^T$. The use of this expression requires the two following assumptions. Firstly, all channels can be estimated at the receiver irrespective of the number of contending users. This has to make proper use of the cyclic shifts to guarantee contention-free access for channel estimation. In practice, for loaded cells with only CBA access, this will require association of UEs to orthogonal CBA resources (in time/frequency) and on a particular CBA resource a maximum of 12 contenting UEs can be accommodated. Secondly, the expression assumes Gaussian signals and that the eNB receiver uses an optimal multi-user receiver (i.e. it performs complete joint detection.) These expressions can be found in [71].

Assuming there are i active UEs contending on the same CBA resource unit, for a given UE the probabilities of the four events caused by one CBA transmission are:

$$P_{E_1,i} = P_{S,i} + (1 - P_{S,i})p(I_{ctrl} < R_c, I_{data} < R_d), \tag{4.6}$$

$$P_{E_2,i} = (1 - P_{S,i})p(I_{ctrl} < R_c, I_{data} > R_d), \tag{4.7}$$

$$P_{E_3,i} = (1 - P_{S,i})p(I_{ctrl} > R_c, I_{data} < R_d), \tag{4.8}$$

$$P_{E_4,i} = (1 - P_{S,i})p(I_{ctrl} > R_c, I_{data} > R_d), \tag{4.9}$$

where $P_{S,i}$ is the probability that other UEs use the same cyclic shift on one CBA resource unit provided that there are i contending UEs (we assume that if the multiple UEs select the same cyclic neither control information nor data can be decoded by eNB as the eNB cannot correctly estimate the channel). In general, the control information is more protected than the data, i.e., $R_c < R_d$, so $P_{E_2,i} \approx 0$.

Then the expected value for the probabilities of the four events are:

$$P_1 = \sum_{i=1}^{N} P_{A,i} P_{E_1,i} \tag{4.10}$$

$$P_2 = \sum_{i=1}^{N} P_{A,i} P_{E_2,i} \approx 0, \tag{4.11}$$

$$P_3 = \sum_{i=1}^{N} P_{A,i} P_{E_3,i} \tag{4.12}$$

$$P_4 = \sum_{i=1}^{N} P_{A,i} P_{E_4,i} \tag{4.13}$$

where $P_{A,i}$ is the probability that there are i active UEs contending on one CBA resource unit; N is the total amount of UEs in a cell.

To minimize the latency for the MTC traffic, the CBA resource should be available in each subframe. The resource allocation can be performed in the following steps:

1. Set the CBA resource unit

2. Initialize the amount of CBA resource unit to 1
3. Calculate the probabilities of the four events caused by a CBA transmission.
4. Calculate the latency based on the measured amount of CBA resource unit
5. If the estimated latency is larger than the latency constraint, increase the amount of resource unit by one and go back to step 3. Else end

It can be seen that as the latency decreases with amount of CBA resource unit, therefore with the above method we always find the minimum amount of CBA resource. It has to noted that here we assume that there is always enough resource. For a system which has a constraint on CBA resource, more intelligent scheduler can used to address this problem, for example a scheduler which consider the priorities between real time and non-real time traffics.

4.4.1 Estimation of the Probabilities of Events in Step 3

To estimate the probabilities of the four events caused by a CBA transmission, we drive a Semi-Markov chain model as shown in Fig. 4.12, where

- S_0 means that there is no packet in the UE's buffer;
- S_{2i-1}, $i \in [1, M]$, means the ith CBA transmission of the UE, where M is the transmission limit;
- S_{2i}, $i \in [1, M-1]$, means that the UE is waiting for the ACK or SG information.

The UE transfers between states as:

- When the UE is at state S_0, if a packet arrives, it transfers to states S_1 to start the first transmission; otherwise it remains at state S_0;
- When the UE is at state S_{2i-1}, $i \in [1, M-1]$, it sends the packet and transfers to S_{2i};
- When the UE is at state S_{2M-1}, it sends the packet and transfers to S_0;
- When the UE is at state S_{2i} $i \in [1, M-1]$: (1) if ACK is received it transfers to state S_0; (2) if SG is received it sends the packet as shown in Fig. 4.8(left) and then transfers to state S_0; (3) if neither ACK nor SG is received at the expected time instant, it transfers to state S_{2i+1} to retransmit the packet as shown in Fig. 4.8(right).

Denoting $p_{i,j}$ as the state transition probability from state S_i to state S_j, $i, j \in [1, 2M-1]$, the state stationary probability of state i can be calculated as:

$$\begin{cases} \pi_0 = \pi_0 p_{0,0} + \sum_{i=1}^{M-1} \pi_{2i} p_{(2i),0} + \pi_{(2M-1)} p_{(2M-1),0} \\ \pi_{2i-1} = \pi_{2i-2} p_{(2i-2),(2i-1)}, i \in [1, M] \\ \pi_{2i} = \pi_{2i-1} p_{(2i-1),(2i)}, i \in [1, M-1]. \end{cases} \tag{4.14}$$

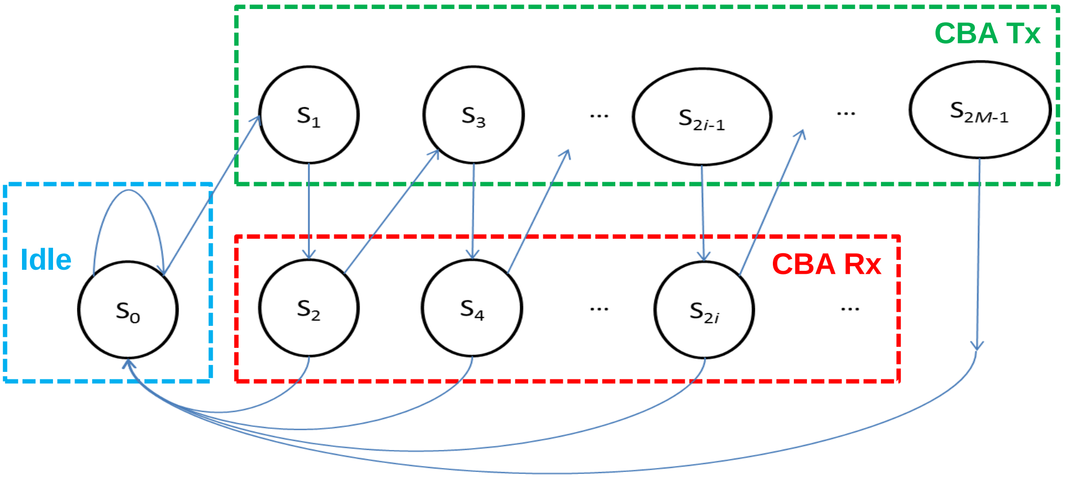


Figure 4.12: *Markov chain model for contention based access*

With the above equations, we can get

$$\pi_i = \prod_{j=1}^{i} p_{(j-1),j}\pi_0, i \in [1, 2M-1]. \tag{4.15}$$

Substituting (4.15) into the following equation

$$\sum_{i=0}^{2M-1} \pi_i = 1, \tag{4.16}$$

we can get

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{2M-1} \prod_{j=1}^{i} p_{(j-1),j}}. \tag{4.17}$$

The state transition probability can be calculated as following. In each subframe (1ms) if a packet arrives, the UE transfers from state S_0 to state S_1. Supposing the packet arrives following a Poisson distribution with the arrival rate λ, we have $p_{0,1} = 1 - e^{-\lambda}$. When the UE is at state S_{2i-1}, after transmission it transfers to state S_{2i}, therefore $p_{(2i-1),2i} = 1$, $i \in [1, M-1]$.

When the UE is at state S_{2i}, it transfers to state S_{2i+1} if neither ACK nor SG is received, i.e., it transfers state S_{2i+1} if neither the control information nor the data are detected, therefore

$$p_{2i,(2i+1)} = P_1. \tag{4.18}$$

With derived transition probability, π_0 can be calculated as

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{M-1} 2(1-e^{-\lambda})P_1^{i-1} + (1-e^{-\lambda})P_1^{M-1}} \tag{4.19}$$

and π_i can be calculated using (4.15). We can see that π_i, $i \in [1, 2M-1]$, is a function of P_1.

Now let us calculate the state holding time D_i (in ms) for state S_i, $i \in [1, 2M-1]$. In state S_0 for every subframe the UE checks if a packet arrives. If so, it transfers to state S_1, therefore $D_0 = 1$.

In state S_{2i-1} as shown in Fig. 4.7(right) the UE first waits for resource allocation information for CBA and then sends the packet; finally it transfers to state S_{2i}, therefore $D_{2i-1} = 3.5$, $i \in [1, M]$.

When the US is in state S_{2i}: (1) if ACK is received it transfers to state S_0, the state holding time for this case is $11.5 - 3.5 = 8$ms as shown in Fig. 4.7(right); (2) if SG is received it sends the packet on the allocated resource as shown in Fig. 4.8 and then transfers to state S_0, the state holding time for this case is $11.5 - 3.5 = 8$ms; (3) if neither ACK or SG is received at the expected time instant, the UE transfers to state S_{2i+1} to start a retransmission as shown in Fig. 4.8(right), the state holding time for this case is also $11.5 - 3.5 = 8$ms. Hence, $D_{2i} = 8$, $i \in [1, M-1]$.

Denoting Q_i, $i \in [1, 2M-1]$, as the proportion of time that the UE is in state i, it can be calculated as

$$Q_i = \frac{\pi_i D_i}{\sum_{i=0}^{2M-1} \pi_i D_i}, \tag{4.20}$$

which is a function of P_1.

A UE trigger a CBA transmission in state S_{2i-1} and the time used for a CBA transmission is 1ms. Therefore the probability that a UE is performing a CBA transmission is

$$\tau = \sum_{i=1}^{M} Q_{2i-1} \frac{1}{D_{2i-1}}. \tag{4.21}$$

which is also a function of P_1.

For a UE which is performing a CBA transmission, the probability that there are i another UEs contending on the same CBA resource is

$$P_{C,i} = \sum_{j=i}^{N-1} \binom{N-1}{j} \tau^j (1-\tau)^{N-1-j} \binom{j}{i} (\frac{1}{N_{RE}})^i (1 - \frac{1}{N_{RE}})^{j-i} \tag{4.22}$$

where $i \in [0, N-1]$, N is the total amount of UEs in a cell and N_{RE} is the amount of CBA resource unit.

Therefore, the probability that there are i contending UEs use the same CBA resource unit is

$$P_{A,i} = P_{C,(i-1)}, i \in [1, N]. \tag{4.23}$$

which is a function of τ.

Moreover assuming the amount of UE which contends on the same CBA resource is i, for a given active UE the probability that other UE selects the same cyclic shift is

$$P_{S,i} = 1 - (\frac{11}{12})^{i-1}. \tag{4.24}$$

It has to be mentioned that above equation holds since the maximum available cyclic shifts in one CBA resource unit is 12. Hence, with equations (4.24) and (4.6) we can calculate $P_{E_1,i}$ for i contending UEs.

With the above results, the probability for the first event is

$$P_1 = \sum_{i=1}^{N} P_{A,i} P_{E_1,i} \tag{4.25}$$

which is a function of τ.

We can see that equations (4.21) and (4.25) comprise a system of equations with two unknown P_1 and τ, which could be solved by numerical methods. Hence, we can calculate P_3 and P_4 using (4.12)-(4.13), respectively.

4.4.2 Estimation of the Latency in Step 4

With the results derived in last subsection, we can estimate the latency for given amount of CBA resource.

As stated at the beginning of this section, for each CBA transmission we have four events. Here we denote the packet transmission latency for the four events as T_1, T_2, T_3, and T_4 (in ms), respectively. Hence the average latency can be calculated as:

$$T = P_1T_1 + P_2T_2 + P_3T_3 + P_4T_4. \tag{4.26}$$

As $P_2 \approx 0$, so the above equation can be simplified as:

$$T = P_1T_1 + P_3T_3 + P_4T_4. \tag{4.27}$$

For an unsuccessful CBA transmission where both data and control information cannot be decoded retransmission happens 11.5ms after the initial transmission as shown in Fig. 4.8(right), therefore T_1 can rewritten as $T_1 = T_5 + 11.5$, where T_5 is packet delivery latency for a new CBA transmission. Moreover, as shown in Fig. 4.7(right) and 4.8, we have T_3=15.5, and T_4=7.5. With the above results, we have $T = (T_5 + 11.5)P_1 + 15.5P_3 + 7.5P_4$.

Since $E(T) = E(T_5)$, the expected channel access latency is

$$E(T) = \frac{11.5P_1 + 15.5P_3 + 7.5P_4}{1 - P_1}. \tag{4.28}$$

where P_1, P_3 and P_4 are calculated in the third step.

4.5 Simulation Results

To evaluate the performance of proposed contention based access method and its resource allocation scheme, simulations are performed using MATLAB. We assume that the system is operating FDD-LTE; the SNR is set to 5 dB; transmission limit M is set to 5 and the number of receiving antennas is 2; the coding rate for the control information $R_C = 0.2$. For simplicity we have assumed a line-of-sight dominant channel model with randomized angle-of-arrival at the eNB receiver in order to model the $\mathbf{H}_i[k]$. The CBA resource unit is set to be 6 resource blocks, i.e. $6 \times 12 = 72$ subcarriers, which is same as the resource of the PRACH channel. Moreover, the packet size is assumed to be of small size, following an exponential distribution with average packet size of 100 bits. The packet arrival rate λ is 1/100 packet/ms.

4.5.1 Performance Evaluation

Firstly, to validate the proposed contention based access (CBA) method, we compare the channel access delay of CBA with that of random access (referred as PRACH method). We compare these performance of two methods with the same amount of resources. Concretely, for the CBA method, we allocate one CBA resource unit containing 6 resource blocks in every subframe. While for the PRACH method, the preamble is set as 64 and the PRACH resource configuration index is set to 14, which occupies the same resource as CBA (6 resource blocks in every subframe) and it is the maximum allowed resource for PRACH in LTE. The transmission limit for random access is 5; the random access response window size is 10 subframes; and the contention resolution timer is 24 ms. Table 4.5 shows simulation parameters.

Table 4.5: SIMULATION PARAMETERS

Parameter	***Value***
PRACH Configuration	14
RAR Window Size	10 SF
Contention Resolution Timer	24ms
SNR	5dB
Transmission Limit M	5
Number of RX Antenna	2
Coding Rate	0.3
CBA Resource Unit	6RB
Avg. Packet Size	16 Bytes (128bits)
Packet Arrival Rate	100ms

Fig.4.13 shows the simulation results. We can see that the latency of CBA is much smaller than that of the PRACH method. It shows that the latency gain to use CBA is around 30 ms, which validates that CBA outperforms the PRACH method in the term of latency.

In the following, we analyze the effect of coding rate of control information, number of receiving antenna, CBA resource unit size on the CBA performance.

The coding rate R_c determines the amount of resource used for control information. More resource is used for control information when the coding rate decreases, which makes the control information more robust to the wireless channel error. However, since the resource for CBA transmission is fixed (the resource is shared by information and data), the resource

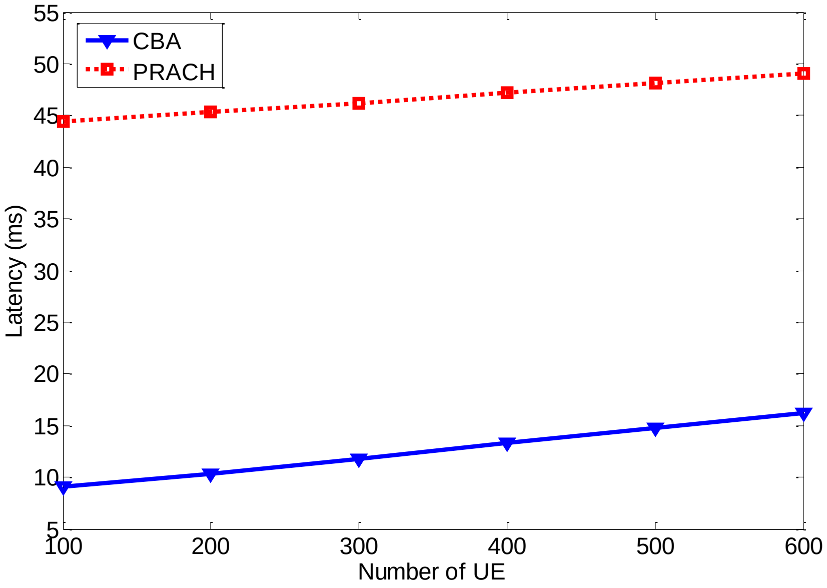


Figure 4.13: *Latency comparison*

used for data transmission is reduced when the coding rate decreases, which indicates that the data becomes more sensitive to wireless channel error. A robust control information is more likely received by eNB which reduces latency, while a sensitive data is more easily corrupted which in turn increases latency. Therefore, the coding rate R_c has strong effect on the CBA performance.

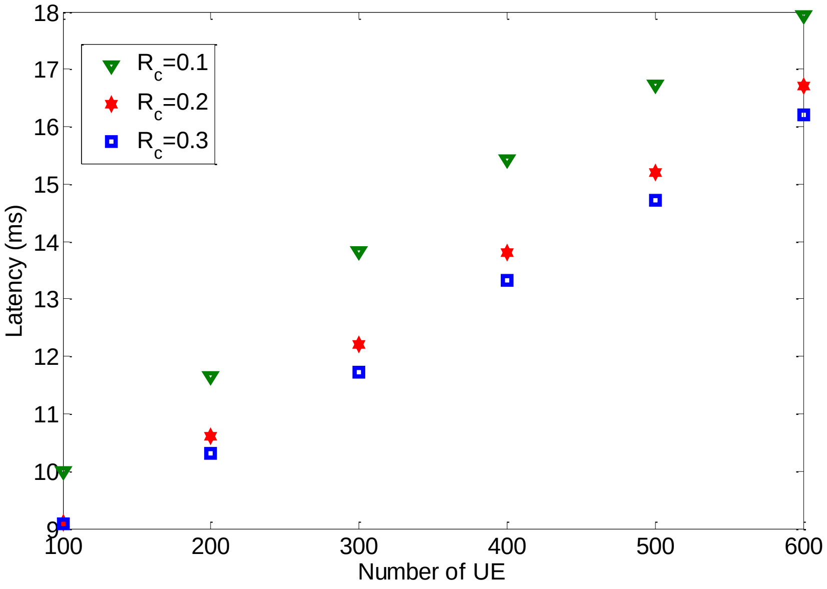


Figure 4.14: *Effect of R_c on the CBA performance*

Fig. 4.14 shows the effect of R_c(coding rate for the control information) on the CBA performance. We can see that the latency when $R_c = 0.3$ is less than other two cases. For example, the latency is 15.6ms when $R_c = 0.3$ and number of user is 600, while it is 15.9 or 17.4 when R_c equals 0.2 or 0.1, respectively. Therefore, to achieve the best performance of CBA, the code rate for the control information should be carefully selected.

Using more receiving antenna increases the successful rate to receive the control information as well as data. Fig. 4.15 presents the latency under different number of receiving antennas. It can found that the latency decreases when the number of receiving antennas increases. For example, the latency decreases from 15.9 ms to 12.2ms when the number of UE is 600 and number of receiving antenna increases from 2 to 3, and it further reduces to 10.9 ms when the umber of receiving antenna increases to 4.

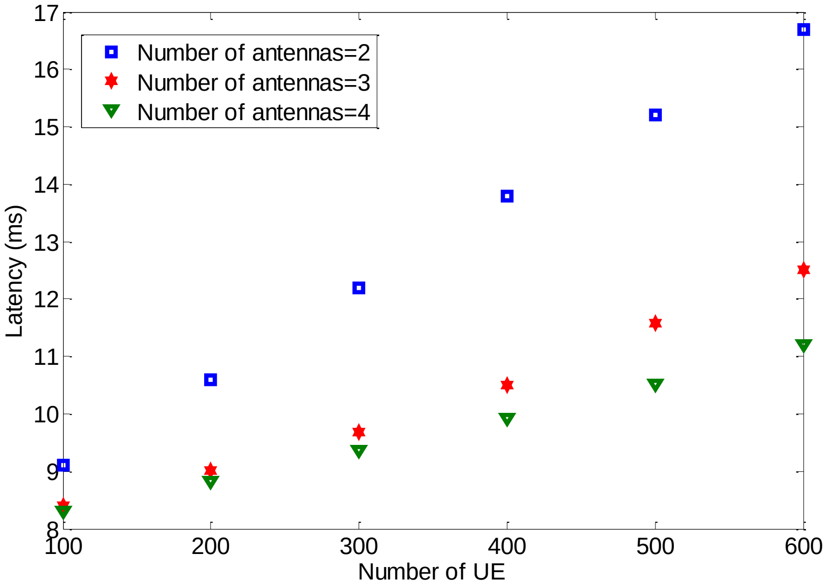


Figure 4.15: *Effect of number of receiving antennas on the CBA performance*

Assuming the total amount resource allocated for CBA is fixed, the size of the CBA resource unit also effect the performance. Larger CBA unit size is beneficial to the transmission of data and control information. However, larger CBA unit size yields smaller amount of CBA resource unit, which increases the collision rate for cyclic shift and hence the latency. Fig. 4.16 demonstrates the latency under different CBA resource unit. It is shown that by setting the CBA resource unit size to 6 resource blocks, the minimum latency can be achieved. Therefore, to attain the best performance of CBA, the CBA resource unit size should be carefully tuned.

4.5.2 Performance of the Resource Allocation Scheme

Fig. 4.17(left) shows the resource allocation results using our proposed method with different packet arrival rates λ (packets/ms) and number of UEs when the delay constraint is 30ms. We can see that the allocated resource units non-decrease with the increase number of UEs and/or

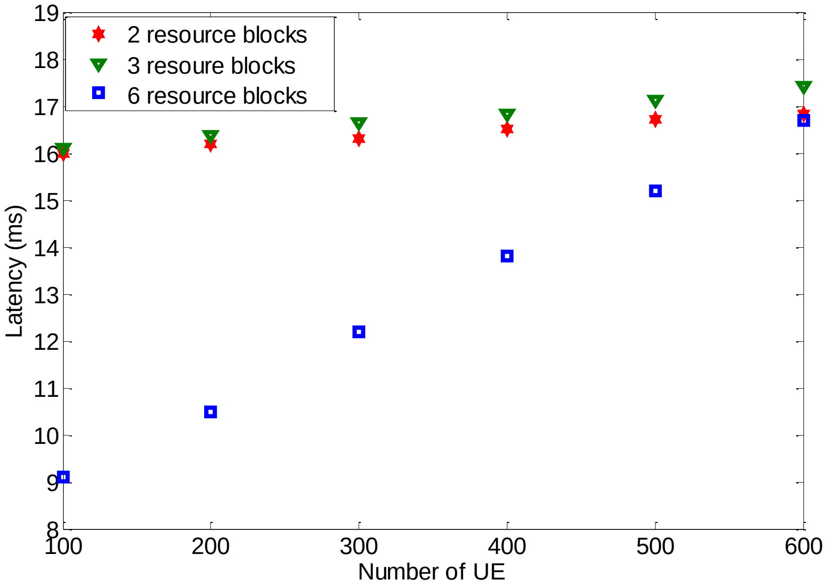


Figure 4.16: *Effect of CBA resource unit size on the CBA performance*

packet arrival rate. This is because the packet collision rate increases with number of UEs and packet arrival rate, which hence increases latency. To satisfy the delay constraint, more resource should be allocated. For instance, when $\lambda = 1/30$ and the number of UE is 300, the CBA resource unit is one and the latency is 28.9 ms which is very close to the threshold 30ms. Therefore, when the number of UE increases to 400, two CBA resource units are allocated which reduces the latency to 18ms. Similarly, when the number of UEs reaches 600, the CBA resource unit is increased to three, and the latency decreases to 18ms. Fig. 4.17(right) demonstrates the delay when using the allocated amount of resource shown in Fig. 4.17(left). It can be seen that the delay is smaller than the delay constraint 30ms, which validates the proposed resource allocation method.

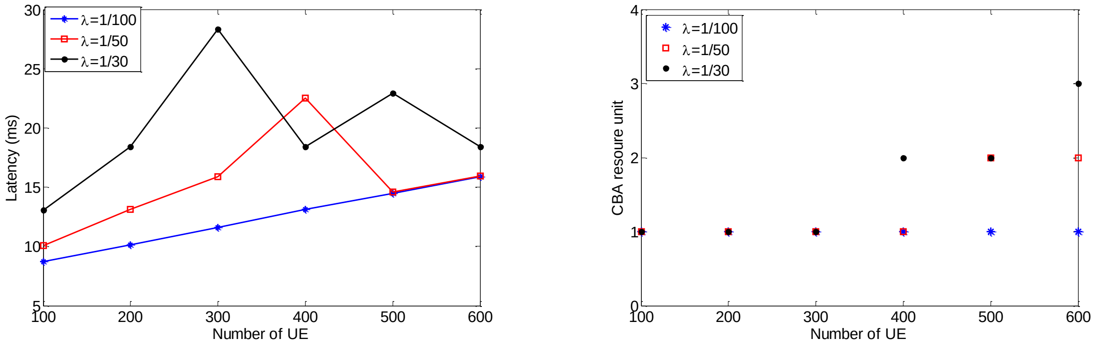


Figure 4.17: *Number of allocated resource unit (left) and latency (right) of CBA resource allocation scheme for different number of UEs*

4.6 Experimental Results

This section first presents details of the experiment setup. Then, the applied grouping scheme and scheduling algorithm are detailed. Next, performance results in terms of latency, goodput, and number of collisions considering both uncoordinated and coordinated traffic patterns are provided. The rationality of CBA is validated through the results obtained using the OpenAirInterface system-level emulation platform. Protocol operations across all the layers are also validated in appendix A.

4.6.1 Experiment Setup

To evaluate the performance of CBA in a realistic setting, the OpenAirInterface emulation platform is used. Full implementation of the proposed method is available online [72]. The method is integrated with the legacy 3GPP access layer protocols for both eNBs and UEs. The impact of three types of parameters are investigated, which are described below.

1. **Traffic Pattern:** The benchmarking methodology presented in [55] are enhanced and applied here to evaluate the performance of CBA. A subset of popular and emerging applications are selected to perform the measurements with their representative traffic patterns. As illustrated in Fig. 4.18, each considered applications represents an area or a point in a two dimensions, namely packet size PS in bytes and inter-departure time (IDT) in second. The region of interest is bounded by $PS_{min} - PS_{max}$ for the packet size, $IDT_{min} - IDT_{max}$ for packet inter-departure time, and $R_{min} - R_{max}$ for the data rate. The boundaries are selected to cover most of the popular applications. For the uncoordinated traffic pattern, the PS and IDT are uniformly sampled over 10 non-overlapping areas for a duration of 1 minute (corresponds to 6000 LTE frames), and for the coordinated traffic pattern, both PS and IDT are fixed for each area and that the traffic is generated exactly at the same time. The generated traffic pattern is not a single packet transmission but represents an entire application session for a duration of 55 seconds with a variable and fixed traffic pattern. The traffic generated using Openair traffic generator [30]. The respective measurement results consist of multiple figures highlighting the CBA performance trend for each traffic pattern represented by the considered area. However, the results does not capture the performance for a specific set of parameters or applications and need to be interpreted with caution.

2. **Backoff window size:** Backoff is used UE to avoid collision by spreading CBA transmission over a time window determined by a backoff counter. Thus, a large backoff counter increases the time a UE has to defer before a CBA transmission. To see the effect of backoff on the performance of CBA, different backoff values are used (c.f. Table 4.6). When backoff is zero, a terminal transmit with a probability 0.5. Note that backoff is one of the key parameter of CBA, especially for coordinated or heavy traffics.

3. **Number of CBA groups:** Allocating UEs into larger number of groups is another key parameter of CBA, and it is used to reduce the collisions. This is because the total number of transmissions per group is proportionally reduced. Here, a simple identity-based grouping is applied to partition UEs into a set of non-overlapping groups (c.f.

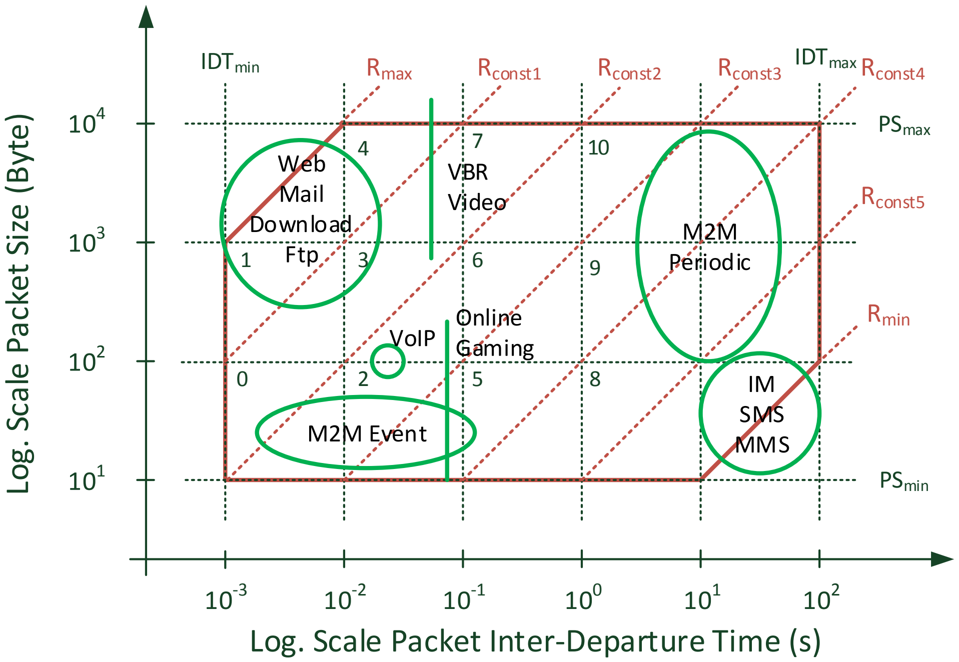


Figure 4.18: *Benchmarking region consists of PS-IDT plane, including popular applications and areas of interest.*

Section 4.6.2). Total number of groups varies from 0 to 4 as shown in Table 4.6. Available resources per group are determined by a round-robin like scheduling operating on the remaining resources (c.f. Section 4.6.2). Note that increasing the number of groups decreases the available resource per group.

In the considered validation scenarios, there exists one eNB and 7 eUEs located in an area of $2000m^2$. CBA allows different allocations for the uplink subframes to adapt to the load condition as well as to the frame configuration, namely TDD or FDD. While for FDD, CBA can be active in all subframes, in TDD, this has to be a subset of the active uplink subframes according to the TDD frame configuration. In the considered experiments, only subframes 2,3,7,8, and 9 are configured for CBA transmission based on the FDD frame structure (c.f. Table 4.6), and the CBA resource allocation only operates on the unallocated resource blocks of the CBA subframe. This gives the highest priority to the regular resource allocation. The MU-MIMO capability are emulated at the eNB by allowing eNB to decode the RNTI of the collided UEs.

Table 4.6 summarizes the system configuration setup. The results presented here can be reproduced using the templates 120-130 corresponding to the area 0-10 in Fig. 4.18. Templates are located at “targetsEXAMPLESOSDWEBXML” in the SVN repository given in [72]. Each experiment is performed at the powerup state to avoid any performance bias among the templates caused by the history of the traffic.

Table 4.6: LTE FDD SYSTEM CONFIGURATION.

Parameter	*Value*	*Parameter*	*Value*
Carrier Freq.	1.9 GHz	*Traffic Type*	VBR & CBR UDP
Bandwidth	5MHz	*Fading*	AWGN Ch.
Frame Duration and type	10ms FDD	*Pathloss at 0 dB*	-100dB
TTI	1 ms	*Pathloss Exp.*	2.67
UEs	7	*Mobility*	Static
CBA groups	1,2,3,4	*Exp. Duration*	60000 LTE frames
Backoff	0, 15, 30, 60, 120	*Transmission mode*	2 (Tx Diversity)
CBA Allocation policy	round robin	*RLC Mode*	UM
CBA subframes allocation	2,3,7,8,9	*RLC reordering timer*	50ms

4.6.2 Applied Grouping Scheme and Scheduling Algorithm

To see the impact of the number of CBA groups, a simple grouping method is applied which consists of UE identity (a positive integer) modulo total number of CBA groups, and can be shown as $(\mathbb{Z}_{NUM\_CBA\_GROUPS}, +)$. As mentioned earlier, the MAC layer will first perform the resource allocation for the regular uplink access followed by the contention-based uplink access, and that the resource allocation for CBA is performed on per-group basis and operates only on the unallocated resources according to the predefined policy, namely round robin. The pseudo code of the CBA top-level group-based scheduling is shown in Algorithm 4.

4.6.3 Results

This section briefly highlights the general latency trend of CBA for sample area 2 and 5. Then, a complete set of results are presented for both uncoordinated and coordinated traffic pattern as explained in Section 4.6.1.

4.6.4 General CBA Latency Trend

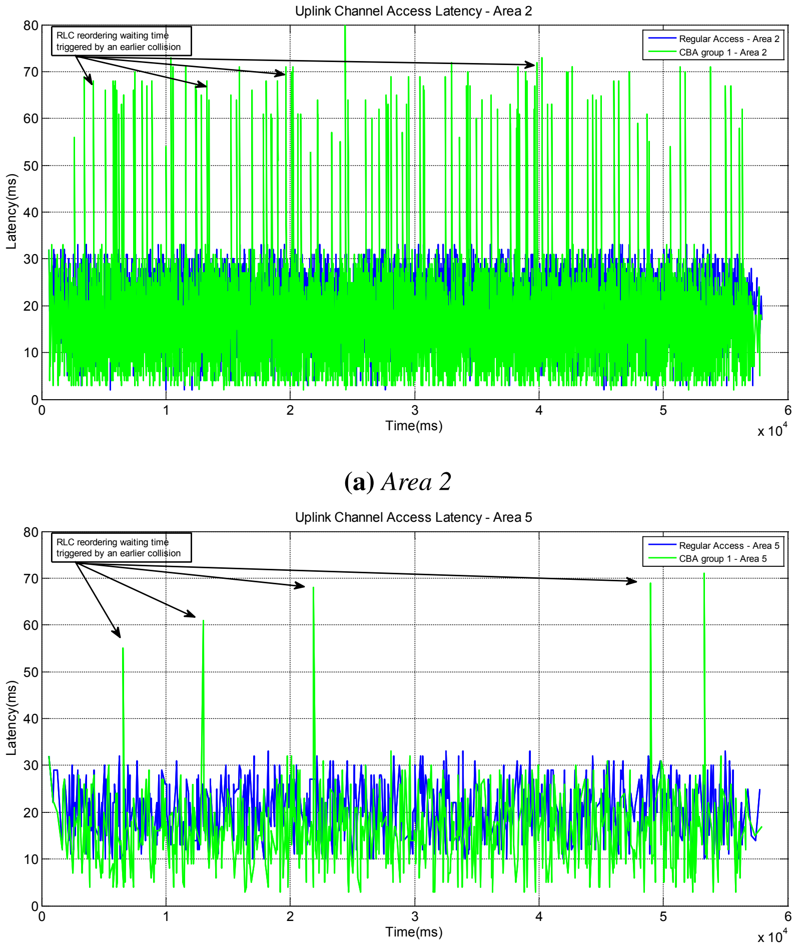
The general latency trend of CBA with 1 active group for sample area 2 and 5 over time is compared with that of regular access, as shown in Fig. 4.19. It can be seen that with CBA the average latency (green curve) is lower than that of regular access (blue curve) but with several latency peaks that produce higher variability. It is found that those peaks corresponds to the RLC reordering timer triggered by an earlier collision caused by CBA. The reordering timer starts assuming that the missing PDU is still being retransmitted in the HARQ protocol and waits until the timer expires (the value of reordering timer depends on the configuration and can vary from 0 to 200ms). When the timer expires, the receiving RLC entity declares the missing RLC PDUs as lost and delivers the reassembled SDUs from the PDUs stored in the reception buffer to PDCP. Therefore, the delay of packets received after the the first missing PDU are proportional to the reordering timer. Similar observations have been reported the impact of reordering timer and in-sequence delivery on the performance of TCP [73, 74].

In practice, the RLC reordering timer is a function of MAC HARQ RTT. The worst-case delay situation happens when PDUs N and $N+1$ are sent in the same TTI, and PDU N is successfully

Algorithm 4: CBA top level group-based scheduling.

```
Input :
        Let $u$ be the user identity $\in UE\_List$.
        Let $C\_RNTI[u]$ be the RNTI for the user $u$.
        Let $g$ be the group identity $\in NUM\_CBA\_GROUPS$.
        Let $CBA\_RNTI[g]$ be the CBA RNTI for the group $g$.
        Let $CBA\_SF$ be the active CBA subframes.
        Let $p$ be the policy $\in CBA\_SCHED\_POLICY$.
        Let $R_{available}$ be the available resource blocks for CBA.
        Let $R_{min}$ be the minimum resource block unit.
        Let $R_{required}[g]$ be the required resource blocks by group $g$.
        Let $UE_{active}[g]$ be the active number of CBA users
        Let $T_{predicted}[g]$ be the predicted traffic pattern for group $g$ in terms of packet size $PS[g]$
and inter-departure time $IDT[g]$ statistics.
Output: CBA allocation for each group.
Result: Generate $n$ CBA DCI0A, where $n \in NUM\_CBA\_GROUPS$.
foreach $SF \in CBA\_SF$ do
    foreach $g \in NUM\_CBA\_GROUPS$ do
        $T_{predicted}[g] = f(PS[g], IDT[g], ...)$;
        foreach $u \in UE\_LIST$ do
            /* $u$ has a valid C-RNTI */
            if $C\_RNTI[u] > 0)$ then
                /* $u$ belongs to the group $g$ */
                if $g$ == $u\%NUM\_CBA\_GROUP$ then
                    $UE_{active}[g] + +$;
                end
            end
        end
        if $R_{available} \geq R_{min}$ then
            $R_{required}[g] = f(R_{available}, R_{min}, UE_{active}[g], p, T_{predicted}[g])$;
            $R_{available} - = R_{required}[g]$;
            $N_{DCI} + +$;
        else
            break;
        end
        Generate CBA DCI for $CBA\_RNTI[g]$;
    end
end
```

received while PDU $N + 1$ is recovered in the last HARQ retransmission round (M). Thus, the reordering timer has to be set to the reception time of the second PDU, that is $(M - 1) \times RTT_{HARQ}$.



(a) *Area 2*

(b) *Area 5*

Figure 4.19: *CBA Latency trend.*

4.6.5 Uncoordinated Traffic Pattern

Fig. 4.20 compares the latency of regular access with and without CBA for different number of groups and a fixed backoff in the considered benchmarking scenario. It can be seen that CBA is able to lower the latency up to 50%, in particular for small to medium size packets (left and middle column). The latency gain here is mainly due to the reduction in the signalling overhead. When the traffic load is high (e.g. area 1, 4, 7, 10), CBA provides a negligible latency gain. This is because the traffic is continuous and regular uplink grants are constantly signalled to the terminal based on the buffer status report. Furthermore, the CBA allocation is performed on the remaining resources after the regular scheduling reducing the total CBA transmission opportunities.

Another observation is that the best latency performance is achieved when the packet inter-departure time is larger than the service time forcing the terminal to request for resource (e.g. sending an SR). This gain is directly related to the signalling overhead of the regular access (c.f. Fig. 4.9). However, the jitter (not shown here) is increased as the CBA delay deviates from the delay obtained through the periodic regular scheduling.

From the figure, it can be seen that increasing the number of groups brings additional latency reduction only for small to medium packet sizes (relative with respect to CBA allocation unite). This is because the amount of available resources per group proportionally decreases as the number of groups increases, which triggers packet fragmentation and reassembly at the RLC layer for the big packets causing extra latency. However, grouping significantly reduces the collision rate by partitioning the resources into a set of non-overlapping blocks within the same subframe (c.f. Fig. 4.24). This has dual benefits, for an eNB to perform one allocation per group in a subframe (each group has a dedicated CBA-RNTI) as multiple allocations for the same group are not supported in LTE release 10 (under consideration for the future releases), and for a UE to randomly select one or more resources across multiple groups (additional degree of freedom). Recall that the applied grouping scheme is based on the UE identity and as a result a UE can not belong to more than one group.

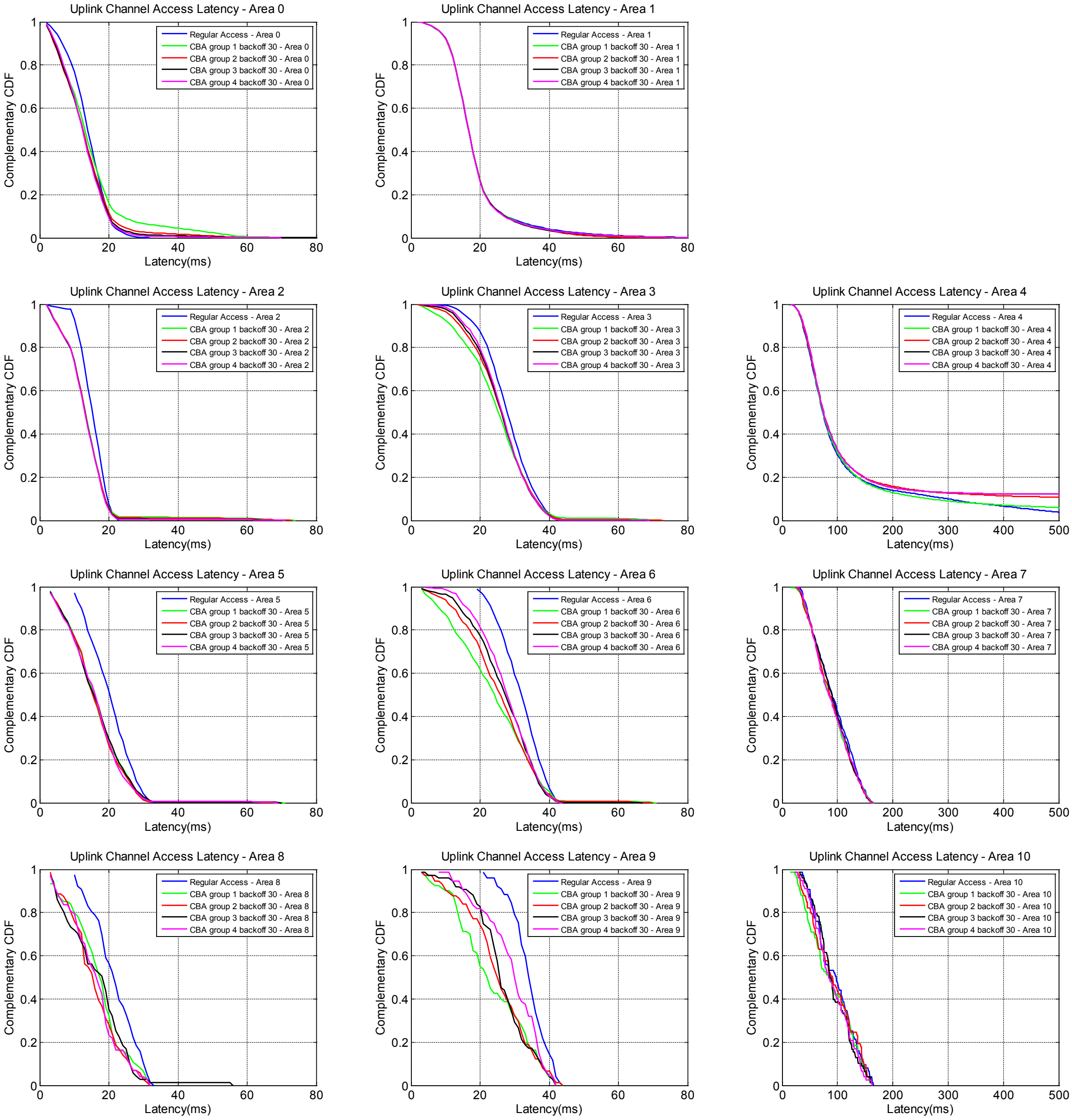


Figure 4.20: *CBA Latency for different number of groups and areas with backoff 30 and uncoordinated traffic pattern. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. Fig. 4.18).*

Fig. 4.21 compares the latency of regular access and CBA with different backoff values and 1 active group for the considered benchmarking scenario. As expected, the latency decreases as the backoff timer decreases (backoff 120, 60, 30, 15). When backoff is zero (i.e. UE transmits with a probability of 0.5 on the CBA resources), a significant latency reduction is achievable. In particular for small and medium size packets (area 2,3,5,6,8,9), average one-way-delay is 5ms. When the traffic load is high (area 1, 4, 7, 10), CBA transmission opportunities decreases (less available resources) and the performance gain declines.

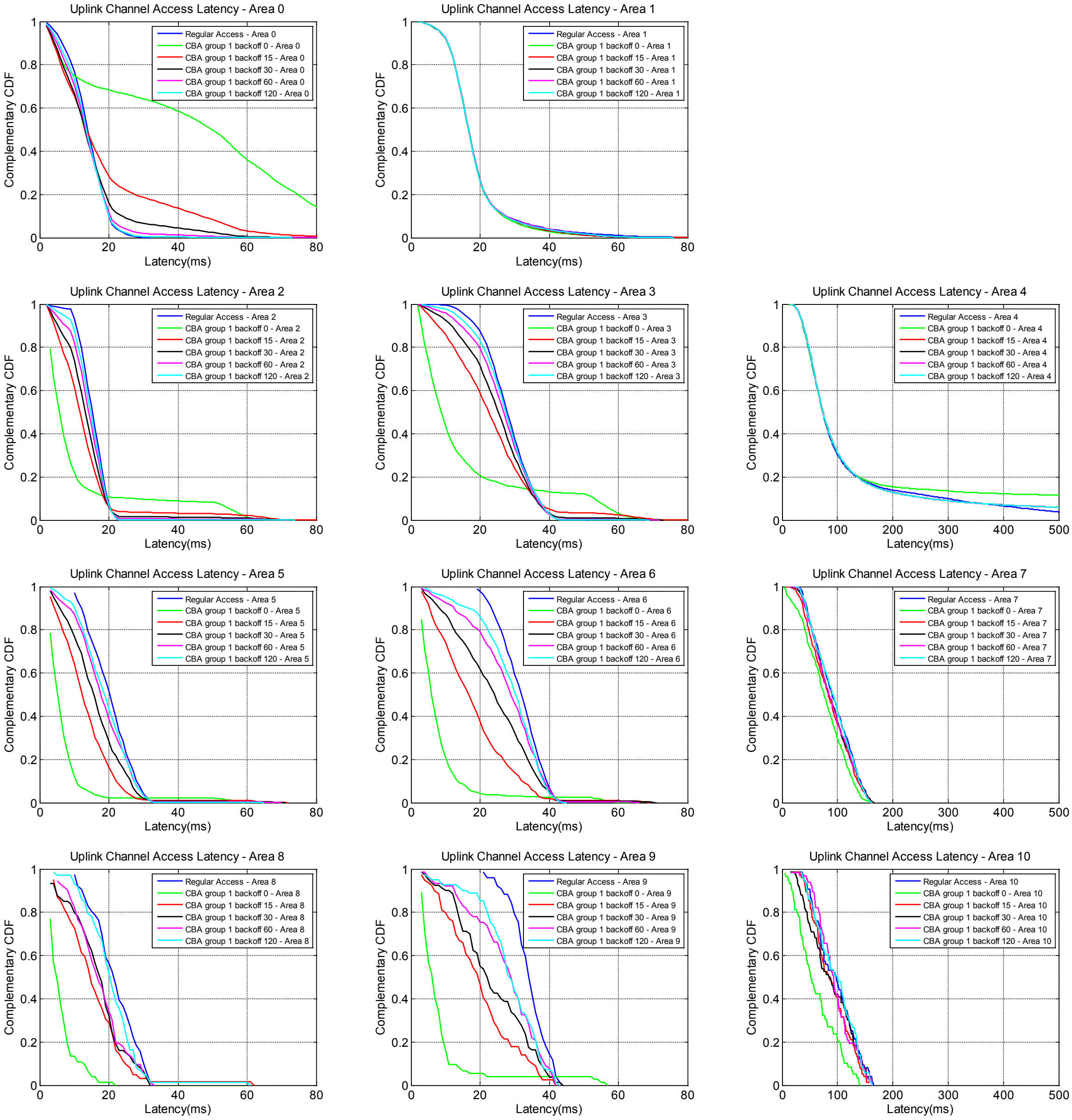


Figure 4.21: *CBA Latency for different backoffs and areas with 1 group and uncoordinated traffic pattern. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. Fig. 4.18).*

Fig. 4.22 shows the achievable CBA goodput for different CBA groups and a fixed backoff in comparison with the regular access. It can be observed that with CBA, the goodput is increased by up to 75%, in particular for small to medium size packets (left and middle columns) as it obtains the smallest latency (c.f. 4.20). For the high traffic load (e.g. area 1, 4, 7, 10), the goodput in all cases are very close. Note that the higher the number of group is, the lower the goodput becomes for area 3, 6, 9. This is because the resource share among the groups decreases as the number of group increases. This highlights the impact of grouping scheme on the performance of resource allocation policy, and calls for a resource allocation policy with the ability to estimate or predict the traffic load across multiple groups. This is particularly important to spread the spatio-temporal traffic load across different group allocations.

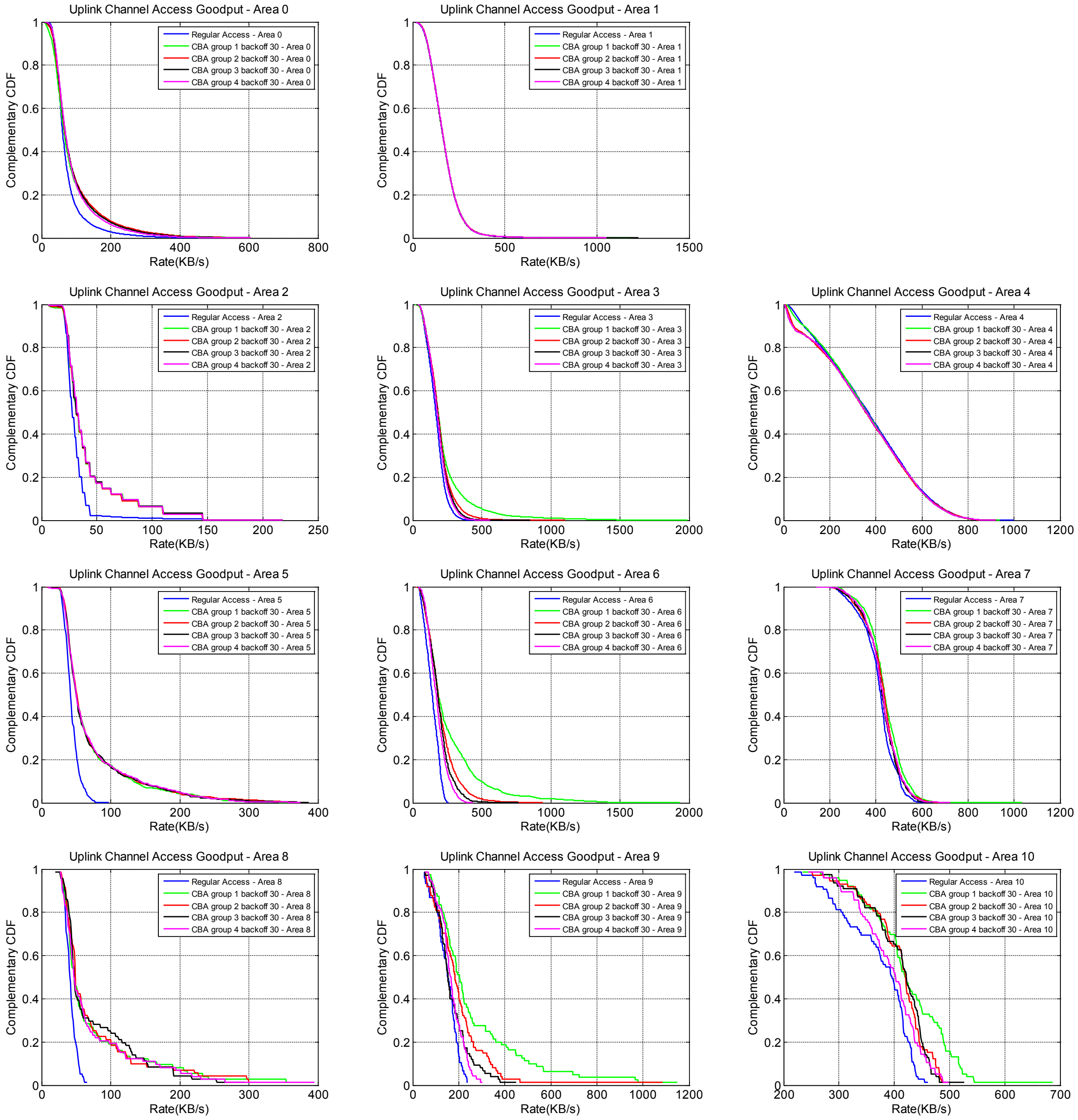
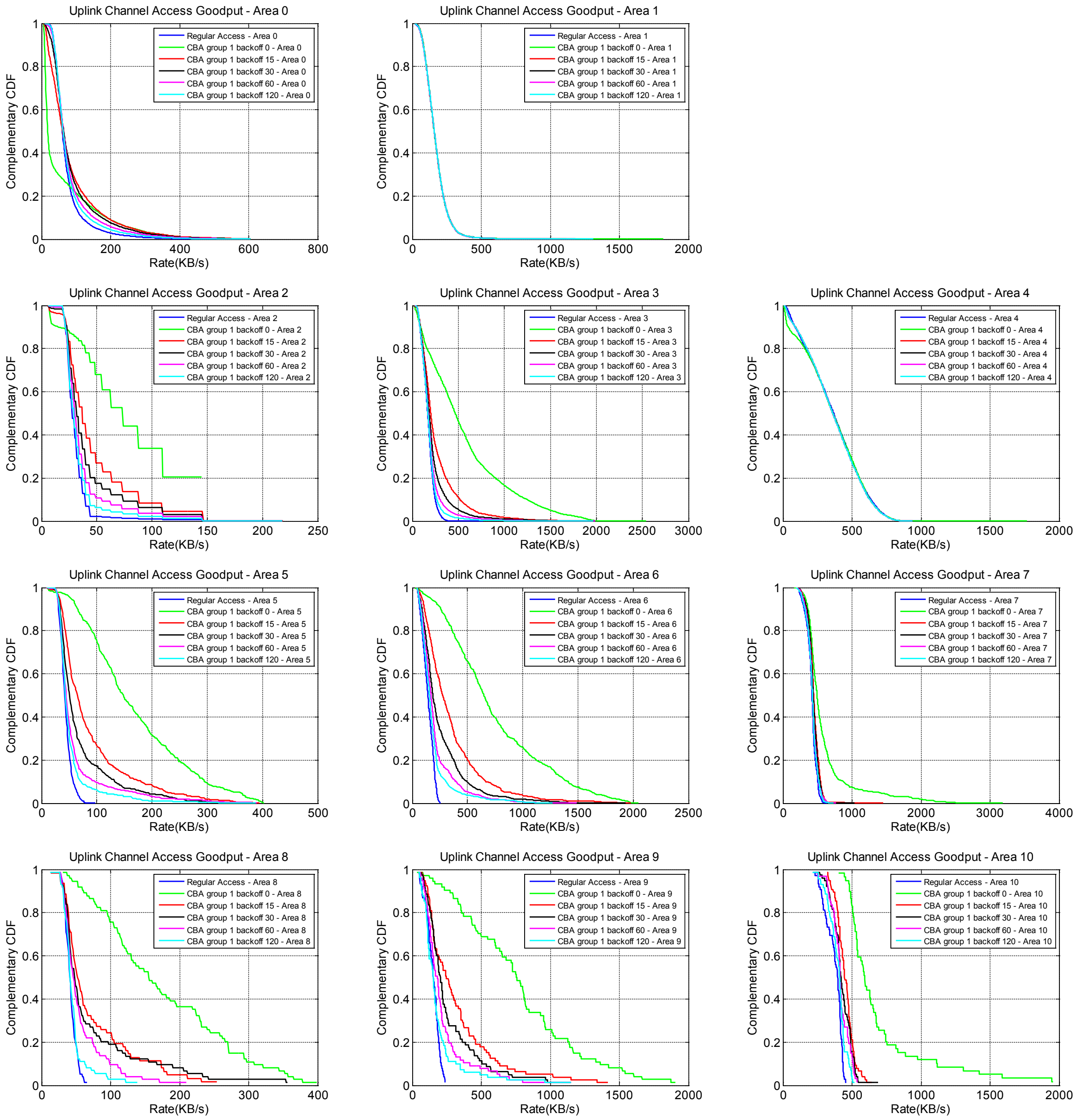


Figure 4.22: *CBA Goodput for different number of groups and areas with backoff 30 and uncoordinated traffic pattern. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. 4.18).*

Fig. 4.23 shows the achievable CBA goodput for different backoffs and 1 active group in comparison with the regular access. It can be seen that the goodput increases as the backoff decreases. When backoff is zero, the goodput is increases by a factor of 3, except for the area 0 (high collision rate, c.f. Fig. 4.24) and area 4, 7, 9 (few CBA transmission opportunities due to high load).



Uplink Channel Access Goodput - Area 0

Uplink Channel Access Goodput - Area 1

Uplink Channel Access Goodput - Area 2

Uplink Channel Access Goodput - Area 3

Uplink Channel Access Goodput - Area 4

Uplink Channel Access Goodput - Area 5

Uplink Channel Access Goodput - Area 6

Uplink Channel Access Goodput - Area 7

Uplink Channel Access Goodput - Area 8

Uplink Channel Access Goodput - Area 9

Regular Access - Area 9
CBA group 1 backoff 0 - Area 9
CBA group 1 backoff 15 - Area 9
CBA group 1 backoff 30 - Area 9
CBA group 1 backoff 60 - Area 9
CBA group 1 backoff 120 - Area 9
Complementary CDF
0
0.2
0.4
0.6
0.8
1
0
500
1000
1500
2000
Rate(KB/s)

Uplink Channel Access Goodput - Area 10

Regular Access - Area 10
CBA group 1 backoff 0 - Area 10
CBA group 1 backoff 15 - Area 10
CBA group 1 backoff 30 - Area 10
CBA group 1 backoff 60 - Area 10
CBA group 1 backoff 120 - Area 10
Complementary CDF
0
0.2
0.4
0.6
0.8
1
0
500
1000
1500
2000
Rate(KB/s)

Figure 4.23: *CBA Goodput for different backoffs and areas with 1 group and uncoordinated traffic pattern. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. Fig. 4.18).*

Fig. 4.24 shows the number of collisions that occurred for each area for different backoffs and number of groups. It can be seen a high number of collisions in area 0, which corresponds to the case where large number of small packets are generated. In such a case on average the packet size is smaller than the transport block, and as a result multiple packets are combined and transmitted within the same transport block. Thus one collision causes multiple packet losses. As expected, increasing the backoff and number of groups increase the traffic temporal distribution and thus decreases the collision.

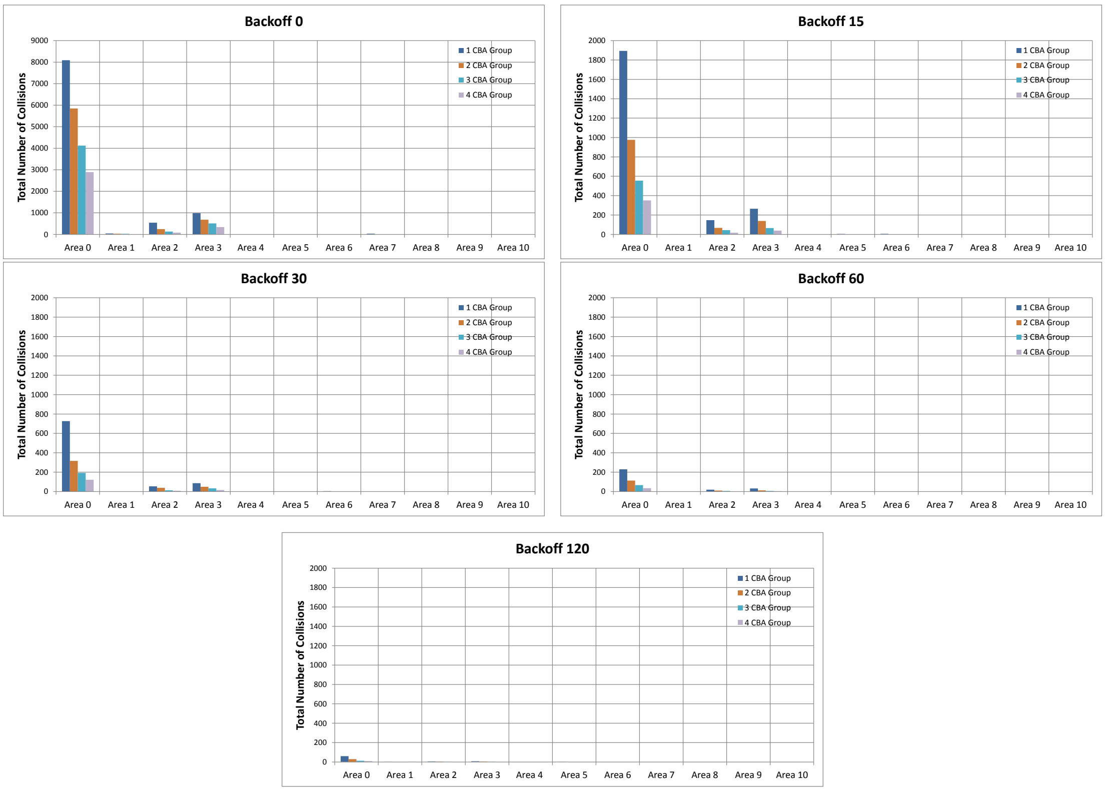


Figure 4.24: *Total number of collision as a function of backoff for all areas (uncoordinated traffic).*

4.6.6 Coordinated Traffic Pattern

Fig. 4.25 and 4.26 shows the latency of CBA with different number of groups and backoff in comparison with the regular access for the considered benchmarking scenario. For high traffic load (area 4, 7, 10), almost all the uplink subframes are used by the regular scheduling leaving few available subframes for the CBA transmission opportunities. For low to medium traffic load, it can be observed that increasing the number of groups contributes to the latency reduction (c.f. Fig. 4.25), while increasing backoff increases the latency for the head distribution and decreases for the tail distribution with the same average latency (c.f. Fig. 4.26). This is an indication for a large number of collisions caused by the coordinated traffic patterns and the resulted waiting time due to the RLC reordering timer. To separate the coordinated traffic in time and frequency when it is generated, multiple mechanisms are required. While grouping partition UEs on per subframe basis and backoff spreads the traffic across subframes, multi-allocation within each group on per subframe basis can further spread UEs inside a group. The observation here also confirms the need for a traffic-aware grouping scheme and resource allocation policy, especially for the coordinated traffic.

Fig. 4.27 shows the number of collisions that occurred for each area for different backoffs and number of groups. We note a larger collision area (0, 1, 2, 3, 4) compared to the uncoordinated traffic. This corresponds to the case where large number of coordinated packet are generated (i.e. small IDT). Small backoff value increases the collision rate as it triggers a large number of CBA transmission attempts. Increasing the number of groups, on the other hand, decreases significantly the collision rate for small backoff value. This shows the interplay between back-

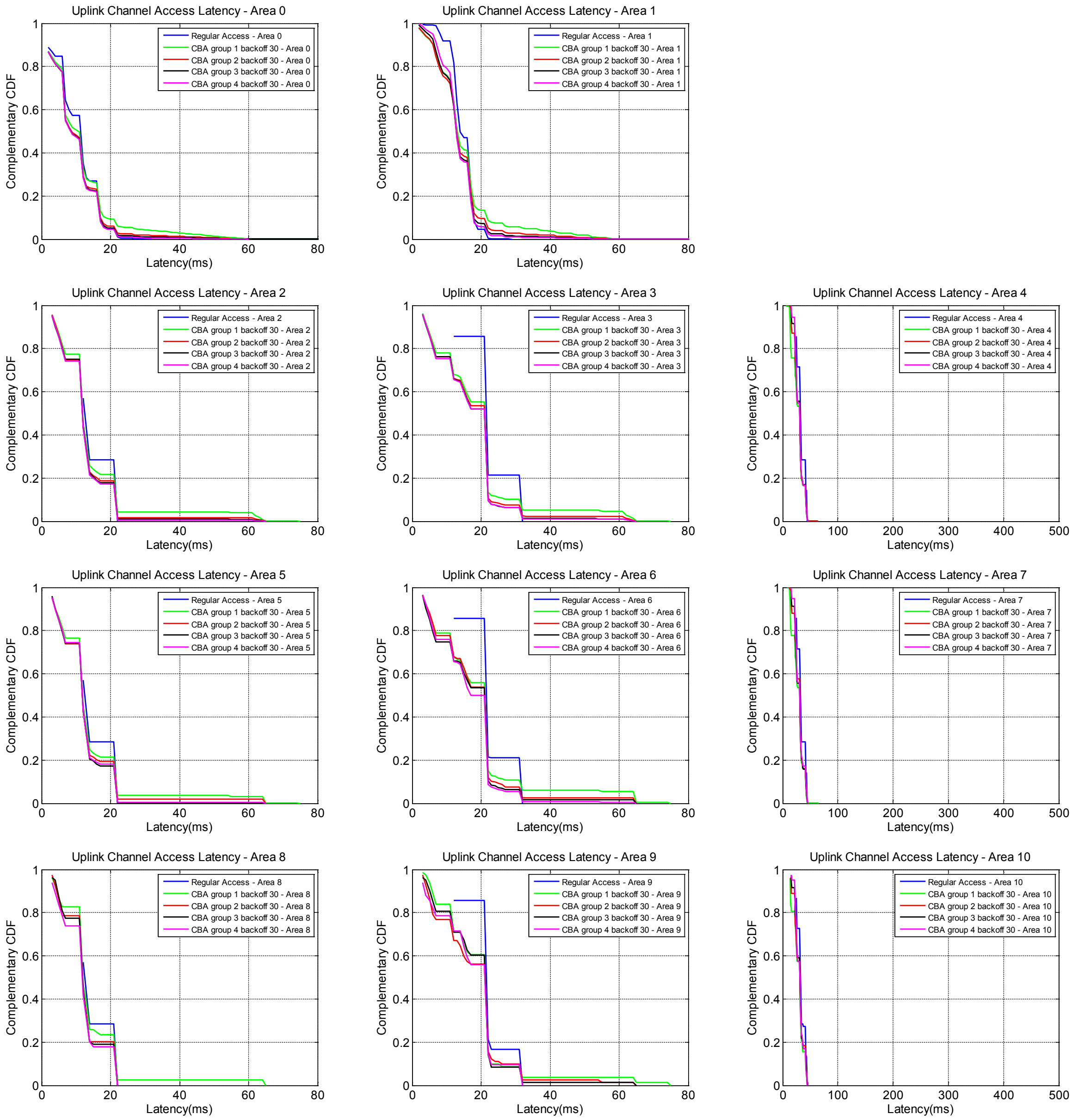


Figure 4.25: *CBA Latency for different number of groups and areas with backoff 30 and coordinated traffic. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. Fig. 4.18).*

off and grouping to distribute the traffic over time and frequency to achieve a trade-off between the delay and collision rate.

4.7 Comparison

Table 4.7 compares the median uplink channel access delay of the emulated LTE with and without CBA method. For CBA, results are taken for the low collision cases, namely backoff 0 and 1 CBA group in case of uncoordinated traffic (c.f. Fig. 4.24, and backoff 30 and 4 CBA groups for area 0 - 4 and backoff 0 and 4 CBA groups for area 5 - 10 in case of coordinated traffic. When examining the results presented above, the following conclusions can be drawn.

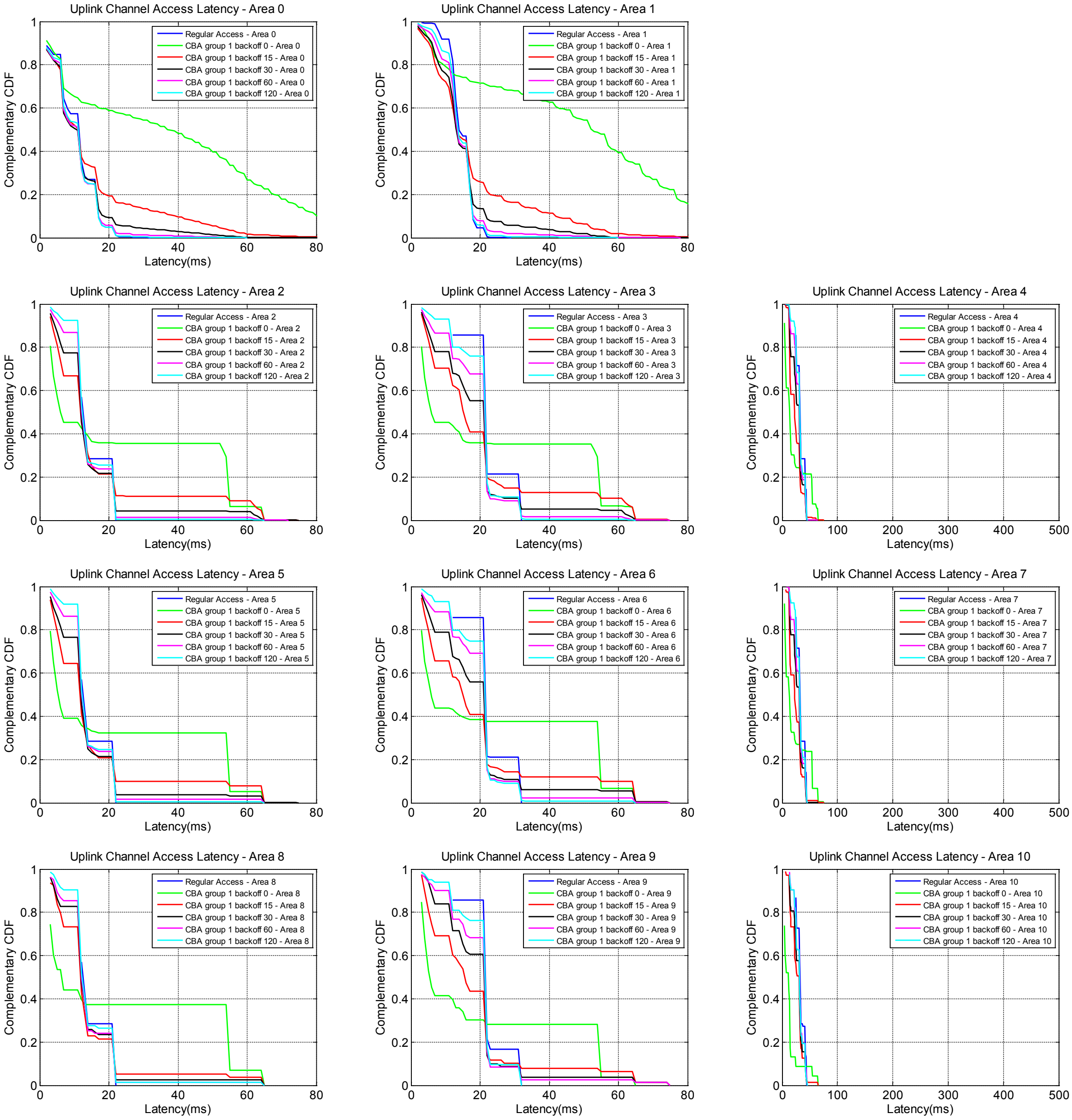


Figure 4.26: *CBA Latency for different backoffs and areas with 1 group and coordinated traffic. Benchmarking region organized according to the packet size: (left) small size packets, (middle) medium size packets, (right) large size packets (c.f. Fig. 4.18).*

- CBA significantly reduces the uplink channel access for small to medium packet sizes relative to the allocated transport block size.
- Backoff and grouping are two key parameters to lower the latency and collision rate simultaneously.
- CBA meets the requirements of online gaming and M2M applications considered in Section 4.1.3.

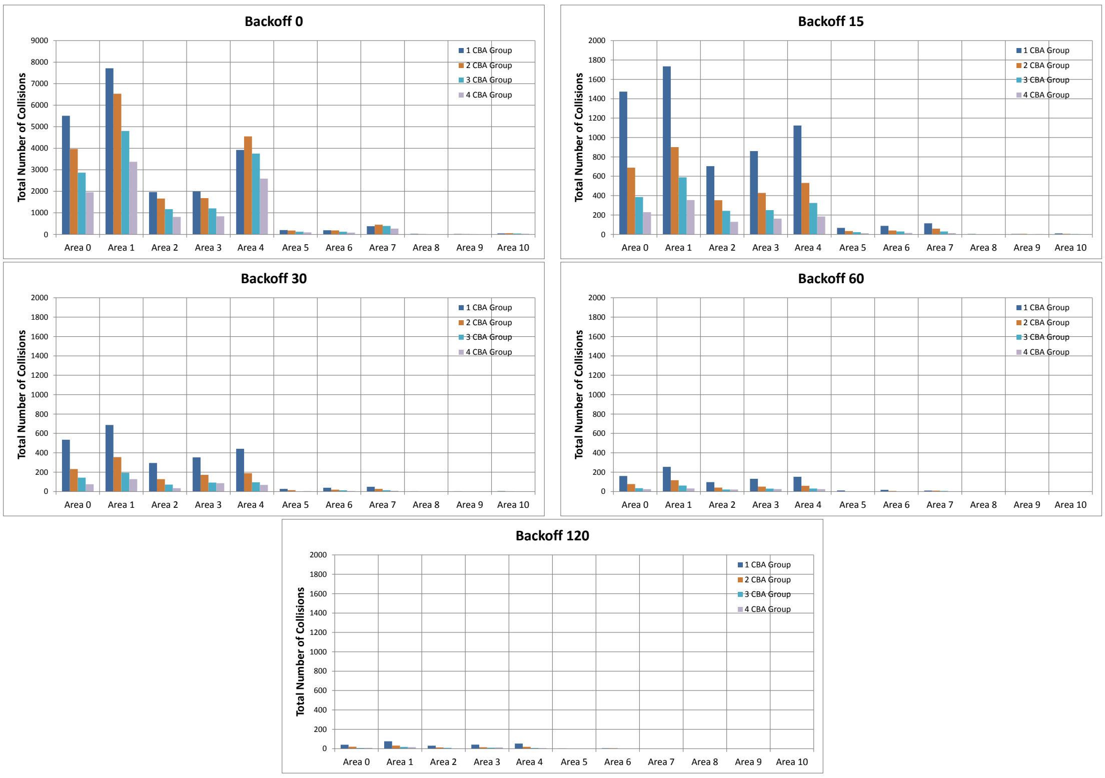


Figure 4.27: *Total number of collision as a function of backoff for all areas (coordinated traffic).*

4.8 Conclusion

To eliminate the signaling overhead of the random access for data transmission, a contention based access (CBA) method is proposed in this chapter. With CBA, UEs select resource randomly without indications from eNB, which saves signaling overhead and hence the latency can be reduced. To address the problem of collision, a control header (C-RNTI) with higher protection combined with MU-MIMO detection at the eNB allows for the identification the collided UEs so as to allocate dedicated resources in subsequent subframes.

Simulation results validate the rationality of the CBA method and demonstrate the achievable performance. They show that:

- CBA outperforms the random access method in the term of latency;
- Latency constraint can be satisfied with the resource allocation method for CBA;
- Coding rate for the control information, the CBA resource unit size, and the number of receiving antenna have strong effect on the performance of CBA.

Furthermore, the required LTE modifications to host CBA methods are presented. The results obtained from the prototype implementation on the top of the OpenAirInterface LTE platform [72] demonstrates that the proposed architecture is feasible in LTE/LTE-A and the latency can be significantly reduced.

Table 4.7: Delay requirement per application type for LTE and LTE-CBA

Area	(PS,IDT)	Uncoordinated Traffic		Coordinated Traffic	
		LTE	LTE-CBA	LTE	LTE-CBA
Area 0	(S,H)	14 ms	50 ms	12 ms	9 ms
Area 1	(M,H)	17 ms	17 ms	14 ms	13 ms
Area 2	(S,M)	16 ms	6 ms	13 ms	12 ms
Area 3	(M,M)	28 ms	9 ms	22 ms	22 ms
Area 4	(L,M)	74 ms	74 ms	33 ms	32 ms
Area 5	(S,L)	21 ms	6 ms	13 ms	4 ms
Area 6	(M,L)	33 ms	6 ms	22 ms	4 ms
Area 7	(L,L)	91 ms	76 ms	33 ms	16 ms
Area 8	(S,XL)	22 ms	5 ms	13 ms	4 ms
Area 9	(M,XL)	35 ms	6 ms	22 ms	4 ms
Area 10	(L,XL)	98 ms	52 ms	33 ms	16 ms

Chapter 5

Evolving UEs for Collaborative Wireless Backhauling

As the definition of 5G technologies is still in progress, the need for solutions that will bring to cellular networks improved capacity, coverage and energy efficiency renders new communication trends for seamless connectivity, heterogeneous networking and interoperability more-and-more attractive [75, 76]. Those trends stipulate a combination of sophisticated techniques that have been in the foreground research promising to be the key enablers for future cellular networks, not least of which are small cell and heterogeneous network deployments, data offloading techniques, tighter 4G/Wi-Fi interworking, advanced interference coordination techniques and spectrum management, coordinated multipoint (CoMP) transmissions, relaying and multiflow techniques across small cells, device-to-device (D2D) and machine-to-machine (M2M) communication. Despite their promising benefits, all the above techniques call for a network architecture that can simultaneously provide lower costs, lower latency, and greater flexibility to offer improved networking, ubiquitous access and extended coverage.

In this context, a disruptive and forward-looking idea is for the future cellular networks to exploit UEs as active network elements to collaboratively convey traffic. Recent studies have shown that users are willing to participate in such collaboration and delay their own traffic or increase their power consumption if, for example, some incentives in terms of reduction of subscription costs are provided [77–79].

This chapter introduces a new paradigm for link virtualization through cooperative L2/MAC information (packet) forwarding enabled by a group of UEs, denoted as evolved UE (eUE). The considered architecture is depicted in Fig. 5.1. Theoretical discussions in that respect about future and upcoming advances for next generation wireless networks more akin to a "*flatter*" network architecture, are quickly developing now [76]. The proposed approach promotes a clear trend to rethink and redesign what is perceived today as wireless end-to-end information transfer. The gist of this approach relies on the exploitation of a new virtual air-interface for next generation radio access network (RAN) systems that extends the classical point-to-point physical links and enables new use-cases, possibly raising the ante to create a new marketplace among users and carriers [77]. To this end, the X2-air interface used to interconnect eNBs is re-established by utilizing intermediate collaborative eUEs to create a virtual air-interface through L2/MAC packet forwarding with the objective of achieving a low latency yet reliable

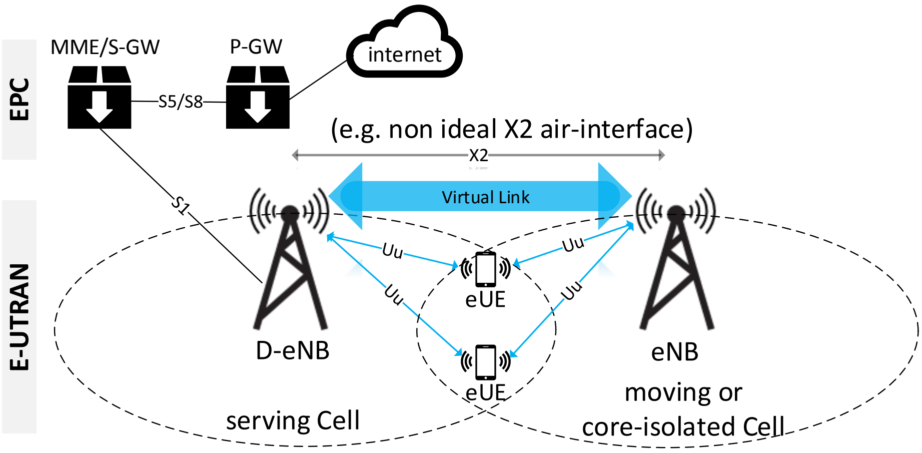


Figure 5.1: *Evolved UEs (eUEs) are leveraged to become active network elements to form a virtual antenna and to interconnect eNBs.*

over-the-air transmissions.

Currently, solutions that are used in CoMP techniques such as Joint Transmission (JT) cannot achieve the maximum performance gain and fail to exhibit the full collaborative potential using the traditional X2-air interface due to low latency and high bandwidth requirements [7, 80]. As dynamic coordination that is used for resources reconfiguration requires very strict time limitation on the update of the control information, the need for an advanced solution that can provide low latency communication for CoMP techniques by forwarding packets at the MAC/L2 becomes essential.

In addition, there is a simmering interest about non-ideal backhaul solutions for CoMP as it is an active area under discussion in the 3GPP for integration within the LTE release 12 framework [81]. A non-ideal wireless backhaul solution requires eUEs to establish multiple connections to different base stations. Such operation allows eUEs to consume a larger portion of radio resources provided by at least two different network points (master and secondary eNB) connected with non-ideal backhaul. The benefits for eUEs that motivate their participation is to experience improved aggregated throughput, mobility robustness especially in case of handover at the cost of higher signaling and power consumption. Additionally, dual connectivity is under study by 3GPP to improve performance of small cell networks [82].

5.1 Design Challenges and Contribution

The integration of the proposed architecture in today's cellular networks raises basic implementation and design challenges that need to be addressed. Next, we outline the most important of them and discuss their solutions.

Challenge 1 - Evolve radio access network - Enable new use cases: Evolving UEs for enabling new use-cases such as moving cells which are required by public safety and intelligent transport systems (ITS) applications is important to future networks. For example as illus-

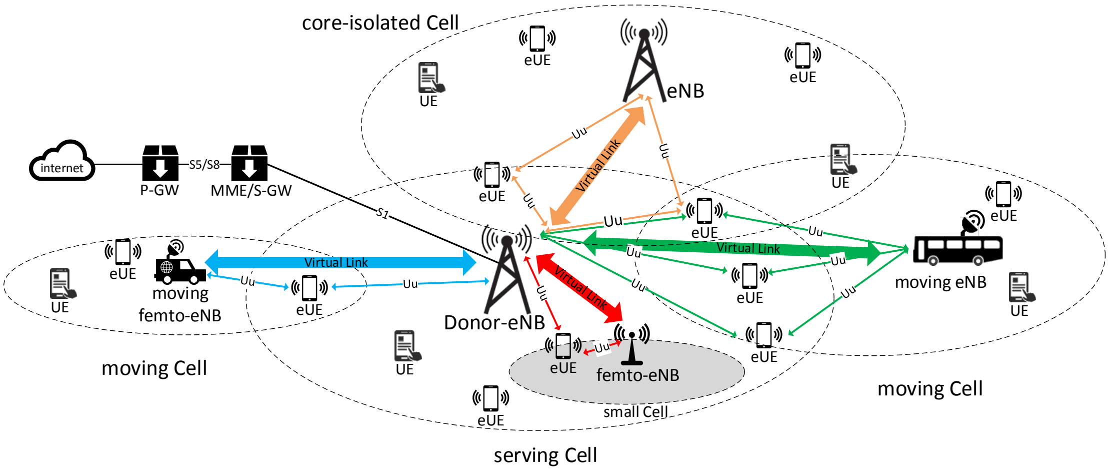


Figure 5.2: *Network Topology: Moving, small and core-isolated cells exploit eUE-enabled multipoint connectivity to access backhaul, when either non-ideal X2-air interfaces cannot be effectively utilized to inter-connect eNBs or wired interfaces are infeasible or too costly to be deployed.*

trated in Fig. 5.2, network coverage extension can be realized rapidly by taking advantage of the diverse and multiple paths that eUEs create in scenarios where network planning cannot be previously contemplated or designed. Moreover, core-isolated cells that miss wired/fiber connection to the core network or experience poor wireless connectivity to another eNB due to a non-ideal X2 air-interface, can be capable of accessing the core through the diverse paths created by users. In parallel multiple eNBs provide alternative paths for routing and service to users.

Challenge 2 - Improve network performance and provide low latency communications using a light-weight architecture: Extending UEs capabilities with smart protocols and advanced RF processing in order to be able to forward packets is essential for wireless mesh networking over the cellular network. Toward this direction, in the proposed architecture UEs are evolved into *on-demand* intermediate data forwarders (called eUEs) that convey traffic among eNBs and form a virtual MIMO antenna. eUEs are enabled as a service by the eNBs [83] for relaying traffic. Relaying involves advanced signal level cooperation to combine transmitting signals as well as sophisticated packet forwarding at L2/MAC of eUEs. Additionally, the challenge of limited capacity can be tackled effectively by enabling multiple associations of an eUE to multiple eNBs in order to benefit from inter-node radio resource aggregation and CoMP reception.

Challenge 3 - Provide incentives to eUEs to admit conveying traffic: On one hand, at a cost of a more dynamic network and resource management, eNBs can leverage eUEs to assist them with cooperative forwarding so as to improve their performance and re-establish non-ideal backhaul. On the other hand, eUEs at the expense of increased battery consumption use multiple data flows through alternative paths for their own benefits in order to increase their throughput by (i) receiving multiple flows and (ii) selecting among multiple interfaces that meet their QoS requirements. Those incentives are given to eUEs to participate and help eNBs to re-establish the X2 air interface. As a consequence, a promising economical business model can be enabled, where operators compensate users for assisting them (see [77–79]).

Challenge 4 - End-to-end Architecture: Leveraging a light radio architecture to enable a cost-effective solution for providing end-to-end services is essential both for users and operators. To this end, a light-weight radio architecture that allows for low-latency access on the core network (C-plane) and low-latency communication transfer (U-plane) is an advocate of enabling end-to-end Quality of Service (QoS) assurance for isolated or moving cell users. Moreover, for operators, underpinning an enriched broadband environment that supports device and end-to-end services intelligence so as to allow for enhanced proximity services and low-latency cooperative communication in L2/MAC layer fosters for an evolved radio access network.

To address the above challenges, a novel architecture which evolves UEs (eUEs) in the context of future cellular networks is proposed. A forward-looking idea introduces a more intelligent user equipment capable of associating with multiple eNBs. Through collaboration, eUEs are utilized to provide reliable communication via over-the-air signal combining and low-latency communication via L2/MAC packet forwarding and as a consequence they become enablers for wireless multi-hop networking among eNBs. The intellectual merit of this concept is considered for certain use cases, as it enables the establishment of a new virtual air-interface that helps operators to improve network coverage and performance and accordingly helps users to benefit from their participation and improve their received throughput at an expense on their battery power consumption. Particularly in three use-cases. In moving cell scenarios, eUEs extend the network coverage area by building virtual links (VL) to re-establish backhaul access to moving and/or core-isolated eNBs and by allowing end-point eNBs to exploit eUE functionalities as a service to improve the network performance (see i.e. [84]). In small and/or densified cell scenarios, network and subsequently eNBs provide multiple connectivity and data pipes to the eUEs through different radio bearers so as to increase their capacity and enable seamless handover between the connected eNBs [85].

5.2 Explore for New Use Cases

5.2.1 Moving Cells

Any network type, ranging from an unplanned (moving or isolated) deployment dedicated to a tactical response system up to a planned (fixed) deployment in (sub)-urban areas, require reliable and resilient communication for provision of services, especially for enabling new use-cases found in moving cell, government and public safety scenarios. There is a strong need to provide direct base station connectivity (preferably wireless) when existing communication is lost, or cannot be established with the existing infrastructure [84, 86]. For that purpose, the X2 interface is introduced to provide means for rapid coordination of radio resources [87]. It is currently used to interconnect neighbouring eNBs in a mesh topology in order to coordinate base stations, assist UEs' handover procedure and provide backhauling access to the neighbouring cells. However in such scenarios, planning of an X2 interface with wireless backhaul access may be infeasible or too costly to be established between moving and/or static cells.

Public Safety

When a major emergency situation such as an earthquake, wildfire, hurricane or warfare strikes communication networks related to civil or military purposes, need to be built rapidly and on-the-fly [86, 88]. That is the case where first responders and military require immediate communications support to save lives, establish relief operations and provide ongoing assistance in affected communities. In such tactical response cases, providing backhaul access to a rapid network deployment and core-isolated cells of communication trucks can be effectively enabled by leveraging the respond commander terminals (UEs) to convey critical control information.

In order to build networking and emergency communication solutions for public safety or crisis support, the current practice considers emergency response vehicles being used to deploy a cellular network that provides IP connectivity to first respond commanders. Furthermore, topological and geographical constraints may prevent responders belonging in different groups from rapid network deployment and instant physical access to the core. The above may incur severe shortages and unreliable communication. Moreover, satellite communications that have been utilized so far for accessing the core, incur increased delay and latency for voice, data and video transmissions.

Therefore, the delivery of mission critical contents has been of focal interest to 3GPP so as to provide robust proximity services and group communication support. Particularly, "User equipment to network relay" is a prominent LTE feature that enables a mobile user to act as a relay for another and provides access to network services when they are not available [89]. Providing backhaul access to a rapid network deployment can be effectively enabled by leveraging the respond commander terminals (UEs) to convey traffic.

Intelligent Transport Systems

In planned deployments for public transport, employing moving relay nodes in vehicles (buses, metro, trains, etc.) is a promising solution to overcome potential degradation issues like shadow fading that cause poor QoS and QoE to end-users [84]. Currently, solutions stemming from heterogeneous and small cell networks, data relaying and offloading methods promise performance improvements and are quite attractive to immerse into future cellular networks [90].

However, what is missing is a light-weight and cost-effective solution for the unplanned deployments. Core-isolated eNBs of moving vehicles often fade away from the macro eNB's coverage range as they move out of the predefined trajectory which ensures communication. By exploiting the potential of eUEs to convey traffic within the network, operators expand their network coverage and provide resilient access to the core for these moving cells. This solution comes also with zero cost for network planning and infrastructure deployment.

5.2.2 Small Cells

In a dense urban area, where large physical obstacles such as buildings create a harsh communication environment, coverage holes may often occur due to volatile ambient conditions, even when network planning had been contemplatively designed. Although the solution of small cells can offer improved capacity and extended coverage to users, an UE may still experience poor performance, mainly at the cell edge or during handover, since it is only served by only one eNB regardless of the number of macro or small base stations in its vicinity. Dual connectivity is currently under study by 3GPP for the small cell enhancement [82]. The benefits of dual or more generally multiple connectivities to different base stations are the improved aggregated throughput and mobility robustness, which come at the cost of higher signaling overhead and power consumption. Such operation allows eUEs to consume radio resources provided by at least two different network points (master and secondary eNB) connected with non-ideal backhaul while in RRC_CONNECTED.

Moreover, as the deployment of small cells becomes denser, the physical distance between terminals belonging to neighboring cells becomes also smaller, restraining proximity service to be applicable only to intra-cell terminals. In such dense communication environments, direct communication through device-to-device (D2D) and locally routed communication are two standard methods proposed by 3GPP as a solution. However, the aforementioned solutions do not provide consistent communication with the core as they do not face the eNB isolation problem. Instead, by restoring the X2 air-interface through eUE's assistance and enabling inter-eNB routing, extensions to support proximity service among neighboring eNBs are possible.

5.3 Architecture

In the last years, a consistent effort has been achieved at 3GPP to integrate part of PMR service constraints into the standard 3GPP LTE cellular system. In late 2011, Working Group (WG) SA1 started building a set of service scenarios and requirements, recorded mainly in TS 22.278, for so-called Proximity Services (or ProSe). From these requirements, SA2 WG started feasibility studies and gathering of possible solutions for system architecture in TR 23.703. In RAN1 a big amount of work was done for the physical layer of the new D2D air interface in TR 36.843, which is an essential feature for PMR. However, work on quickly deployable mesh networks has not yet really started, even if this is an important feature for PMR systems responding to natural disaster scenarios, for instance.

This section describes a light-weight and flexible architecture that employs eUEs to form a virtual MIMO antenna on behalf of eNBs and forward packets at L2/MAC for low latency communication.

5.3.1 LTE Mesh Network Topology

The network topology that we consider in this paper, is a wireless mesh network that is built on the top of LTE. This topology is assumed to be 2-level hierarchical or clustered, where a cluster is defined as the set of nodes which are characterized by one-hop connectivity with the eNB macro base station. Fig. 5.2 illustrates the network topology and the new use-cases introduced by eUE-assisted packet forwarding. In this topology, there are three types of node.

- **eNBs** are the legacy 3GPP eNBs with extended functionalities to support *i*) meshing, *ii*) the coordination of user traffic, *iii*) the management and scheduling of radio resources (i.e. time, frequency, and space) within a cell and *iv*) the routing for intra and inter cell communication. It should also be considered that user traffic is not necessarily passed to the core network through eNBs.
- **UEs** are legacy 3GPP user equipment.

- **eUEs** are actually evolved UEs with enhanced capabilities of associating to multiple eNBs and thus interconnecting adjacent eNBs. They act as 3GPP UE terminals maintaining their initial operation and also act as a slave with respect to the eNBs perspective. As UEs do, they also interpret the scheduling messages coming from eNBs on signaling channels so as to enable traffic routing and forwarding relying on the allocated physical resource blocks RBs. eUEs can be also used to extend the cell serving area and provide backhaul access to core-isolated eNBs. eUEs as intermediate nodes are utilized so as to forward the traffic originating from or destined to eNBs. They belong to the control of the radio access network (RAN) of the bridged eNBs.

5.3.2 Virtual Overlay - Enable Mesh Networking

In a typical deployment of cellular systems, base stations are connected to each other and to the core network using wires (fibre/copper). Although this is a standard method in LTE and LTE-A deployments to access an internet gateway, the proliferation of relays and small cells promises to tackle this problem with additional costs both for the operators and the users. The former need to invest and the latter are called for buying new equipment (e.g. Home-eNBs). While in LTE the X2 air-interface can be utilized for interconnecting eNBs, it has been mainly formed for exchanging control plane information between eNBs for assisting handover procedures and advanced inter-cell interference coordination (ICIC) techniques.

Contrary to the above, the proposed networking approach rethinks the standard way of wireless point-to-point cellular communication. It enables a virtual overlay wireless mesh on the top of cellular topology that is abstracted by the eUEs collaboration. Multiple eUEs can collaboratively participate to form VLs. As shown in Fig. 5.3, a VL is the composition of classic point-to-point physical links (PL) and can be realized in two phases: a broadcast point-to-multipoint (P2MP) phase from (source) eNB to eUEs and a cooperative multi-point-to-point (MP2P) phase from eUEs to (destination) eNB.

Fig. 5.4 illustrates the resulting overlay mesh network in large scale on the top of VLs. The efficiency of such a overlay depends on the network-wide control and coordination of radio resources. Moreover, in order to establish a virtual link by appropriately selecting the subset of eUEs, it should be considered also the level of cooperation they are able or willing to provide to the network according to a service-level-agreement (SLA). The selected eUEs list is then provided to the corresponding eNBs so as to initiate the establishment of a collaborative virtual link [77, 79]. This makes software-defined networking highly applicable to the virtual overlay network for enabling the wireless meshing, which is out of the scope of this chapter.

Specifically, the interaction among the layers that is dynamically enabled by the eUEs requires a novel architecture to suggest a new type of collaborative transmission for cooperation that is realized as a CoMP in uplink where eUEs form a virtual MIMO antenna for transmitting to the destination eNB. Particularly, this architecture implies the PHY layer to present a VL as a link abstraction to the MAC layer with a given probability of packet erasure and subsequently the MAC layer to present a VL as a channel abstraction to the network layer by enabling collaborative bearers that are used for local traffic routing between eNBs and end-to-end services. Three levels of cooperation is considered in the architecture as described below.

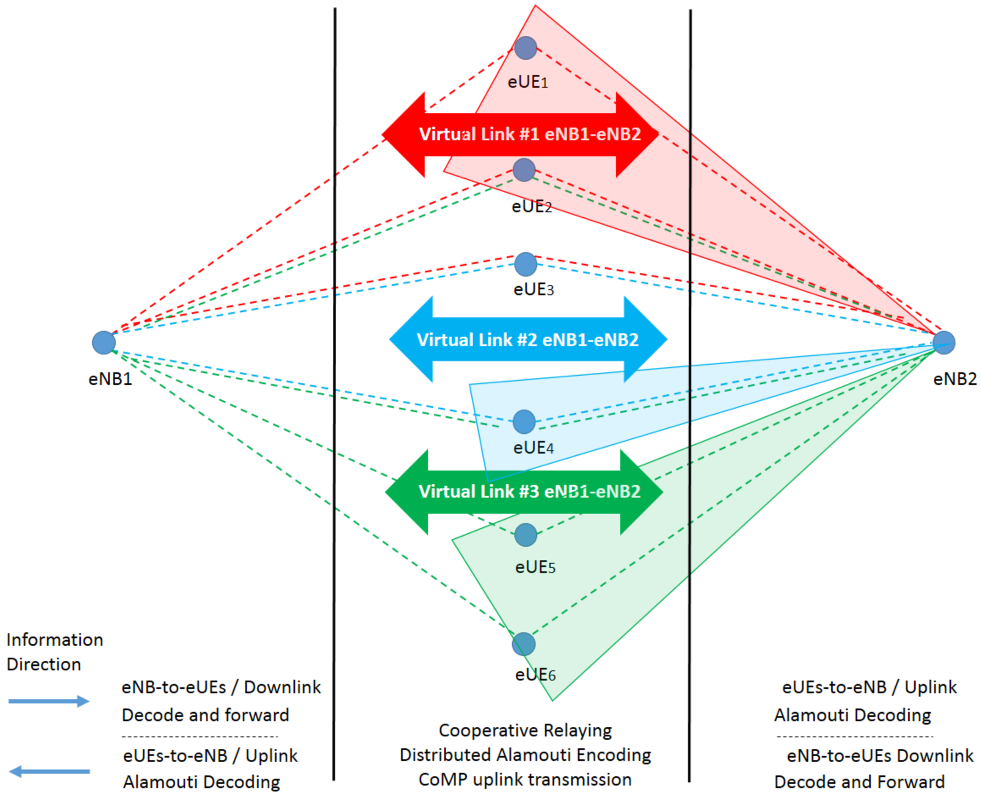


Figure 5.3: *Multiple point-to-point Physical Links (PLs) compose Virtual links (VLs) so as to re-establish the X2 air-interface for interconnecting two isolated-eNBs: first phase broadcast from eNB to UEs, second phase from eUEs to eNB cooperative relaying, third phase cooperative reception from eNB.*

- **Signal-level Cooperation** is operated by the PHY layer, which is responsible for identifying the optimal way to cooperate at the signal-level so that the bit error probability is minimized with respect to predefined quality constraints. Signal-level cooperation presents an interesting abstraction to higher layers: that is, a VL with a given probability of packet erasure. Moreover, cooperation at signal-level implicates all eUEs regardless of the perceived link quality in TX or RX mode with the interconnected eNBs. An appropriate selection of a relaying and coding scheme e.g. Decode-and-Forward (DF) or Quantize-Map-Forward (QMF) and distributed Alamouti codes allows for independent coordination among eUEs and enables an over-the-air signal combination towards the destination eNB [83, 91].

- **Packet-level Cooperation** is operated by the MAC, or more generally Layer 2 (L2), which is responsible for packet-forwarding and scheduling. Specifically, L2 creates a virtual link by leveraging the legacy 3GPP connection establishment procedures in order to complete packet transmissions between two specific end-points. It identifies which physical links (PLs) and their respective end-points need to be activated so that end-to-end frame error rate is minimized, hence drastically improving the efficiency of the signal-level cooperation. The actual decision about VL establishment and PL activation is obtained by the higher layers and L2 from its side identifies and reports this induced

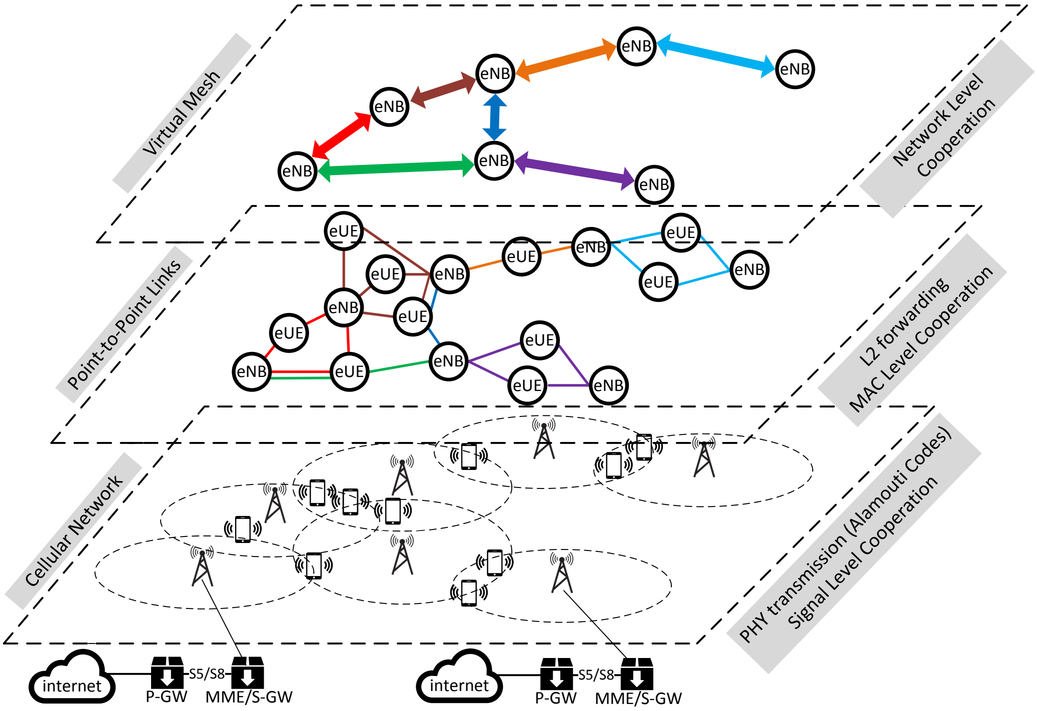

Virtual Mesh
Network Level Cooperation
Point-to-Point Links
L2 forwarding MAC Level Cooperation
Cellular Network
PHY transmission (Alamouti Codes) Signal Level Cooperation
eNB
eUE
internet
P-GW
S5/S8
MME/S-GW

Figure 5.4: *An Overlay Mesh Network of eNBs: Virtual links are composed on the network layer to interconnect eNBs with the aid of eUEs.*

relay selection to the higher layers. In addition to regular scheduling MAC performs scheduling of collaborative broadcast in DL and CoMP transmission in UL[1]. The routing path is optimized as packets do not have to traverse the whole protocol stack and when identified by the MAC they are forwarded for collaborative transmission. Reliable end-to-end packet delivery over a VL may be also handled by L2 through retransmission and/or forwarding-error-correction-codes (FEC), e.g. Hybrid-ARQ (HARQ).

- **Network-level Cooperation** is operated by the network or higher layers, which is responsible for local traffic routing and relay selection (control plane) over a VL. This kind of information is passed to the MAC. Therefore, there is a need to select one or a group of eUEs that will serve as relays to enable signal and packet level cooperation (data plane). Furthermore, the control plane and the data plane are decoupled as the routing decision and relay selection are performed at the higher layers while data forwarding at the MAC/-PHY layer. Therefore, a sophisticated mechanism to support the cooperation by giving access to the forwarding table of the MAC is required and it needs to be enabled. Such a mechanism can be implemented either locally or over the network. In the former case, the MAC/L2 forwarding table can be simply built based on the routing table in a similar way as done in the L2.5/L3 forwarding (e.g as in the multiprotocol label switching (MPLS)). In the latter case, a SDN approach can be applied to interface between the control and data plane.

[1] We assume that the introduced CoMP in UL performed by users (eUEs) considers the distributed Alamouti coding as a general class for an independent yet coordinated transmission scheme.

5.3.3 PHY Layer Design

Consider the scenario where a source eNB transmits to the destination eNB over a VL. The timing of scheduling messages and data transmission adopt the standard procedures in legacy LTE. This end-to-end transmission is realized reciprocally over the two hops that constitute the VL in downlink and in uplink direction. Thus, when the first hop is on downlink the second is on uplink. In Table 5.1, we give the adopted TDD frame configurations for the proposed architecture.

Table 5.1: UPLINK/DOWNLINK SUBFRAME CONFIGURATIONS FOR LTE TDD (TD-LTE).

UL/DL configuration	*DL-to-UL switch point periodicity*	*Subframe number*	
		0 1 2 3 4 5 6 7 8	9
0	5 ms	D S U U U D S U U	U
1	5 ms	D S U U D D S U U	D
2	5 ms	D S U D D D S U D	D
6	5 ms	D S U U U D S U U	D

D,U,S refer to downlink, uplink and special subframe respectively. The S is used for guard time.

The course of actions and associated procedures that occur in the PHY are described in the following.

Cell Search

Search procedure is the primary step to access the LTE network, and consists of a series of synchronization stages to determine time and frequency parameters required for correct timing in uplink and downlink. Standard LTE synchronization procedure allows a terminal to detect the primary and subsequently the secondary synchronization sequences (PSS, SSS) from at most 3 eNBs distinguished by their cell ID group (also known as physical layer identity) representing roots of the Zadoff-Chu sequences [92]. Using this property, the procedures by which an eUE is attached to the network could be activated for non-primary eNBs. The attachment procedure, that an eUE follows so as to associate with an eNB follows the standard 3GPP RRC connection reconfiguration process [60].

Synchronization

For core-isolated eNBs, over-the-air decentralized network synchronization can be utilized by allowing a designated (usually the *Donor* eNB) to provide a time reference synchronization within the network. Then, eUEs will propagate the signal to the core-isolated eNBs through a common synchronization channel. This approach also resolves the interference problem for scenarios with multiple transmitters and one receiver as all the core-isolated eNBs are synchronized with the same reference point as well as the cyclic prefix is able to absorb the differential propagation delay. Regular UEs will follow the standard timing advance procedure controlled by their respective eNBs, while the eUEs will select one of the available timing advance value (e.g. the minimum value or that of communicating eNB). Note that this solution does not require any coordination, and scales smoothly with the number of connected eUEs. However, if the reliability of a unique reference point cannot be assured, due to network mobility or harsh environmental conditions, the designated eNB could be dynamically elected based on parameters of interests, e.g. cell stability. Ultimately, if a common external time reference like a GPS signal is not available the fire-fly synchronization technique could be applied if a fully distributed approach is required [93].

Coding

The PHY layer uses orthogonal frequency division multiple access (OFDMA) in a single frequency mesh network, where all network nodes eNBs, eUEs and UEs share the same resources both in DL and UL. In DL (eNB-to-eUE) a *Decode-and-Forward* DF technique is implemented. Then on the second hop in UL, we apply a distributed Alamouti coding scheme [94] to eUEs to form a virtual MIMO antenna. eUEs belonging on a VL can dynamically participate in the collaborative forwarding a-priori regardless their respective to eNBs link quality. The destination eNB specifies the same time-frequency resources for the framing allocation to the collaborative eUEs by sending them a scheduling grant with an additional information related to the PDUs sequence number, size and HARQ id. Next, each eUE after having correctly decoded (positive CRC check) the requested PDU during the broadcast phase, it performs Alamouti coding independently as an autonomous antenna element and transmits the codes to the destination eNB.

Notice that the selection of the antenna element can be done autonomously by each eUE. In fact, the destination eNB will be able to detect the antenna element from the pilot signal. Moreover, the Alamouti code (distributed or not) is robust to the fading or erasure of the antennas (eUEs). Hence, this technique allows for a flexible number of participant eUEs and does not require that the destination eNB signals to each active eUE the information about the antenna element.[2]

5.3.4 MAC Layer Design

To operate effectively using collaborative packet forwarding at MAC layer and to achieve lower latency communication comparing to the L3 forwarding, apart from the legacy 3GPP procedures, the proposed architecture requires a sophisticated MAC mechanism to manage a VL and perform packet forwarding. Packets are encoded in the source eNB with DF and then are broadcasted to the eUEs, where after successfully received by the eUEs, they are decoded and stored in the eUEs buffer queues maintained at the MAC layer. The reason why the packets are not forwarded directly to the destination eNB is twofold: *i*) In legacy 3GPP LTE, eNBs schedule packet transmissions, therefore eUEs cannot autonomously decide to transmit with-

[2] Of course, many others strategies are possible, depending also on the distributed space-time code implemented. Selection of the best eUE among the ones participating to the VL according to a given metric is possible, as well as the assignment of the precise virtual antenna element role at each selected eUE. However, this more optimal procedure has an increased signalling cost, which is more and more justified as the deployment scenario has less and less propagation variability due to speed and unpredictable interference.

out having received a scheduling grant request by the destination eNB [3]. *ii*) If eUEs perform packet transmissions as soon as they receive them, synchronization and over-the-air signal level combination of the packets cannot be guaranteed at the second hop (eUEs-to-eNB).

The new MAC layer that is designed to enable eUE packet forwarding for collaborative transmission is illustrated in Fig. 5.5 and is composed of five additional functional blocks to handle the VL between two end-points, namely:

- **Queuing:** It handles packet storage in buffers maintained by the MAC layer. When a packet is correctly received by eUEs, it is stored locally at MAC buffers waiting to be scheduled by the destination eNB. The buffer supports indexing mechanisms using AVL trees and hash functions for PDUs storage so as to optimize requests for PDUs that are identified by their sequence number (SN) and their PDU size.
- **Reporting:** It sends periodically the MAC buffer status report (BSR) to the destination eNB indicating which MAC protocol data units (PDUs) have been correctly received and stored.
- **Aggregation:** It is used to concatenate the requested MAC PDUs instructed by the destination eNB.
- **Forwarding:** It identifies whether an incoming PDU on the intermediate eUEs is related to a VL, in which case queuing block will be instructed to store the PDU in a buffer associated with the destination eNB.
- **Co-scheduling:** It schedules the outgoing PDUs on the intermediate eUEs corresponding to a VL requested by the destination eNB.

UE Cell Association and Initialization

eUE initialization follows the same process of a legacy UE performing “attach” to its serving eNB and access to the core is provided by the S-GW and P-GW functionalities. The eUE retrieves configuration parameters from this certain eNB through the control-plane messaging and retrieves also a list of other eNBs to which it is allowed to attach additionally. Then, an addiotional attach procedure is triggered with respect to one of the neighboring eNBs [60]. After the completion of this establishment procedure each eNB initiates the virtual data radio bearer interfaces and the corresponding PDU buffer queues.

Virtual Link Setup

When instructed by the higher layer, a VL establishment procedure is triggered by the source eNB to setup a collaborative radio bearer (CO-RB). Through this procedure, the VL will be mapped to a set of physical links (PLs) from a source eNB to eUEs and from eUEs to a destination eNB. A VL provides an abstraction to the cooperative transmission at the MAC layer.

[3] It should also be clarified here that the eUEs have already notified eNBs through a buffer status report (BSR) about their PDU availability.

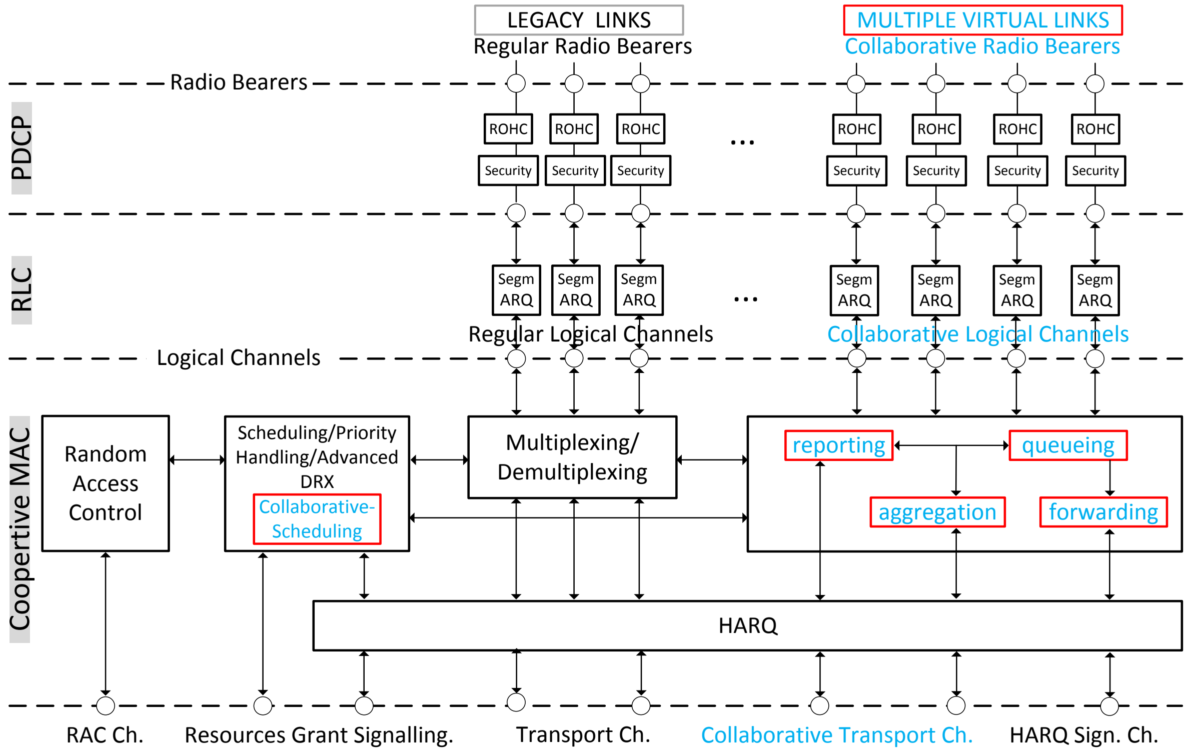


Figure 5.5: *eUE MAC layer architecture and collaborative logical channels: The functional blocks* aggregation, forwarding, reporting *and* queuing *and* co-scheduling *allow for buffer-aided collaborative packet forwarding to interconnect isolated eNBs.*

Thus, the multiple access scheme at the higher layer perceives the lower PHY layer of the protocol stack still as a packet erasure link even though it may be decomposed into several point-to-point links. A VL is used as a means of hiding the information to higher layers: that is, a VL between two points is composed of several point-to-point links formed with the aid of intermediate forwarding eUEs. An eUE can participate at the same time in multiple VLs. (see for example Fig. 5.2 where a PL can be used by multiple VLs and Fig. 5.5, where these VLs are contemplated on the eUE side.) The MAC layer is responsible for managing the virtual/logical links. Particularly, the MAC layer is responsible for identifying the links that will be created in order to complete a single packet transmission between two specific endpoints. Moreover, it is responsible for the identification and scheduling of collaborative transmissions both in downlink and uplink direction.

For that reason, the concept of the Collaborative-RNTI (Radio Network Temporary Identifier) is introduced as an identification number to differentiate a regular transmission from a collaborative one and identify that a certain packet belongs to a certain collaborative transmission via a VL. The CO-RNTI is carried as a part of the MAC header of the control packets that are transmitted from an eNB to eUE in order to establish the VL. A collaborative transmission over a VL requires at least one eUE acting as packet forwarder and two CO-RNTIs that describe the point-to-point transmission on the (eNB-eUE-eNB) physical links. Two CO-RNTIs (an ingress and an egress) can participate to form a VL setup. The ingress CO-RNTI is used by the source eNB to perform a collaborative broadcast and allow the eUEs to store the received data in the destination buffers associated with the egress CO-RNTI. The destination eNB will

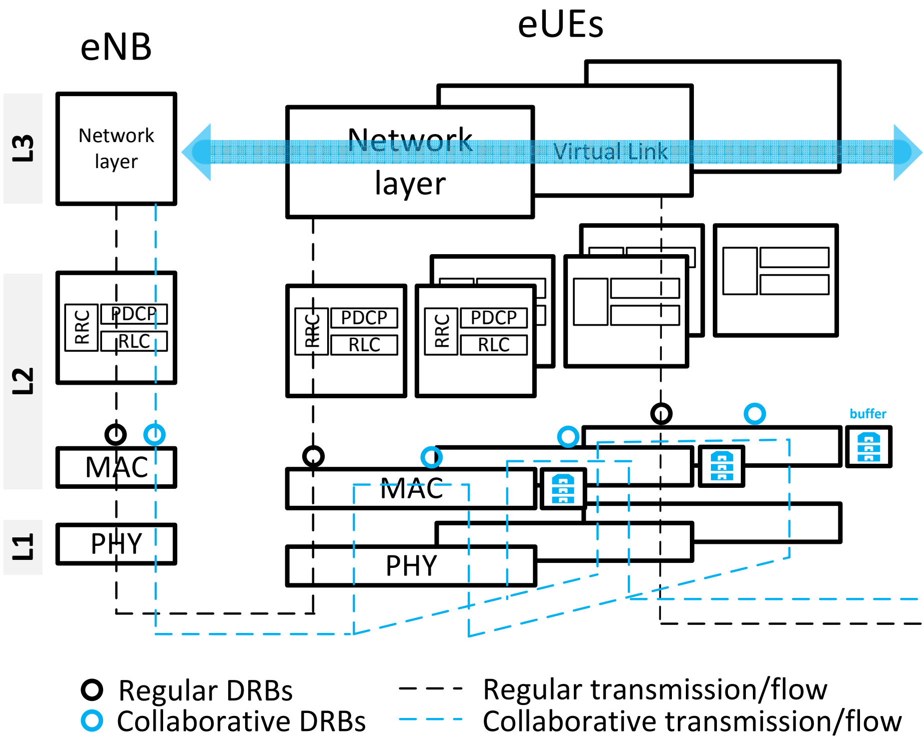


Figure 5.6: *Collaborative Transmission over a virtual link. eUEs adopt a dual protocol stack so as to be able to associate with two eNBs and perform efficiently L2 packet forwarding.*

then schedule a collaborative transmission on this CO-RNTI based on the previously reported collaborative buffer status report (CO-BSR). From the perspective of the destination eNB that needs to communicate back to the source eNB over a collaborative transmission, this design is symmetric and the ingress CO-RNTI (which is the egress CO-RNTI of the source eNB) is used to perform the transmission from the destination eNB in this point-to-point link towards source eNB via an eUE. Therefore, the least number of CO-RNTIs at a given eUE that is essential to compose a bidirectional VL is two.

Channel Estimation

The feedback - channel state information (CSI) that is reported back to the eNBs is of paramount importance, as the availability of accurate channel estimation and traffic information helps to either maximize the data rate and/or achieved throughput or minimize latency consumption during transmissions. The above introduces a tradeoff that is attained at the expense of more induced overhead. Channel estimation can be performed by exploiting specific purpose reference signals that do not carry any data.

Virtual Link Hybrid Automatic Repeat Request (VL- HARQ) strategy

HARQ strategy over a VL with multiple eUEs is not trivial, since the eUEs cooperate to send the same information but are physically separated. This fact creates, for example, possible loss of coherence inside the eUE HARQ buffers which must be dealt with. The HARQ strategy, proposed here, tends to minimize latency and resource use while being robust [91]. During the broadcast phase, the source eNB keeps sending redundancy versions (RV) of the packet with the ingress CO-RNTI, until all the eUEs have correctly detected the packet. A new ACK/NACK message (in fact the one for Scheduling Requests) is used: the absence of this signal is a ACK, while the presence of it is a NACK. Notice that the source eNB will see a NACK as long as one eUE has not correctly received the packet (the identity of the eUEs having sent a NACK is not known to the eNB).

In order to reduce latency, as soon as one of the eUEs correctly decodes the MAC PDU, it can send a CO-BSR to the destination eNB. If the destination eNB decides to schedule the MAC PDU on the egress CO-RNTI, the scheduling information will be received by all the eUEs, even to those not having correctly decoded the MAC PDU yet. Then, all the eUEs create a (virtual) HARQ process associated to the sequence number (SN) of the MAC PDU contained in the scheduling information (i.e. DCI). Hence, the eUEs having the MAC PDU with the requested SN in their queue will send the MAC PDU to the (virtual) HARQ process, for transmission. On the other hand, the eUEs not having yet the MAC PDU with the requested SN in their queues do not send anything. Nevertheless, they maintain the HARQ process as if they have sent the requested MAC PDU. In fact, the virtual HARQ processes in the eUEs are different instances of a unique HARQ process shared among the group of eUEs participating to the VL and the destination eNB. These processes are all synchronized and follow the timing and procedures described in the standard. This mechanism allows the insertion of the eUEs which have decoded with a delay the packet from the broadcast phase, in the HARQ transmissions of the second hop, without any additional signaling cost. In this way robustness in increased.

Upon reception of an ACK from the destination eNB, the virtual HARQ process are released, even by the eUE which has never succeeded in receiving the packet from the source eNB. However, if not all the eUE have successfully decoded the packet in the broadcast phase, the source eNB is still sending redundancy versions to the eUEs thus wasting resources. In order to reduce this wastage an explicit ACK/NACK is sent over the same resources of the implicit ACK/NACK as soon as the eUEs receive an ACK from the destination eNB. The adopted strategy is selected to ensure robustness comparing to other HARQ strategies that maximize throughput and sacrifice robustness.

Link Adaptation, Adaptive Modulation and Coding (AMC) - Coding Rate

In LTE, Adaptive Modulation and Coding (AMC) Scheme is performed according to the CQI values that UEs report back to the eNBs so as to support the highest Modulation and Coding Scheme (MCS) that can effectively decode packets with a BLER probability not exceeding 10% [74]. For a given MCS an appropriate code rate is chosen relying on the Table 7.2.3.1 of 3GPP TS36.213. Therefore, link adaptation matches the transmission parameters of MCS and coding rate to the channel conditions. It should be clarified here that UEs in LTE, and hence eUEs are not permitted to deliberately decide about an autonomous MCS and coding rate selection. This is a control information that is instructed by the eNBs so as to optimally control and configure transmissions within the cell. Moreover, all the resource blocks within a subframe that are allocated to a certain user should use the same MCS. A key issue in the design of AMC policy in the two-hop topology interconnecting two eNBs is whether the MCS assigned to a specific eUE for a collaborative transmission should be the same over the two hops or different exploiting the intermediate buffer storage at the eUEs. In the 1st case, the source

Algorithm 5: A MAC Layer CO-scheduler of an eNB for PDUs collaborative transmission.

```
Input  : $u \in \mathcal{U}$ of selected eUEs and $V \in \mathcal{V} = \{u \in U$ belonging to the Virtual links$\}$.
Output: Collaborative PDUs transmission over VLs enabled by eUEs.
Data: Request $N$ PDUs
Result: Grant resources for $u \in \mathcal{U}$ eUEs.
foreach TTI $t$ do
    foreach $V \in \mathcal{V}$ do
        foreach $u \in \mathcal{U} \bigcap V$ do
            Receive a BSR for $N$ PDUs identified by their SN, size and HARQ id.;
        end
        if $\mathcal{U}' \subseteq \mathcal{U} \bigcap V$ respond with a positive BSR for $N' \leq N$ PDUs then
            foreach $u \in \mathcal{U}'$ do
                Destination eNB grants resources for scheduling $u$ eUE to transmit $N'$ PDUs.;
                and acknowledges PDU reception/failure to HARQ for transmitted PDUs in
                $t - 1$ TTI.;
            end
        else
            Notify HARQ to manage a reschedule of $|N - N'|$ PDUs.;
        end
    end
    foreach $u \in \mathcal{U}$ do
        Provide the Channel State and CQI reports to the higher layers for the PL between $u$ and
        eNB.;
    end
end
```

eNB uses that MCS that captures a representive CQI (e.g. it can be dynamically selected using metrics i.e. average or worst over the two consecutive physical links) for the eUE configuration so as to minimize packet drops and sustain adequate end-to-end communication quality and reliability. In the 2nd case, each interconnected eNB can opportunistically use a different MCS for the transmissions with the bridging eUE relying on the fact that packets are temporarily stored in the buffer queues in order to be transmitted with the best possible MCS over each physical link. This feature is enabled by the packet aggregation service at the MAC layer (see Section IV-D) which concatenates multiple packets together which are being identified by their sequence numbers to fill the allocated transport blocks. This is further explained in the following paragraph.

eNB MAC CO-scheduler

In LTE cellular networks, packet scheduling decisions are orchestrated by eNBs. Therefore, eNBs are responsible to decide which packets should be transmitted by requesting a buffer status report (BSR) from the collaborating eUEs. A source eNB schedules the broadcast transmission in dowlink, while the destination eNB schedules a CoMP transmission in uplink. Until now, eNB schedulers have aimed either at the sole optimization of a performance metric (i.e max. throughput, or min. delay) or aimed at attaining desired trade-offs for achieving a balanced compromise between different competing interests (i.e. Proportional Fairness or Min.

Power vs. Delay). To effectively leverage eUEs for benefiting from a collaborative transmission at the MAC layer, we advance the eNB scheduler - apart from applying a specific policy - so as to be able to identify the common packets that are stored in eUEs' buffers and are identified by their sequence number (SN) and PDU size.

The introduced eNB CO-scheduler that is presented in Algorithm 1 is able to select the eUEs that are currently participating to a VL and grant them resources for scheduling if they reply with a positive BSR for a requested packet. However, the association of the eUEs with a certain VL is based on higher layer network policies and network-level cooperation instructions.[4] The selected eUEs are leveraged to store incoming packets and convey traffic when this would be instructed to them by the destination eNB.

eUEs with bad link qualities that cannot support a predefined CQI C_{th} to sustain a certain MCS M_{th} and coding rate, are still able to contribute to the signal by transmitting a common packet by exploiting Alamouti coding and CoMP techniques. However, those eUEs should be notified by the eNB scheduler that the packet has (or not) already been received so as to update appropriately their buffer status.

Flexible eUE Protocol Stack

eUEs requires in L2 (RRC, RLC and PDCP sub-layers) a multiple-stack protocol in C-plane and D-plane. This allows for eUEs to associate and communicate in parallel with multiple different eNBs and handle simultaneously regular and collaborative transmissions. Fig. 5.6 illustrates the protocol stack of this mechanism that enables collaborative packet forwarding at L2 and multiple DRB reception. The goal is to prevent packets that belong to a collaborative transmission from passing through the whole protocol stack aiming to reduce latency. At L1 a source eNB broadcasts packets to collaborative eUEs. If these packets are correctly received by the eUEs and belong to a Collaborative Data Radio Bearer, the L2/MAC of eEUs identifies their CO-RNTI and stores them temporarily in buffers. Then a collaborative transmission in uplink is scheduled by the destination eNB so as to activate eUEs to transmit the requested PDUs identified by their sequence numbers.[5] L2 transmission presents an abstraction to the L3 layer where the VL is established by hiding the point-to-point physical transmissions.

5.4 Performance Evaluation

In this Section, we demonstrate the performance evaluation and the validation of the rationality of the proposed architecture. We first present the obtained gains considering packet level forwarding when interconnecting two eNBs for various number of employed eUEs, in terms of throughput, latency and packet loss rate. Moreover, we demonstrate eUEs' benefits that stem from multiple connectivity to eNBs and from the exploitation of diverse radio data paths in terms of received throughput. Required protocol operations across all the layers are validated in appendix B.

[4] The eNB cannot associate eUEs to VLs autonomously, relying only on PHY layer information.

[5] The collaborative transmission realizes a distributed Alamouti coding scheme perceived as CoMP transmission.

To conduct experimentation, we leveraged OpenAirInterface (OAI) in order to evaluate the performance of the collaborative forwarding in a practical setting, the distributed synchronization procedures and the 3GPP protocol operations for eNBs and eUEs (full implementation code is available online [95]).

The results presented here can be regenerated using the template 23 and 24 located at "targetsEXAMPLESOSDWEBXML" in the SVN repository given in [95].

5.4.1 Experimentation

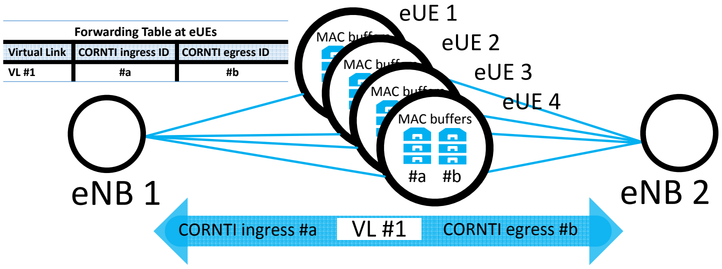
Topology Description

In the considered validation scenario, there exist two eNBs and four eUEs located in an area of $500m^2$. Table 5.2 summarizes the system configuration setup. A 5MHz channel bandwidth (25 RB) is used where the maximum data rate of the collaborative link (UL) is 12 Mbps. It has to be mentioned that the capacity of this collaborative link is dominated by the capacity of the UL, which depends on the channel quality (CQI) of the involved eUEs that is mapped to an MCS. The highest MCS in UL for a 5 MHz channel is 12 Mbps (corresponding to a 16QAM). To analyze the behaviour of the proposed method, no regular uplink traffic is considered in the scenario. However, in practice, the available resources (RBs) are shared between the collaborative link of eUEs and the regular link of UEs/eUEs.

Table 5.2: LTE-A TDD System Configuration.

Parameter	*Value*	*Parameter*	*Value*
Carrier Freq.	1.9 GHz	*Traffic Type*	VBR UDP
PRB	25	*Channel*	AWGN
Frame Type	10ms TDD Config 1	*Pathloss 0 dB*	-100dB
TTI	1 ms	*Pathloss Exp.*	2.67
UEs	$1, 2, 3, 4$	*Mobility*	STATIC
DL BLER	$0 - 0.2$	*UL BLER*	0.18
Exp. duration	6000ms	*RLC Mode*	UM

Fig. 5.7 illustrates the logical topology overview.



Forwarding Table at eUEs

Virtual Link	CORNTI ingress ID	CORNTI egress ID
VL #1	#a	#b

Figure 5.7: *Logical topology for the performance evaluation scenario: A VL is setup to interconnect two eNBs with the aid of 4 eUEs. Each eUE maintains a forwarding table of CO-RNTIs so as to identify the VL and the respective MAC buffers.*

Efficient L2/MAC forwarding

The MAC layer performance is measured in terms of latency, packet loss rate and throughput for different number of UEs=$\{1, 2, 3, 4\}$ and for different BLER probabilities for the backhaul link ($1st$ hop: DL source eNB-to-eUEs) and for a bad channel configuration on the $2nd$ hop UL (eUEs-to dest eNB) characterized by a BLER probability equals to 0.18. The above setup captures a harsh scenario where eUEs assistance is validated. The traffic pattern is defined by a fixed packet inter-arrival time of 20ms and a uniformly distributed packet size from 512 to 1408 bytes.

Fig. 5.8 illustrates the obtained results for the above scenario and demonstrates clearly the eUEs contribution. As the number of employed eUEs increases, the latency and packet loss rate reduces while there is an improvement on end-to-end throughput performance. For the sake of comparison 3GPP, latency requirements for QoS Class Identifiers QCIs 1 and 4 that characterize two guaranteed bit rate (GBR) bearer types for VoIP call and Video streaming are set to 100ms and 300ms respectively [74]. Using 4 collaborative eUEs the measured latency is constantly below 60ms for all BLER probabilities, thus achieving low latency.

Collaborative Performance Rationale

The analysis explicitly focus on the achieved latency and packet loss rate of the collaborative link rather than the data rate as this is very critical for the control plane information exchanged between the eNBs. An important finding is that as the number of eUEs increases the respective periodicity that the eNB receives the PDUs from the collaborative MAC actually decreases, thus reducing drastically the communication latency.[6] Indicatively, experimentation results reveal a significant reduction in latency (up to 16.94%) and improvement on packet loss rate (up to 59.25%) for BLER equals to 18 % on the first and second hop (see Fig. 5.8(a) and (b)). Moreover, for the considered traffic load, we observe a significant gain (up to 68.49 %) on the achievable throughput (see Fig. 5.8.(c)).

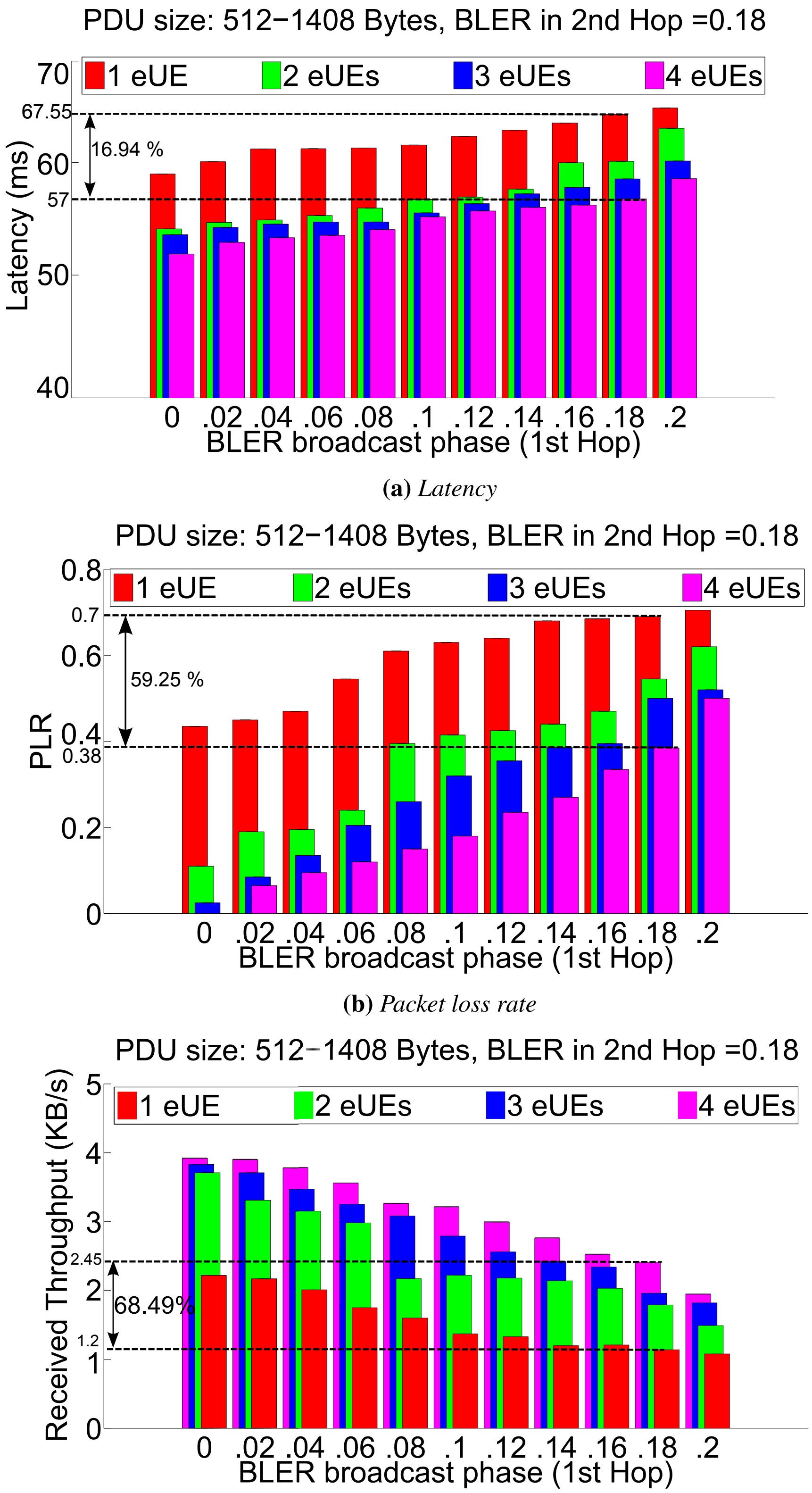
Impact of Queuing Storage:

Each eUE maintains for each VL two MAC buffers for the corresponding ingress and egress CO-RNTIs (see Fig. 5.7). Those buffers are utilized reciprocally in both directions to store the incoming PDUs identified by their ingress and egress CO-RNTIs. The absence of the buffers would cause all the PDUs to be lost as it would be impossible to be forwarded directly to the destination eNB without scheduling. In our experimentation we used a maximum buffer size equals to 100 PDUs. As the buffer storage capacity increases, the PLR is expected to be reduced. However, this comes at a cost of increased overhead and storage for the MAC layer that needs to be attained. Another benefit from maintaining buffers is that they used to store the PDUs until their reception will be acknowledged. As the BLER increases, the PLR grows slightly constant (see Fig. 5.8(b)) as buffers aid in robust transmission and packet recovery.

Benefit of Signal Level Cooperation in Throughput:

The actual throughput benefit that is attained by the destination eNB (see Fig. 5.8(c)) is due to signal-level cooperation. The more the number of collaborating eUEs, the over the air signal combining allows the destination eNB to increase its received throughput (up to $\sim$60% using 4

[6] Recall the four way handshake that needs to be acknowledged between the eNB and each eUE for performing the scheduling request and the scheduling grant.



(a) *Latency*

(b) *Packet loss rate*

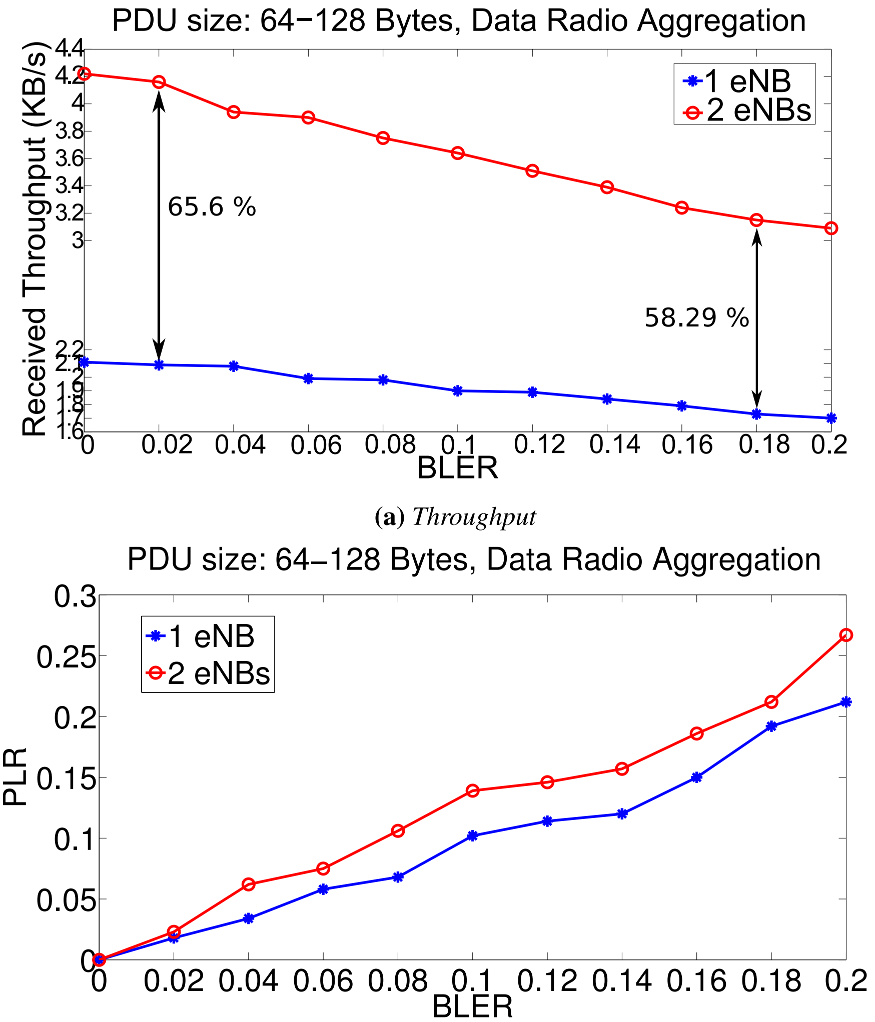
(c) *Received throughput*

Figure 5.8: *OAI Measurement Results of a LTE wireless mesh network enabled by eUEs.*

eUEs) even in bad communication condition with BLER up to 20 %.

eUE Performance Improvements

Fig. 5.9 illustrates the measured results for the scenario where an eUE exploits multiple eNB connectivity (here two) to receive the desired service. In this scenario, the payload size ranges from 64-128 Bytes and we measure the received throughput gain when the eUE is served by two eNBs versus a sole eNB service for different BLER probabilities. UDP constant bit rate traffic of 2.1 KB/s is transmitted by both eNBs. The queue size has no impact at all as the eUE absorbs traffic. As it can be observed in Fig. 5.9(a) the eUE improves its throughput (up to ∼65%) when experiences a dual eNB connectivity and maintains this difference slightly reduced as the BLER increases. This slight throughput reduction is due to the PLR that increases as the bad channel quality affects the communication (see Fig. 5.9(b)).



(a) *Throughput*

(b) *Packet loss rate*

Figure 5.9: *OAI Measurement Results of an eUE experiencing multiple eNB communication.*

5.5 Related Work

In pursuit of designing a novel architecture for future cellular networks, we differentiate from prior studies in the literature as they are summarized below.

Backhaul access: The explosion in mobile networking and multimedia applications have swiftly shifted backhauling from a state of low priority to a centre of attention. In [75], Andrews distills the most important shifts in HetNets arguing that efficient backhauling through picos and femtos can significantly affect both operators and users. An extensive study is made about the verification of functionality and feasibility for backhaul access solutions in heterogeneous networks with self-organizing features. Work in [96] constitutes a case study surveying the key factors and specifying requirements to support a highly scalable and flexible backhaul access solution. The above study presents the need to offer an auto-provisioning LTE X2 connectivity as a solution of a network architecture that can provide enhanced user experience at the cell edge as well as a reduced OPEX/CAPEX to the operator. A substantial number of works provide extensive study about the verification of functionality and feasibility for backhaul access solutions in cellular and heterogeneous networks (HetNets). In another survey [76] for HetNet architectures, relaying and mobile wireless backhauling pose significant design and deployment challenges so as to be capable of introducing an automation toward self-organizing-networks (SON). Moreover, enabling HetNet interworking and relaying over backhauls so that femtos and picos assist macro base station are expected to yield important performance improvements in future-generation small cell networks. A significant preliminary study in [97] discusses issues about enabling relaying to LTE Rel-8 compatible systems as dynamic backhauling access solution promising to offer improved load management within the cell and reduced operational expenditure (OPEX)to the operators. A significant limitation for the potential benefits is raised by the appropriate selection of the backhaul link. In [98], the authors show that when an over-the-air backhaul opportunity exists, this can offer significant delay improvements through a rate splitting technique at the cost of allocation of spectrum resources. Moreover, in [99] authors devised a scheme for heterogenous backhaul connections that can jointly decide on the femtos cooperation and backhauling activation by enabling coordinated Multipoint (CoMP) transmissions among cooperative femtos. In the above work, it was shown that a suitable backhaul selection constitutes one of the main performance bottlenecks in the emerging femtocell networks. Zhao *et al.* in their work [100] identify the impact of backhauling in mobile broadband networks and devised methods to minimize backhaul user data transfer by exploiting CoMP transmissions.

Our approach not only complements those studies but also differs by proposing a flexible and light-weight architecture that can fully exploit eUEs as active networks elements/resources. In, [98], the authors show that when an over-the-air backhaul opportunity exists, this can offer significant delay improvements through a rate splitting technique at the cost of allocation of spectrum resources. Moreover, in [99] authors devised a scheme for heterogeneous backhaul connections that can jointly decide on the femto eNB cooperation and backhauling activation by enabling coordinated Multipoint (CoMP)transmissions among cooperative femto eNBs.

Relaying and packet forwarding: In their seminal works [101], Sendonaris *et al.* advocate that in a multi-hop relay scenario, cooperative communication is being considered as an efficient solution for transmissions, as it provides robust and resilient forwarding by recruiting multiple intermediate relays to collaboratively transmit the source signal to the destination. Authors in [102,103], compare different relay types and topologies in LTE-Advanced and WiMAX standards. Simulation results show that relay technologies can effectively improve service coverage and system throughput. Those works motivate the applicability of packet forwarding solutions in relay assisted networks.

Our work is differentiated from the above in the sense that it rethinks the end-to-end information transfer as relays terminate the S1-AP protocol. The current practice considers relays as a part of network planning that maintain sufficient backhaul access to the core (MME/S-GW), while in our architecture eUEs adopt a light-weight design to effectively enable new use cases in small and moving cell scenarios.

Although both relays and eUEs operation requires multiple connectivities, a distinctive characteristic is that RNs play the role of UE with respect to Donor eNB and the role of eNB with respect to the UEs, while the introduced eUEs remain always UEs with respect to the connected eNBs. In [83], authors discuss the evaluation of a cooperative relaying scheme that exploits QMF for Wi-Fi infrastructure networks. A PHY layer design is presented motivating the need for a close interaction with a suitable MAC to exploit the benefits of relaying diversity either through link switching or link cooperation. Our approach considers cooperative MAC forwarding by exploiting the underlying relaying scheme.

Cooperative Communications: A significant work [104] that is strongly related to our study present a novel cross-layer design that exploits benefits of PHY/MAC interworking to enable cooperative relaying in a WIMAX system (a competitive to LTE technology). Authors propose a MAC layer protocol, named CoopMAX that is in compliance with WiMAX standards to allow for leveraging intermediate relays to service cell edge users. Contrary to the above, our work introduces the concept of collaborative radio bearer establishment in LTE, where multiple eUEs can be leveraged to support backhaul access connectivity to core isolated or moving eNBs as well as to benefit by multiple inter-node radio aggregation. In a preliminary work in [91], we studied the throughput efficiency in the presence of two UEs when DF is used in system-level simulator in absence of any protocol. We enhanced this preliminary work aiming at enabling low latency transfer in inter-cluster communications, introducing a full protocol implementation mechanism for MAC/L2 packet forwarding that exploits buffer aware scheduling.

5.6 Conclusion

In this chapter, a novel architecture for next generation cellular networks is proposed to evolve eUEs as active network elements. They form a virtual MIMO antenna and forward packets at L2/MAC by performing packet level cooperation over a virtual air-interface. As the simmering interest for efficient solutions on public safety and moving cell scenarios grows, our cost-effective approach exploits eUEs cooperation to enable non-ideal wireless backhaul access and provide resilient and low latency communication. The evaluation of the prototype implementation in OAI [95] demonstrates that the proposed architecture achieves low latency and reliable communication satisfying the guaranteed-bit-rate (GBR) traffic requirements. Moreover, eUEs benefit from their participation in this collaboration by improving their throughput performance.

Chapter 6

Closer to Cloud Radio Access Network: Study of Critical Issues

6.1 From Analog to Virtual Radio

Through the past decades, radio access network (RAN) has significantly evolved from analog to digital and from hardware to software in every aspect ranging from signal processing and hardware technologies. The demand for flexible and reusable radios has led to a shift where all or some of the radio service is software defined [105]. In a pure software-defined radio (SDR), the entire radio service is based on a general-propose processor (GPP) and would only require analog-to-digital and digital-to-analog conversions, power amplifier, and antenna to stream the data in/out the receiver and transmitter, where as in the other cases it is based on a programmable dedicated hardware (e.g. ASIC, ASIP, or DSP) and its associated software. The decision depends on the required flexibility, power efficiency (e.g. ASIC: 1x, DSP: 10x, GPP: 100x), and the cost.

Yet another evolution is that of virtualized RAN. This refers to the abstraction (or virtualization) of the execution environment and optionally the hardware from the radio service. Consequently, radio service becomes a general-purpose application that operates on the top of virtualized environment, and interacts with the physical resources either through a full or partial hardware emulation layer or directly. The resulted virtualized software radio application can be delivered a service and managed through a controller following the cloud computing principles [106]. This gives rise to a new principle of RAN as a service (RANaaService) [107, 108] associated with cloud RAN (C-RAN) [6, 109, 110] that migrates from a slow-moving proprietary and expensive hardware/software platforms (vertical solution) towards an open virtualized software platform based on the commodity hardware (horizontal solution).

6.2 Background

Motivated by the explosive growth of mobile data traffic, mobile operators are facing pressure to increase the network capacity without compromising the coverage to keep up with demand [111]. Leveraging smaller cells could significantly increase the capacity through a better spatial reuse. Nevertheless, this comes with a higher capital and operational expenditure, higher interference from the neighbouring cells, higher chance that a base station carry no or low traffic workload due to spatio-temporal traffic fluctuations, and finally higher power consumption due to increasing number of base stations [6, 109, 112–114]. Cloud-RAN is one of the candidate architecture to answer the above mentioned challenges. Other complementary solutions have been proposed, namely heterogeneous and small cell networks (HetSNets) and massive MIMO [115], and their applicability depends on the use case, cost per bit, and existing infrastructure. However, both HetSNets and massive MIMO can benefit from centralization in the C-RAN architecture. More specifically, centralization enables to exploit workload variations across multiple cell sites in a relatively large geographical area over time [116, 117], save energy and cost obtained from statistical multiplexing gain among multiple RANs, and ease the network upgrade and maintenance [110]. In addition by exploiting joint and coordinated signal processing (property obtained through RAN centralization), significant increase in spectral efficiency is achievable. Nevertheless, the benefits from centralization comes at the cost of tighter real-time deadlines due to extra delay caused by the transport network, slower yet more complex software processing, and overhead of the virtualized environment.

Many architectures have been proposed for C-RAN ranging from a partially accelerated to a fully software-defined modem. Recent efforts have shown the feasibility and efficiency of a full software implementation of LTE RAN functions over General Purpose Processors (GPPs). Several software implementations of the LTE eNB already exist, namely Intel solutions based on a hybrid GPP-accelerator architecture aiming at a balance between a flexible IT platform, high computing performance and good energy efficiency,[1] Amarisoft LTE solution featuring a fully-functional pure-software LTE,[2] and the Eurecom OpenAirInterface platform, which is a fully functional open-source Software Defined Radio (SDR) implementation of LTE/LTE-A.[3]

The full GPP approach to C-RAN brings the cloud and virtualization technologies even closer to the wireless world by leveraging commodity IT platforms and thus increases the flexibility through cloudification of RAN and its delivery as a service. This brings a set of benefits, namely on-demand and elastic resource provisioning, rapid deployment and service provisioning, load balancing, and controlled and metered services [106]. It also enables new set of use cases such as MVNO as a service where value-added content and services could be provided to end users through the emergence of MVNO that are not necessarily dependent to MNO. Enabled by soft and virtualized RAN, the cloud approach to RAN has several attractive properties.

For the operator:

- It reduces the CAPEX of the mobile network operators (MNOs) via a flexible and incremental evolution of the processing power needed by the radio network to follow the service demands;
- It reduces the OPEX of the MNOs by lowering the footprints of the radio network deployment;
- It enables the definition of an attractive bundle of rich services that can be used by MNOs

[1] http://www.intel.com/content/www/us/en/communications/communications-c-ran-solution-video.html

[2] http://www.amarisoft.com/

[3] http://www.openairinterface.org

or mobile virtual network operators (MVNOs) or specific brands of the MNO.

For the end-user:

- It helps in defining feature-rich services in a flexible manner;
- It provides means to ensure a better Quality-of-Experience (QoE) via base station coordination and cooperation;
- It fosters the emergence of new entrants in the Mobile community by decoupling the network investments and operations from the service delivery and content management, leading to innovation and competition.

C-RAN has also been supported by many players, namely mobile operators (e.g. China Mobile and France Telecom), equipment vendors (e.g. Alcatel-Lucent LightRadio and Nokia-Siemens LiquidRadio), European Commission (e.g. MCN and iJoin projects), and standardization bodies (e.g. NGMN Alliance, ETSI). NGMN extended the study on C-RAN in a C-RAN work stream under the project of RAN Evolution investigating key C-RAN technologies including pooling, RAN sharing, function split between the base band unit (BBU) and the remote radio unit (RRU), and C-RAN virtualization [118]. This approach is considered by European projects such as Mobile Cloud Networking (MCN) [107], Sail [119] and iJoin [108].

6.3 Cloud Radio Access Network Concept

Unlike typical RANs, the C-RAN decouples the baseband unit (BBUs) from the radio units by locating the BBUs at the high performance cloud infrastructure. This replaces traditional base stations with distributed radio elements, called remote radio heads (RRHs), with much smaller footprints than a base station with onsite processing, allowing for simpler network densification [120]. A high speed transport medium (e.g. fibre, passive WDM, cable, microwave) is used to carry the data between BBUs and RRH. This portion of the network is called fronthaul and its bandwidth requirements depends on the nature of the data transported (i.e. analog/digital, layer 1/2) [121–123]. The cloud RAN concept is shown in Fig. 6.1 and includes three main components: (1) BBU pool, a set of virtualized base stations (VBS) on the top of cloud infrastructure, (2) RRH, light weight radio unit with antennas, and (3) fronthaul, data distribution channel between the BBU pool and RRHs.

It can be seen that a pool of BBUs is virtualized inside the same (or different) cloud infrastructure and shared among the cell sites. VBS can communicate with core networks (CN) through a dedicated interface (e.g. S1 in LTE), and with each other directly through another interface (e.g. X2 in LTE). VBS can rely on the cloud infrastructure to provide localized edge service such as content caching and positioning, and network APIs to interact with the access and core networks [63]. Since a typical BBU pool hosts 10 - 1000 base stations, transport of the generated in-phase and quadrature (I/Q) data from BBU to RRH requires a high fronthaul capacity. Thus, depending on the fronthaul capacity and the VBS/BBU architecture, different functional split may exist between BBU and RRH to reduce the data rate in fronthaul ranging from all-in-BBU to all-in-RRH. Another aspect is the the mapping or clustering between the BBUs and RRHs in the fronthaul. While such a mapping can be dynamically done depending on different

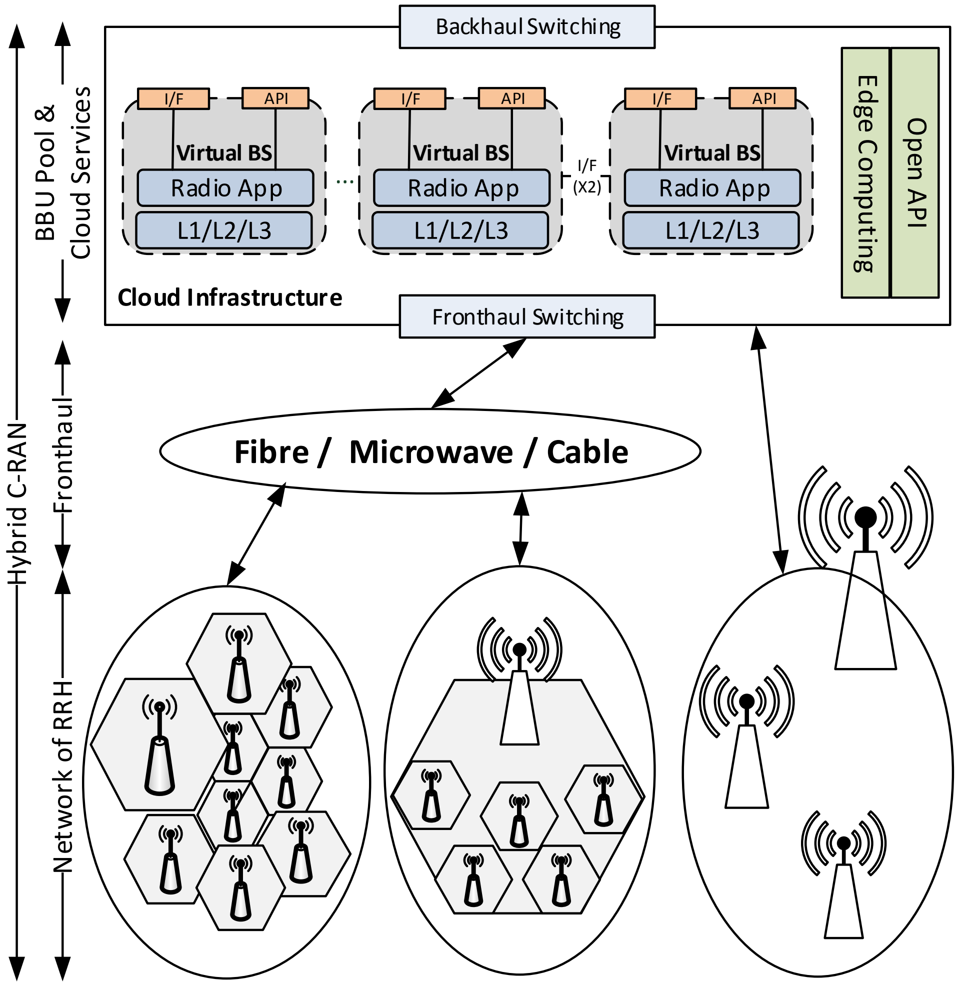


Figure 6.1: *C-RAN Concept*

factors including traffic load, energy efficiency [121, 124], several limitations exist due to the underlying synchronous links and the required high precision clock in the current standard.

6.4 C-RAN Use Cases

This section highlights some examples of use cases and benefits that have already been demonstrated and enabled by the cloudification of RAN. Broadly, there are two classes of use cases as briefly named below.

- **RAN-Centric:** exploits RAN centralization and improves the performance of the radio through a joint and/or coordinated operation in multi-user and multi-cell networks. Well-known examples includes [109, 116]:
  - Capacity and coverage extensions in terms of additional BBU and RRH for dense or sparse areas;

  - Coordinated multi-point transmission (CoMP);
  - Fast (X2 or S1) Handover for high mobility scenarios;
  - Dynamic load balancing between the BBUs and RRH.

- **Cloud-Centric:** leverages the cloud infrastructure to provide a feature rich services through the edge computing, networking and storage. This enables a marketplace for cell application and services. Example includes [63, 125]:

  - RAN-aware optimization: caching, proxy, URL filtering, compression, video optimization;
  - Application-aware optimization: rate and performance adaptation, location and tracking services;
  - Analytic and user experience: personalized advertisement and recommendations, prediction, system knowledge-base, video/data analytic.

6.5 C-RAN Critical Issues

While C-RAN comes with many attractive features and recent efforts have partially shown its feasibility, three critical issues need to be thoroughly investigated in order to assess the feasibility of C-RAN and identify the main design choices.

1. **Capacity requirement for fronthaul:** Because a typical BBU pool should support 10 - 1000 base stations, transport of the I/Q samples from BBU to RRH requires a high fronthaul capacity. To meet the BBU timing requirements, fronthaul must provide an upperbound for the maximum one-way latency. Furthermore, clock synchronization across BBUs and RRH over the fronthaul also imposes a very low jitter.

2. **Latency requirements for BBU:** FDD LTE HARQ requires a round trip time (RTT) of 8ms that imposes an upper-bound for the sum of BBU processing time and the fronthaul transport latency.

3. **Real-time requirement for Operating System and virtualization environment:** Execution environment of BBU pool must provide (statistical) guarantee for the BBU pool successfully meeting their real-time deadlines related to the frame/subframe timing. It should also provide dynamic resource provisioning/sharing and load balancing to deal with the cell load variations.

In addition to above issues, C-RAN also brings many other challenges to BBU, RRH, and fronthaul. Front-haul multiplexing and topology, optimal mapping (clustering) between BBUs and RRHs, efficient BBU interconnections, cooperative radio resource management, energy optimization and harvesting techniques, and channel estimation are just few examples.

The following subsections focus on the critical issues, and present C-RAN feasible architectures.

6.5.1 Fronthaul Capacity

Many factors contribute to the data rate of the fronthaul, which depends on the cell and fronthaul configurations. Equation 6.1 calculates the required data rate based on such configurations.

$$C_{fronthaul} = \underbrace{2 \cdot N_{antenna} \cdot M_{sector} \cdot F_{sampling} \cdot W_{I/Q} \cdot C_{Carriers}}_{\text{cell configuration}} \cdot \underbrace{O_{coding+proto} \cdot K}_{\text{fronthaul configuration}} \tag{6.1}$$

where $N_{antenna}$ is the number of Tx/Rx antennas, M_{sector} is the number of sectors, $F_{sampling}$ represents the sampling rate, $W_{I/Q}$ is the bit width of a symbol,$O_{proto+coding}$ is the ratio of transport protocol and coding overheads, and K is the compression factor. The following table shows the required data rate for different configurations. An overall overhead is assumed to be 1.33, which takes into account the protocol overhead ratio of 16/15 and the line coding of 10/8 (CIPRI case).

Table 6.1: FRONTHAUL CAPACITY FOR DIFFERENT CONFIGURATIONS

Bandwidth	$N_{antenna}$	M_{sector}	$F_{sampling}$	$W_{I/Q}$	$O_{coding+proto}$	$C_{Carriers}$	K	Data Rate
1.4MHz	1x1	1	1.92	32	1.33	1	1	163Mb/s
5MHz	1x1	1	7.68	32	1.33	1	1	650Mb/s
5MHz	2x2	1	7.68	32	1.33	1	1	1.3Mb/s
10MHz	4x4	1	15.36	32	1.33	1	1/2	2.6Gb/s
20MHz	1x1	1	30.72	32	1.33	1	1	2.6Gb/s
20MHz	4x4	1	30.72	32	1.33	1	1/3	3.4Gb/s
20MHz	4x4	1	30.72	32	1.33	1	1	10.4Gb/s

Further data rate reduction can be obtained by offloading the BBU functions to RRH, relevant example is FFT/IFFT (see Fig. 6.2). The trade-off has to be made between the available fronthaul capacity, complexity, and the resulted spectral efficiency. Regardless of different possibilities in BBU function split, the fronthaul should still maintain the latency requirement to meet the HARQ deadlines. NGMN adopts fronthaul maximum one-way latency of 250μ [118]. Different protocols have been standardized for the fronthaul, namely CPRI (common public radio interface) representing 4/5 of the market, OBSAI (Open Base Station Architecture Initiative) representing 1/5 of the market, and more recently the Open Radio Interface (ORI) initiated by NGMN and now by ETSI ISG (Industry Specification Group) [126].

6.5.2 BBU Processing

Fig. 6.2 illustrates the main RAN functions in both TX and RX spanning all the layers, which has to be evaluated to characterise the BBU processing time and assess the feasibility of a full GPP RAN. Since the main processing bottleneck resides in the physical layer, the scope of the analysis in this chapter is limited to the BBU functions. From the figure, it can be observed that the overall processing is the sum of cell- and user-specific processing. The former only depends on the channel bandwidth and thus imposes a constant base processing load on the system, whereas the latter depends on the MCS and resource blocks allocated to users as well as SNR and channel conditions. The figure also shows the interfaces where the functional split could happen to offload the processing either to an accelerator or to a RRH.

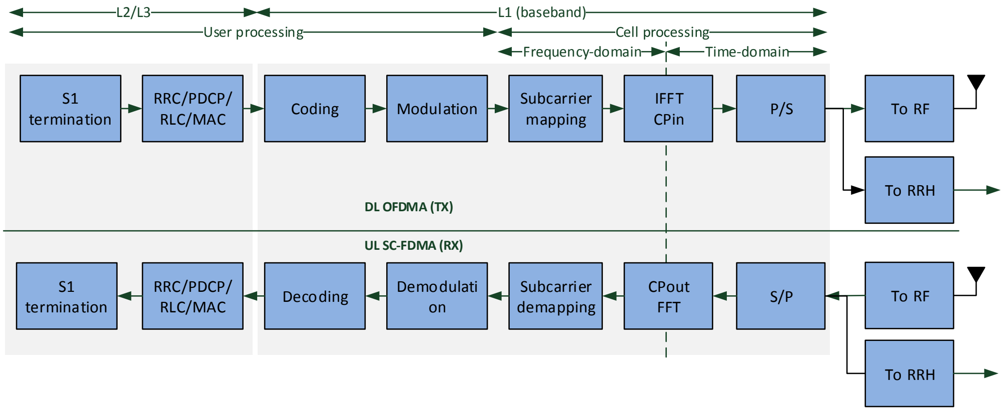


Figure 6.2: *Functional Block diagram of LTE eNB for DL and UL*

To meet the timing and protocol requirements, the BBU processing must finish before the deadlines. One of the most critical processing that requires deadline is imposed by the Hybrid Automatic Repeat Request protocol (HARQ) in that every received MAC PDU has to be acknowledged (ACK'ed) or non-acknowledged (NACK'ed) back to the transmitter within the deadline. In FDD LTE, the HARQ Round Trip Time (RTT) is 8 ms. Each MAC PDU sent at subframe N is acquired in subframe $N + 1$, and must be processed in both RX and TX chains before subframe $N + 3$ allowing ACK/NACK to be transmitted in subframe $N + 4$. On the receiver side, the transmitted ACK or NACK will be acquired in subframe $N + 5$, and must be processed before subframe $N + 7$, allowing the transmitter to retransmit or clear the MAC PDU sent in subframe N. Figure 6.3(a) and 6.3(b) show an example of timing deadlines required to process each subframe in downlink and uplink respectively.

It can be observed that the total processing time is 3ms, out of which 2ms is available for RX processing and 1ms for TX. Thus the available processing time for an eNB to perform the reception and transmission is upper-bounded as follows:

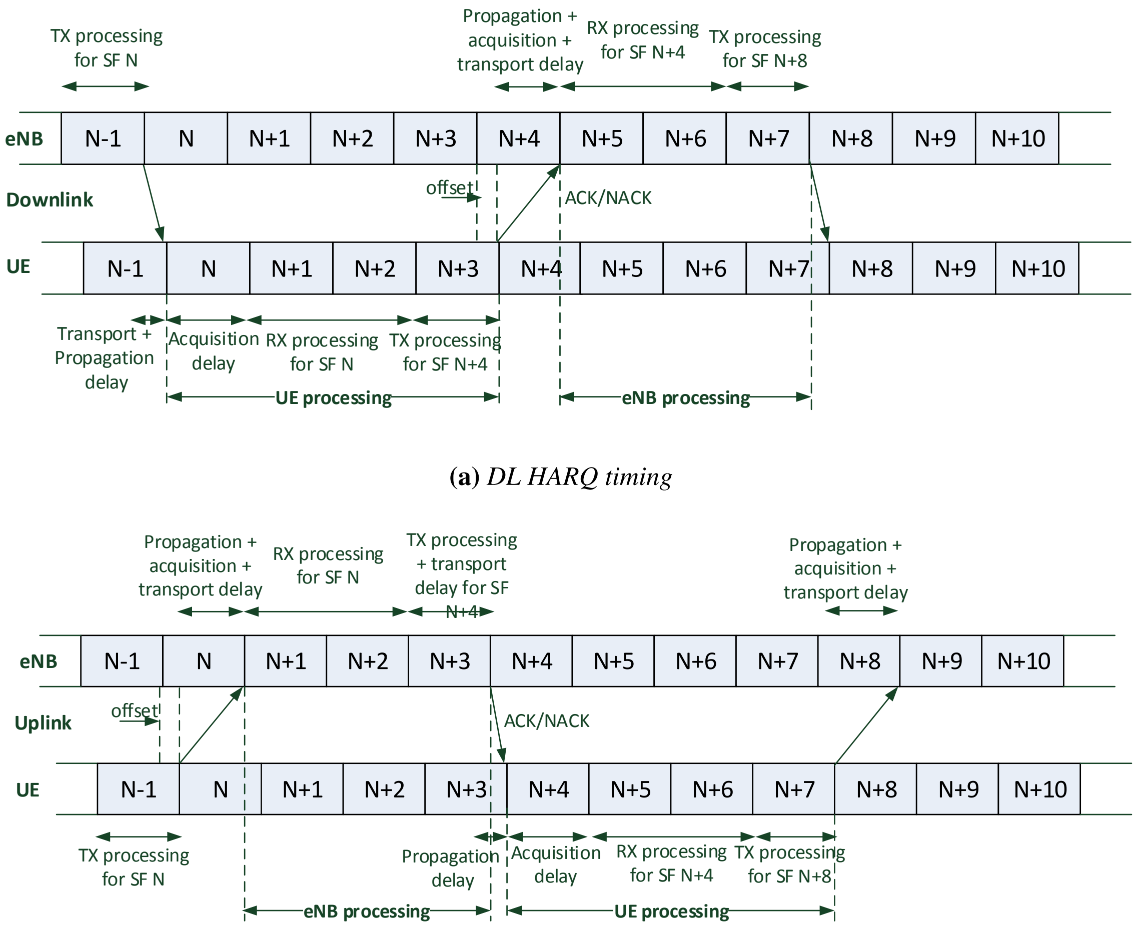
$$T_{Rx} + T_{Tx} \leq T_{HARQ}/2 - (T_{Propagation} + T_{Acquisition} + T_{Transport} + T_{offset}) \tag{6.2}$$

where $T_{HARQ} = 8$, $T_{Propagation} + T_{Acquisition} + T_{Transport} \leq 1ms$, and $T_{offset} = 0$ in DL.

Depending on the implementation, the maximum tolerated transport latency depends on the eNodeB processing time and HARQ period. As mentioned earlier, NGMN adopted a 250 μs for the maximum one-way fronthaul transport latency. Hence, the length of a BBU-RRH link is limited to around 15 km to avoid too high round-trip-delays (given that the speed of light in fiber is approximately 200 m/μs). At maximum distance of 15 km, the remaining overall processing time will be between 2.3–2.6 ms.

6.5.3 Real-time Operating System and Virtualization Environment

A typical general purpose operating systems (GPOS) is not designed to support real-time applications with hard deadline. Hard real-time applications have strict timing requirements



(a) *DL HARQ timing*

(b) *UL HARQ timing*

Figure 6.3: *FDD LTE timing*

to meet deadlines or otherwise unexpected behaviours can occur compromising the performance. For instance Linux is not a hard real-time operating system as the kernel can suspend a task when its runtime has completed and it can remain suspended for an arbitrarily long time. Kernel scheduling is the process in the OS that decides which task to run and allocates certain processing time to it. Such a scheduler is essential to guarantee the worst case performance and also to provide a deterministic behaviour (with short interrupt-response delay of 100 μs) for the real-time applications. Recently, a new scheduler, named SCHED_DEADLINE, is introduced in the Linux kernel mainstream that allows each application to set a triple $(runtime[ns], deadline[ns], period[ns])$, where $runtime \leq deadline \leq period$.[4] As a result, the scheduler preempts the kernel to meet the deadline and allocates the required runtime (i.e. CPU time) to each period.

Software-based Radio, is a real-time application that requires a hard deadlines to maintain the frame and subframe timing. In the C-RAN setting, the software radio application runs on a virtualized environment, where the hardware is either fully, partially, or not virtualized. Two

[4] http://en.wikipedia.org/wiki/SCHED_DEADLINE

main approaches exist to virtualization: virtual machines (e.g KVM[5] and Xen[6]) or containers (e.g. Linux Container LXC[7] and Docker[8]) as shown in Fig. 6.4. In a virtual machine (VM), a complete operating system (guest OS) is used with the associated overhead due to emulating virtual hardware whereas containers use and share the OS and device drivers of the host. While VMs rely on the hypervisor to requests for CPU, memory, hard disk, network and other hardware resources, containers exploits the OS-level capabilities. Similar to VMs, containers preserve the advantage of virtualization in terms of flexibility (containerize a system or an application), resource provisioning, decoupling, management and scaling. In short, containers are lightweights as they do not emulate a hardware layer (share the same kernel and thus application is native with respect to the host) and therefore have a smaller footprint than VMs, start up much faster, and offer near bar metal runtime performance. This comes at the expense of less isolation and greater dependency to the host's kernel.

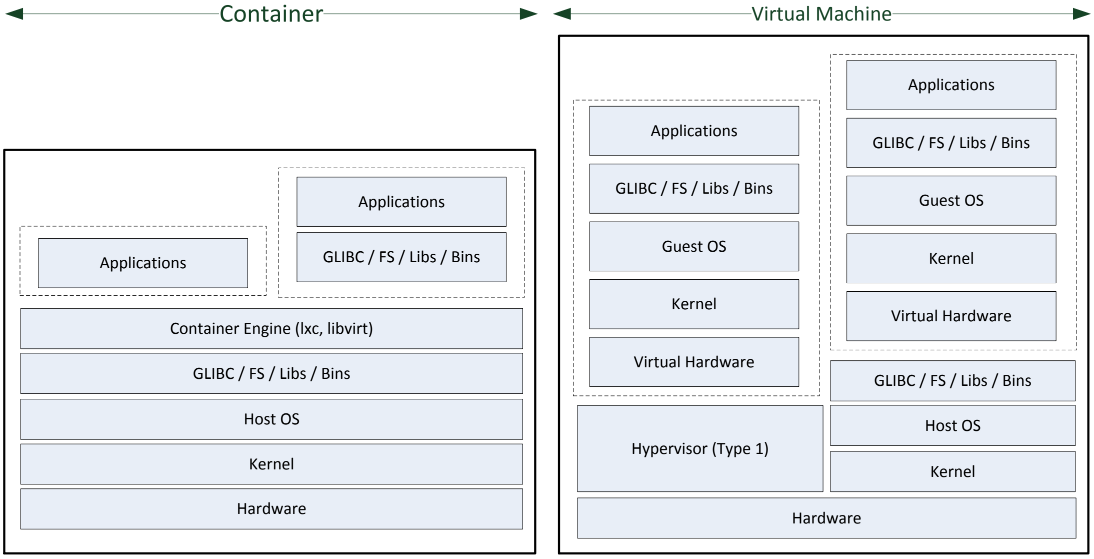


Figure 6.4: *Comparison of a virtual machine and container virtualized environment.*

Two other important aspects when targeting RAN virtualization are:

- **I/O Virtualization:** I/O access is a key for a fast access to the fronthaul interface and to the hardware accelerators that might be shared among BBUs. In VM, IO virtualization is done through the hardware emulation layer under the control of hypervisor, where as in container this is done through the device mapping. Thus, direct access to the hardware is easier in containers than VMs as they operate at the host OS level. In VM, additional techniques might be needed (e.g. paravirtualization or CPU-assisted virtualization) to provide a direct or fast access to the hardware.

[5] http://www.linux-kvm.org
[6] http://www.xenserver.org/
[7] linuxcontainers.org
[8] www.docker.com

- **Service composition of the software radio application:** A VBS can be defined as a composition of two types of service [107], atomic service that executes a single business or technical function and is not subject to further decomposition, and composed service that aggregates and combines atomic services together with orchestration logic. An atomic service in RAN can be defined on per carrier, per layer, per function basis. For instance, a VBS could be defined as a composition of layer 1 and layer2/3 services.

6.5.4 Candidate Architectures

While the concept of C-RAN has been clearly defined, more research is needed to find an optimal architecture that maximizes the benefits behind C-RAN [110], and based on which a true proof-of-concept could be built. From the perspective of the operator such an architecture has to meet the scalability, reliability/resiliency, cost-effective requirements. However, from the perspective of the software radio application, two main requirements have to be met: (1) strict hard deadline to maintain the frame and subframe timing, and (2) efficient/elastic computational resources (e.g. CPU, memory) to perform intensive digital signal processing for different transmission modes (beamforming, CoMP, etc).

Broadly, three main choices are possible to design a C-RAN, each of which provide a different cost-power-performance-flexibility trade-offs.

- **Full GPP:** where all the processing (L1/L2/L3) is performed on the host/guest systems. According to China Mobile, the power consumption of the OpenAirInterface full GPP LTE softmodem is around 70w per carrier [109].

- **Accelerated:** where only certain functions, such as FFT/IFFT, are offloaded to a dedicated hardware (e.g. FPGA, DSP), and the remaining functions operate on the host/guest. The power consumption is reduced to around 13 18w per carrier.

- **System-on-Chip:** where the entire L1 is performed on a SoC and the reminder of the protocol stack runs on the host/guest. The power consumption is reduced to around 8w per carrier.

As shown in Fig. 6.5, the hardware platform can either be full GPP or a hybrid. In the later case, all or part of the L1 functions might be offloaded to dedicated accelerators, which can be placed locally at the cloud infrastructure to meet the real-time deadline and provide a better power-performance trade-off or remotely at RRH to reduce the data rate of fronthaul. Different service compositions can be considered, ranging from all-in-one software radio application virtualization to per carrier, per layer or per function virtualization as mentioned earlier. The virtualization is performed either by a container engine or a hypervisor, under the control of a cloud OS, which is in charge of life-cycle management of a composite software radio application (orchestrator) and dynamic resource provisioning.

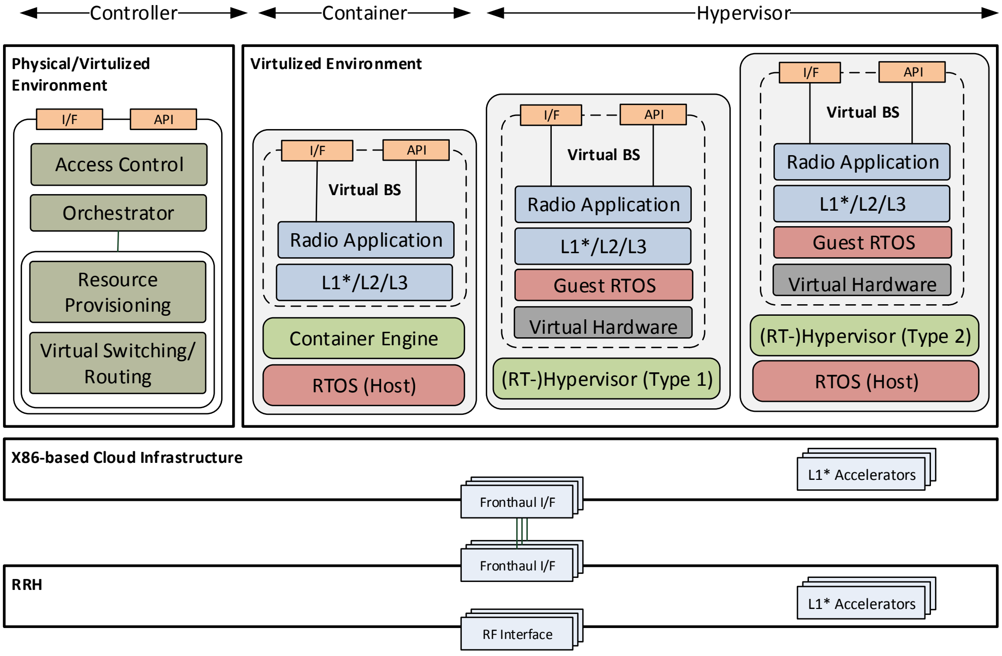


Figure 6.5: *Candidate C-RAN architectures.*

6.6 Evaluation

6.6.1 Experiment Setup

Four set of different experiments are performed. The first experiment analyses the impact of different x86 CPU architecture on BBU processing time, namely Intel Xeon E5-2690 v2 3Ghz (same architecture as IvyBridge), Intel SandyBridge i7-3930K at 3.20Ghz, and Intel Haswell i7-4770 3.40GHz. The second experiment shows how the BBU processing time scale with the CPU frequency. The third experiment benchmarks the BBU processing time in different virtualization environments including LXC, Docker, and KVM against a physical machine (GPP). The last experiment measures the I/O performance of virtual Ethernet interface through the guest-to-host round-trip time (RTT).

All the experiments are performed using the OpenAirInterface DLSCH and ULSCH simulators designed to perform all the baseband functionalities of an eNB for downlink and uplink as in a real system. All the machines (hosts or guests) operate on Ubuntu 14.04 with the low latency (LL) Linux kernel version 3.17, x86-64 architecture and GCC 4.7.3. To have a fair comparison, only one core is used across all the experiments with the CPU frequency scaling deactivated except for the second experiment.

The benchmarking results are obtained as a function of allocated physical resource blocks (PRBs), modulation and coding scheme (MCS), and the minimum SNR for the allocated MCS for 75% reliability across 4 rounds of HARQ. Note that the processing time of the turbo decoder depends on the number of iterations, which is channel-dependant. The choice of minimum SNR for a MCS represents the realistic behavior, and may increase number of turbo iterations. Additionally, the experiments are performed using a single user with no mobility, SISO mode with AWGN channel, and a full buffer traffic ranging from 0.6Mbps for MCS 0 to

64Mbps for MCS 28 in both directions. Note that if multiple users are scheduled within the same subframe in downlink or uplink, the total processing depends on the allocated PRB and MCS, which is lower than a single user case with all PRBs and highest MCS. Thus, the single user case represents the worst case scenario.

The processing time of each signal processing module is calculated using timestamps at the beginning and at the end of each BBU function. OAI uses the `rdtsc` instruction implemented on all x86 and x64 processors to get a very precise timestamps, which counts the number of CPU clocks since reset. Therefore the processing_time is proportional to the value returned by the following pseudo-code:

```
start = rdtsc();
bbu_function();
stop = rdtsc();
processing_time = (stop - start)/cpu_freq;
```

Listing 6.1: *Processing time measurement method*

To allow a rigorous analysis, total and per function BBU processing time are measured as shown in Table 6.2. For statistical analysis, a large number of processing_time samples (10000) are collected for each BBU function to calculate the average, median, first quantile, third quantile, minimum and maximum processing time for all the subframes in uplink and downlink.

Table 6.2: OAI BBU PROCESSING TIME DECOMPOSITION IN DOWNLINK AND UPLINK

RX Function	timing(us)	TX Function	timing(us)
OFDM Demodulation	109.695927	OFDM modulation	108.308182
ULSCH Demodulation	198.603526	DLSCH modulation	176.487999
ULSCH Decoding	624.602407	DLSCH scrambling	123.744984
⌊ *interleaving*	12.677955	DLSCH encoding	323.395231
⌊ *demultiplexing*	117.322641	⌊ trubo encoder	102.768645
⌊ *rate matching*	15.734278	⌊ rate matching	86.454730
⌊ *turbo decoder*	66.508104	⌊ interleaving	86.857803
⌊ *init*	11.947918		
⌊ *alpha*	3.305507		
⌊ *beta*	3.377222		
⌊ *gamma*	1.018105		
⌊ *ext*	2.479716		
⌊ *intl*	5.441128		
Total RX	931	Total TX	730

6.6.2 CPU Architecture Analysis

Fig. 6.6 depicts the BBU processing budget in both directions for the considered Intel x86 CPU architecture. It can be observed that the processing load increases with the increase of PRB and MCS for all CPU architectures, and that it is mainly dominated by the uplink. Furthermore, the ratio and variation of downlink processing load to that of uplink also increases with the increase of PRB and MCS. Higher performance (lower processing time) is achieved by the Haswell architecture followed by SandyBridge and Xeon. This is primarily due to the respective clock frequency (c.f. Section 6.6.3, but also due to a better vector processing and faster single threaded performance of Haswell architecture.[9] For the Haswell architecture, the performance can be further increased by approximately a factor of two if AVX2 (256-bit SIMD compared to 128-bit SIMD) instructions are used to optimize the turbo decoding and FFT processing.

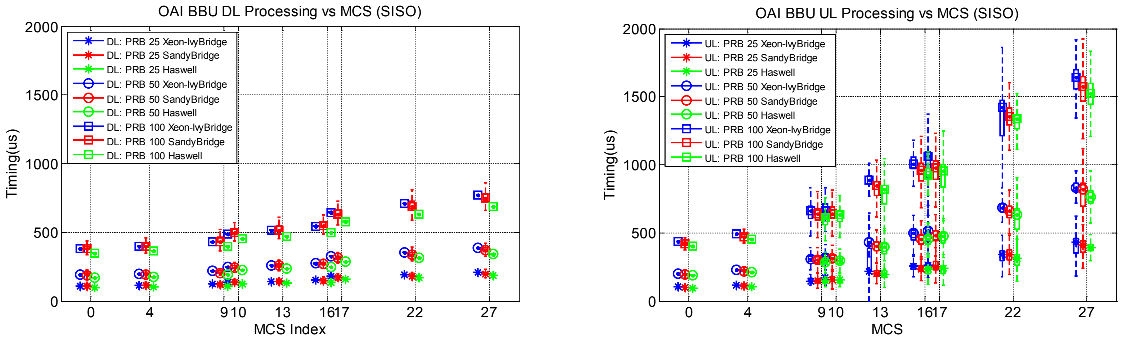


Figure 6.6: *BBU processing budget in downlink (left) and uplink(right) for different CPU architecture.*

6.6.3 CPU Frequency Analysis

Fig. 6.7 illustrates the total BBU processing time as a function of different CPU frequencies (1.5, 1.9,2.3,2.7,3.0, and 3.4 GHz) on the Haswell architecture. The most time consuming scenario is considered with 100 PRBs and downlink and uplink MCS of 27. In order to perform experiments with different CPU frequencies, Linux ACPI interface and *cpufreq* tool are used to limit the CPU clock [127]. It can be observed that the BBU processing time scales down with the increasing CPU frequency. The figure also reflects that the minimum required frequency for 1 CPU core to meet the HARQ deadline is 2.7GHz.

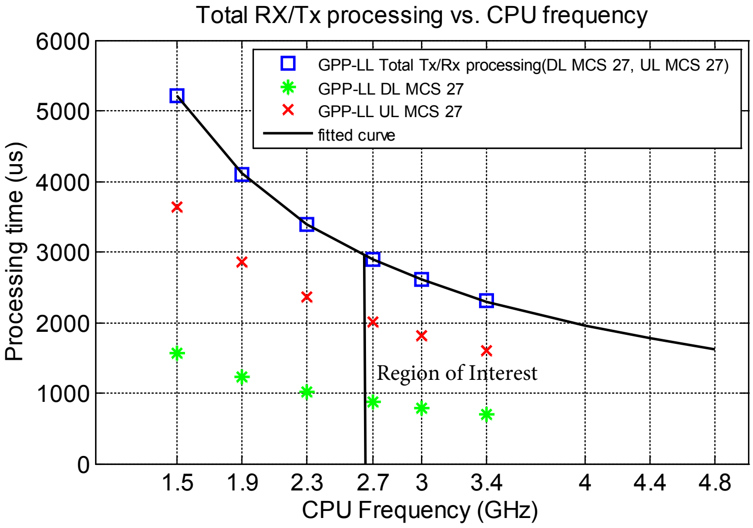


Figure 6.7: *Total processing time as a function of CPU frequency.*

[9] http://en.wikipedia.org/wiki/Haswell_(microarchitecture)

Based on the above figure, the total processing time per subframe, T_{subframe}, can be modelled as a function of CPU frequency [128]:

$$T_{\text{subframe}}(x) \text{ [us]} = \alpha/x$$

, where $\alpha = 7810 \pm 15$ for the MCS of 27 in both directions, and x is CPU frequency measured in GHz.

6.6.4 Virtualization Technique Analysis

Fig. 6.8 compares the BBU processing budget of a GPP platform with different virtualized environments, namely Linux Containers (LXC), Docker, and KVM, on the SandyBridge architecture(3.2GHz). While on average the processing time are very close for all the considered virtualization environments, it can be observed that GPP and LXC have slightly lower processing time variations than that of DOCKER and KVM, especially when PRB and MCS increase.

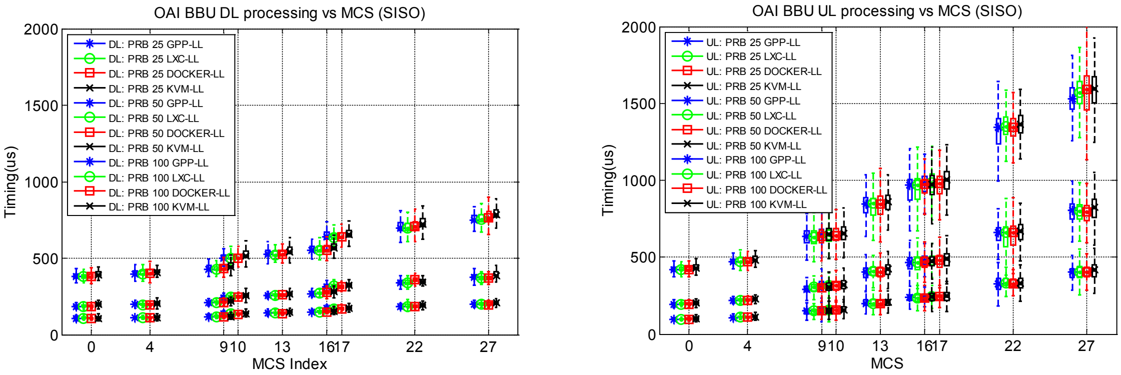


Figure 6.8: *BBU processing budget in downlink (left) and uplink(right) for different virtualized environments.*

Fig. 6.9 depicts the Complementary Cumulative Distribution Function (CCDF) of the overall processing time for downlink MCS 27 and uplink MCS 16 with 100 PRB. The CCDF plot for a given processing time value displays the fraction of subframes with execution times grater than that value. It can be seen that the execution time is stable for all the platforms in uplink and downlink. The processing time for the KVM (hypervisor-based) has a longer tail and mostly skewed to longer runs due to higher variations in the non-native execution environments (caused by the host and guest OS scheduler). Higher processing variability is observed on a public cloud with unpredictable behaviors, suggesting that cares have to be taken when targeting a shared cloud infrastructure [128].

6.6.5 I/O Performance Analysis

Generally, the one-way-delay of fronthaul depends on the physical medium, technology, and the deployment scenario. However in the cloud environment, the guest-to-host interface delay (usually Ethernet) has to be also considered to minimize the access to the RRH interface. To assess such a delay, bidirectional traffics are generated for different set of packet sizes (64, 768,

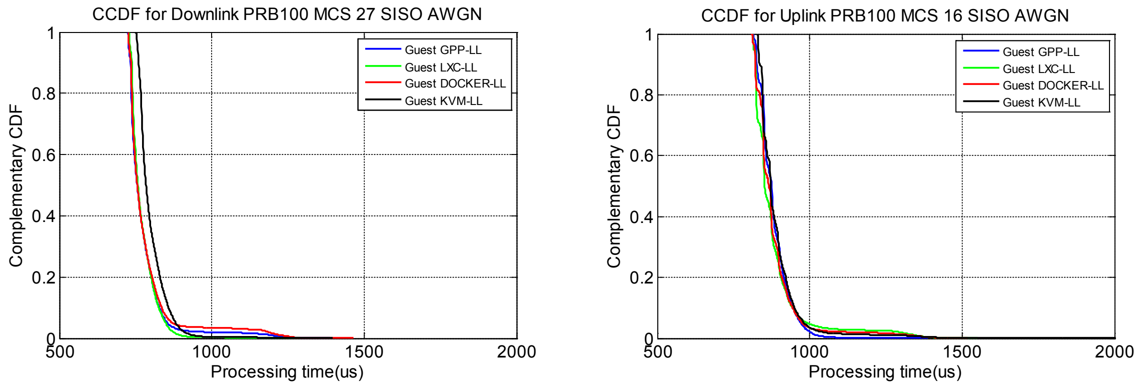


Figure 6.9: *BBU processing time distribution for downlink MCS 27 and uplink MCS 16 with 100 PRB.*

2048,4096,8092) and inter-departure time (1, 0.8, 0.4, 0.2) between the host and LXC, Docker, and KVM guests. It can be seen from Fig. 6.10 that LXC and Docker are extremely efficient with 4-5 times lower round trip time. KVM has a high variations, and requires optimization to lower the interrupt response delay as well as host OS scheduling delay. The results validate the benefit of containerization for high performance networking.

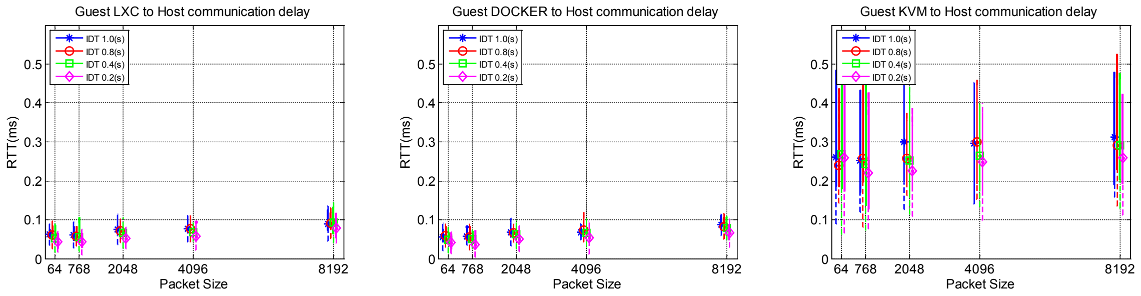


Figure 6.10: *Round trip time between the host and LXC, Docker, and KVM guests.*

6.6.6 Discussion

By analysing the processing for a 1ms LTE sub-frame, the main conclusion that can be drawn for the considered reference setup (FDD, 20MHz, SISO, AWGN) is that with the CPU frequency of 3GHz, (on average) 1 processor core for the receiver processing assuming 16-QAM in uplink and approximately 1 core for the transmitter processing assuming 64-QAM in downlink are required to meet the HARQ deadlines (c.f. Fig. 6.7). Thus a total of 2 cores are needed to handle the total processing of an eNB in 1ms (one subframe). With the AVX2 optimizations for this latest architecture, the computational efficiency is expected to double and thus a full software solution would fit with an average of 1x86 core per eNB.

When comparing the results for different virtualization environments, the main conclusion than can be drawn is that containers (LXC and Docker) offer near bar metal runtime (native) performance while preserving the benefits of virtual machines in terms of flexibility, fast runtime, and migration. Furthermore, they are built on modern kernel features such as `cgroups`, `namespace`, `chroot`, and sharing the host kernel and benefit from the host scheduler, which is a key to meet the real-time deadlines. This makes containers a cost-effective solution without compromising the performance.

6.7 Modelling BBU Processing Time

The evaluation results in Section 6.6 confirm that uplink processing dominates the downlink, and that the total processing increases with PRB and MCS. However, the contribution of each underlying BBU functions to the total processing time and how they scale with the increase of PRB and MCS remains to be analysed so that an accurate model could be build. To this end, three main BBU functions that contribute the most to the total processing are considered including iFFT/FFT, (de)modulation, and (de)coding. For each function, the required processing time is measured for different PRB, MCS, and processing platform (c.f. Fig. 6.11).

The figures reveals that iFFT and FFT increase only with the PRB, while (de)modulation are (de)coding are increasing as a function of PRB and MCS. Each platform also adds a processing offset to each function. It can be seen that decoding and coding functions represent the most time consuming functions in uplink and downlink, and that the decoding is the dominant factor. Note that MCS 9, 16, and 27 corresponds to QPSK, 16QAM, and 64QAM with the highest coding rate. In OAI, decoding and encoding are based on the highly optimized SIMD integer DSP instructions (i.e. 64-bit MMX, 128-bit SSE2/3/4) used to speed up the processing. In a hypervisor-based virtualization, such instructions could add an extra delay if not supported by the hardware emulation layer (c.f. Fig. 6.4).

From Fig. 6.11, it can be observed that the uplink and downlink processing has two components: base processing and dynamic processing load. The base includes cell-processing (iFFT/FFT) for each PRB and the platform-specific processing relative to the reference GPP platform. The dynamic processing load includes user processing (DEC/ENC and DEMOD/-MOD), which is a linear function of allocated PRBs and MCS.[10] The reminder of user processing, namely scrambling, DCI coding, and PDCCH coding, is modelled as the root mean square error (RMSE) for each platform. Fig. 6.12(a) shows the fitted curve for the total processing time for GPP and Fig. 6.12(b) the RMSE for all platforms.

Based on the above results, a model is proposed to compute the total BBU uplink and downlink processing time for different PRB, MCS, and platform, and expressed by the following formula.

$$T_{\text{subframe}}(x, y, w)\ [\text{us}] = \underbrace{c[x] + p[w]}_{base\ processing} + \underbrace{u_r[x]}_{RMSE} + \underbrace{u_s(x, y)}_{dynamic\ processing}$$

where the triple (x, y, w) represents PRB, MCS, and platform. The $c[x]$ and $p[w]$ are the base offsets for the cell and platform processing, $u_r[x]$ is the reminder of user processing, and $u_s(x, y)$ is the specific user processing that depends on the allocated PRB and MCS. The $u_s(x, y)$ is linearly fitted to $a(x)y + b(x)$, where a, b are the coefficients, and y is the MCS. Table 6.3 and 6.4 provide the downlink and uplink model parameters for the equation 6.3. The accuracy of the model can be shown through an example as follows. Let

[10] Note that the dynamic processing load also depends on the SNR and the channel quality.

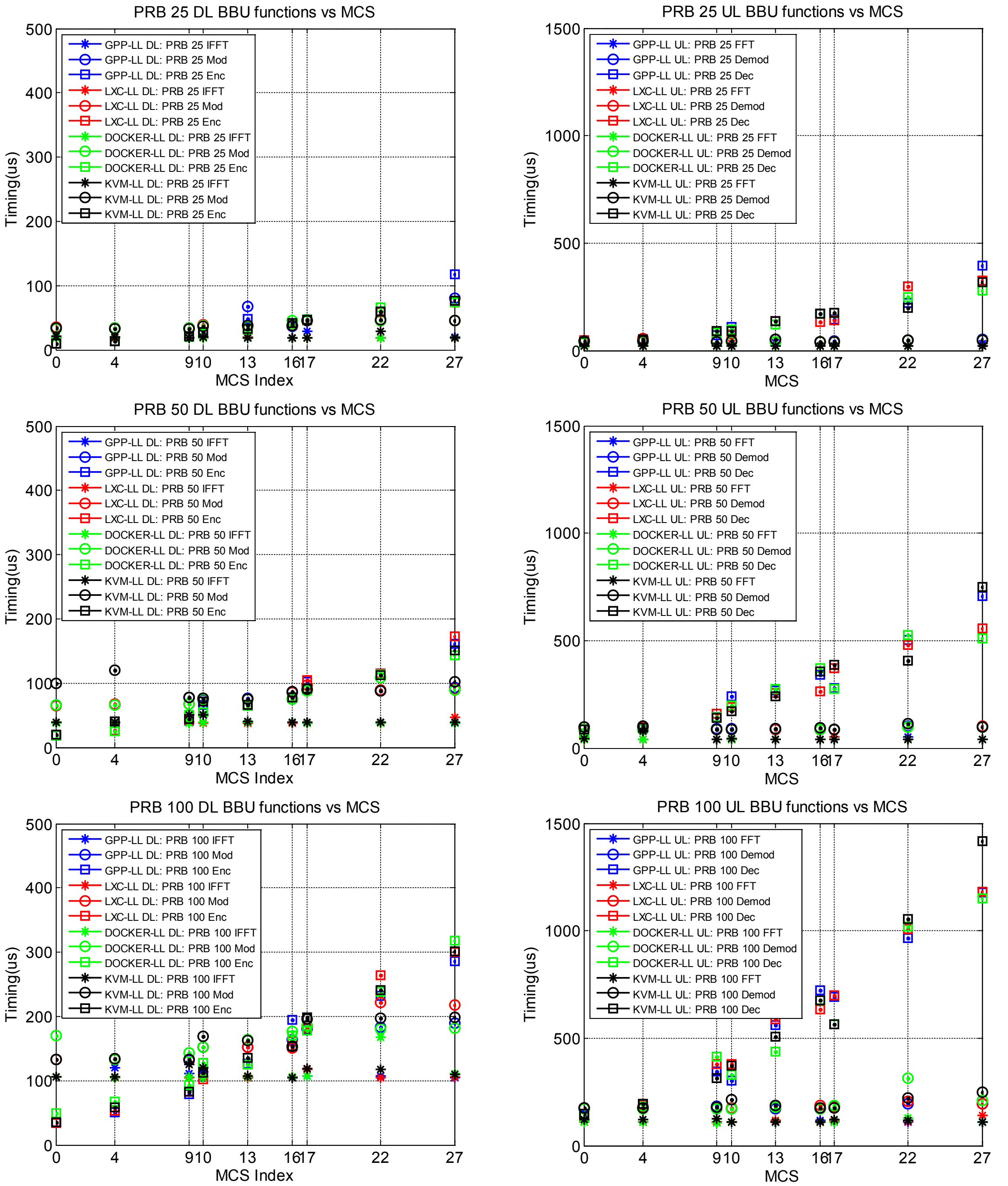
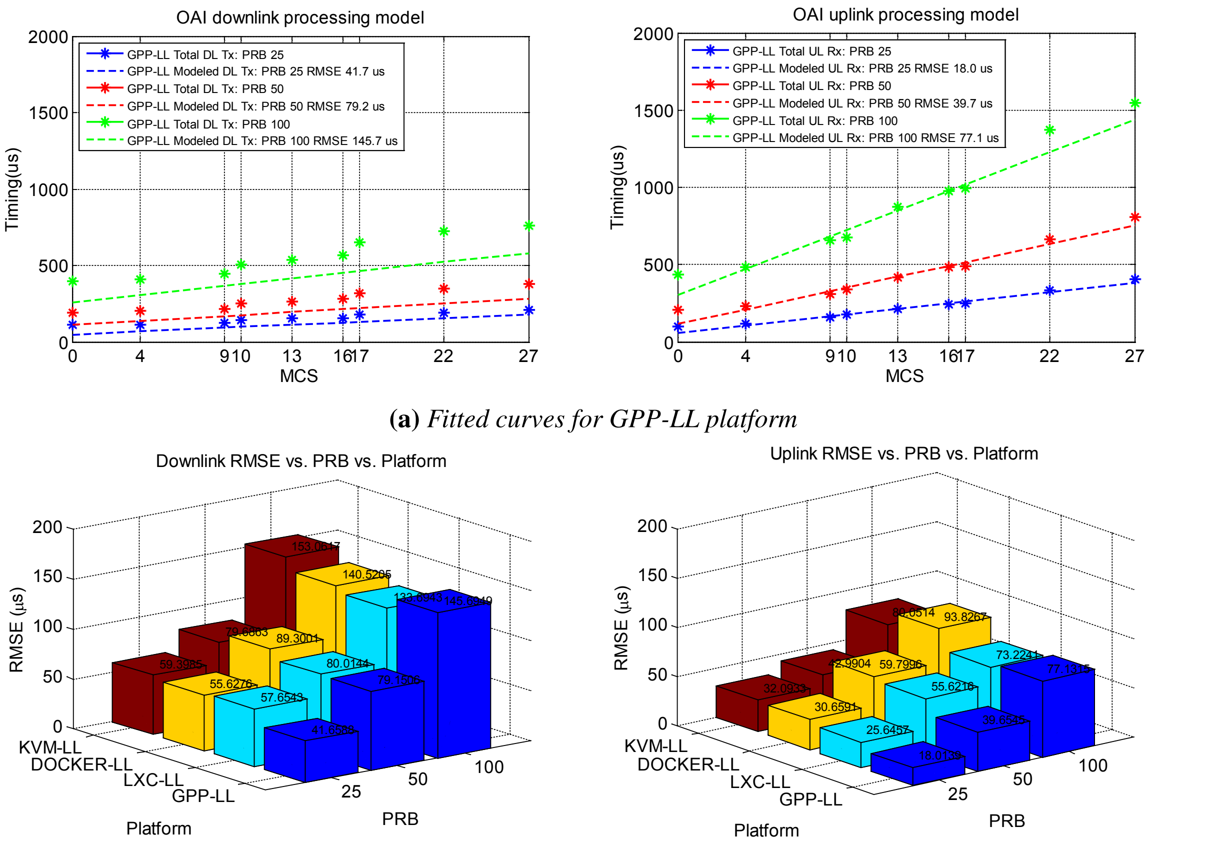


Figure 6.11: *Contribution of iFFT/FFT, (de)modulation, and (de)coding to the total BBU processing for different PRB, MCS, and platforms.*

PRB to be 100, DL MCS 27, UL MCS 16, and platform LXC, the estimated total processing time is 723.5us (111.4+7.4+12*27+147+133.7) against 755.9us in downlink, and 1062.4us (108.8+13.2+41.9*16+196.8+73.2) against 984.9us in uplink.



(a) *Fitted curves for GPP-LL platform*

(b) *RMSE for all platforms*

Figure 6.12: *Modeling BBU processing time.*

Table 6.3: DOWNLINK PROCESSING MODEL PARAMETERS IN US

x	c	p				$u_s(x,y)$		u_c			
		GPP	LCX	DOCKER	KVM	a	b	GPP	LCX	DOCKER	KVM
25	23.81	0	5.2	2.6	3.5	4.9	24.4	41.6	57.6	55.6	59.4
50	41.98	0	5.7	9.7	13	6.3	70	79.2	80	89.3	79.7
100	111.4	0	7.4	13	21.6	12	147	145.7	133.7	140.5	153

Table 6.4: UPLINK PROCESSING MODEL PARAMETERS IN US

x	c	p				$u_s(x,y)$		u_c			
		GPP	LCX	DOCKER	KVM	a	b	GPP	LCX	DOCKER	KVM
25	20.3	0	5.4	4.8	8.8	11.9	39.6	18	25.6	30.6	32
50	40.1	0	6	9.2	15.8	23.5	75.7	39.6	55.6	59.8	42.9
100	108.8	0	13.2	31.6	26.6	41.9	196.8	77.1	73.2	93.8	80

6.8 Conclusion

This chapter investigates three critical issues towards the cloudification of the current LTE/LTE-A radio access network. Extensive set of experiments have been carried out to analyse the BBU processing load under different configurations and environments. The results have shown that the total processing scales up with PRB and MCS and that the uplink is the dominant processing load. Coding and decoding functions represent the most time consuming BBU functions. It is found that container approach to virtualization provides a better performance than hypervisor approach.

Based on the results, a model is presented that accurately estimates the required uplink and down processing as a function of PRB, MCS, and platform.

Chapter 7

Conclusion

The previous chapters have treated particular topics of experimental wireless protocols, algorithms and architectures for which a specific conclusion has been given. This chapter further develops on the selected topics in the perspective of future 5G research.

7.1 Open-Source Tools for 5G Cellular Evolution

Open-source has made a very significant impact in the extremities of current networks, namely in the terminals due to the Android ecosystem and in cloud infrastructure due, in part, to the OpenStack ecosystem [129]. The OAI software platform aims to provide a similar ecosystem for the core (EPC) and access-network (EUTRAN) of 3GPP cellular systems with the possibility of interoperating with closed-source equipment in either portion of the network. In addition to the huge economic success of the open-source model, the platform will be a tremendous tool used by both industry and academia. More importantly it will ensure a much-needed communication mechanism between the two in order to bring academia closer to complex real-world systems which are controlled by major industrial players in the wireless industry. In the context of the evolutionary path towards 5G, there is clearly the need for open-source tools to ensure a common R&D and prototyping framework for rapid proof-of-concept designs.

7.2 RANaaS

The work on radio network cloudification and delivery as a service is getting more and more attention these days with the development of centralized RAN [7]. It is the subject of study in two European projects [107, 108]. RANaaS describes the service lifecycle of an on-demand, elastics, and pay as you go 3GPP RAN on the the top of cloud infrastructure. Thus, lifecycle management is a key for successful adoption and deployment of C-RAN and related services (e.g. MVNO as a Service). It is a process of network design, deployment, resource provisioning, operation and runtime management, and disposal [107].

Following the results presented in Chapter 2 and 6, a proof-of-concept prototype of RANaaS is built in the context of Mobile Cloud Networking project (c.f. Fig. 7.1. The prototype has three main components: a web service, OpenStack, and a Heat stack.[1][2] The web service features a user interface (UI) for the web clients (e.g. MNO, MVNO), a service manager (SO) providing services for the web client, and a service orchestrator (SO) in charge of the RANaaS lifecycle. Openstack includes large pools of computing, storage, and networking resources throughout a datacenter orchestrated by Heat, whose mission is to create a human- and machine-accessible service for managing the entire lifecycle of infrastructure and applications within OpenStack. Heat implements an orchestration engine to manage multiple composite cloud applications described in a form of text-based templates, called Heat stack. Finally, Heat stack is a stack of virtualized LTE network elements with the required networking wrapped up for a particular client.

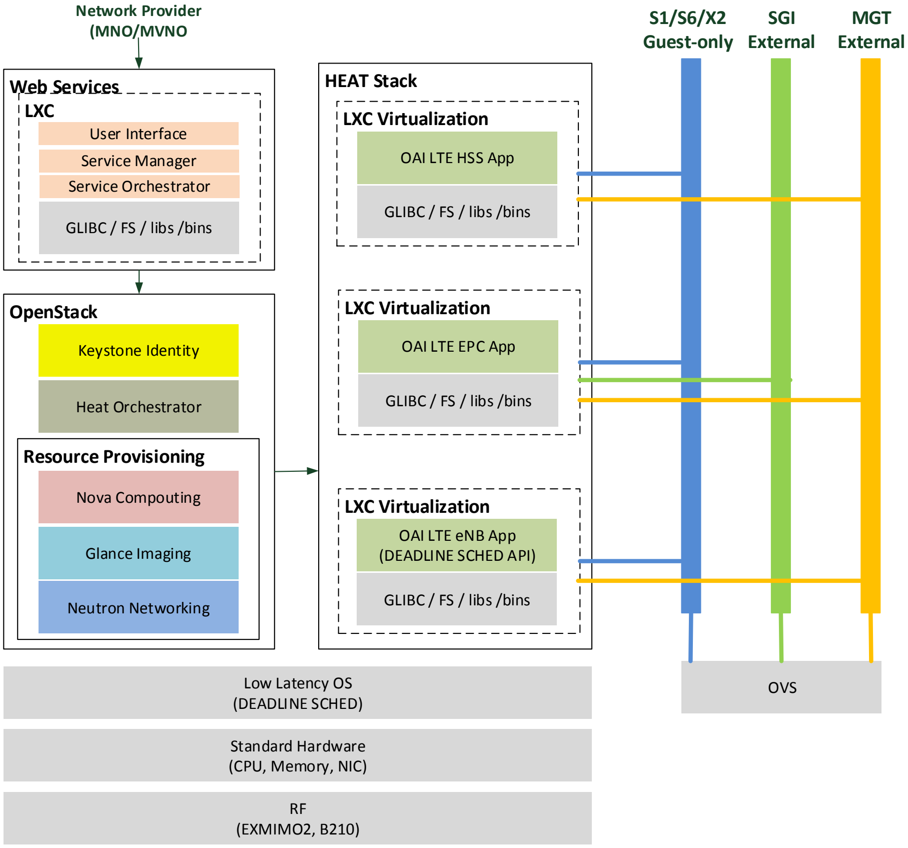


Figure 7.1: *RANaaS prototype.*

Two main avenues of research will be followed. One is about the new opportunities for enhancing the RAN performance in the cloud setting. The research areas of interest include cooperative resource allocation, energy optimization [130], (flow-aware) handover optimization based on X2-like inter-BBU interface, and Ethernet fronthaul architecture. In the latter case, the objective is to build a new (CIPRI, OBSAI, or ORI type) asynchronous interface compliant with IT hardware, with I/Q samples in the Ethernet frames and clock synchronization out of data frames by means of IEEE1588 and/or GPS receiver. This allows to provide a fast and dynamic switching capabilities between BBUs and RRHs. By the usage of OpenFlow switches,

[1] www.openstack.org/
[2] https://wiki.openstack.org/wiki/Heat

load balancing and statistical multiplexing gain can be achieved. The other avenue is related to lifecycle management of RANaaS, looking at the methodology to support service owner (e.g. MNO) to build a complex network topology and configuration.

At the same time a proof of concept prototype based on OpenAirInterface will be enhanced to demonstrate and validate the soundness of these innovative concepts.

7.3 Software-Defined Wireless Networking

In the last years, much attention (both from Industry and Academia) has been placed on software-defined networking (SDN). In one of its most widely acknowledged definition, SDN essentially consists of two elements: (a) a clear separation of control and data plane, and (b) a well-defined interface or abstraction between control and data plane. This decoupling separates the network intelligence from the infrastructure whose complexities are abstracted away from the control applications. Nevertheless, the underlying SDN concepts are not novel per-se, i.e. software has always been used to control the network and control and data plane have been separated in several network architectures (e.g. in optical networks). The fundamental innovation introduced by SDN essentially consists in a vendor- and programming language-agnostic interfaces to networking gear. While SDN indeed reduces part of the intrinsic complexities of the current networking architectures, its actual embodiments pay little attention to the requirements of the radio access domain. As a matter of fact applying the tools currently available in the SDN ecosystem, e.g. OpenFlow, would essentially result in treating mobile terminal as hosts wired to an Ethernet switch. In a radio access network, however, the concept of a “link”, due to the stochastic nature of the wireless medium, is loosely defined, e.g. allocating a flow at a certain BS can affect the available bandwidth at another BS. The definition of new abstraction methodologies is then of capital importance in order to enable true programmability of the RAN. To this end three innovation areas have been identified in control and coordination of 5G networks:

1. **Physical and MAC layer modelling and abstraction:** provides a network view of low-layer reality to allow a scalable and flexible control and coordination framework for complex resource coordination in future cellular networks.
2. **Programmable radio networking interface:** simplifies the management and coordination of of heterogeneous mobile networks based on the low-layer abstraction with well-defined open interfaces and protocols.
3. **Cognitive learning and decision making:** predicts future network and user behaviours and forecasting potential solutions according to the history of the collected data with lowest level of uncertainty. This increases the intelligence of the network protocols and applications and helps automating the entire network resource coordination process and providing an acceptable service quality levels.

Example architecture is presented in Fig. 7.2. It can be seen that the global network graphs representing the physical connectivity are formed at the abstraction layer by extracting parameters from physical and MAC layers for each RAN. Network graphs can be constructed depending on the provider, overlapping or non-overlapping geographical regions or logical zone, and the radio access technology (e.g. WiFi, LTE, HSPA). The abstracted network graphs are built based on partial information of the physical network graph. The level of details depends on the network application requirements and can include information about network topology, connectivity, traffic, and interference. Network application should also learn from the past experience and adapt. Fixed rules may work for a small network, but as the network size increases, evolving rules that can change over time and space have to be applied. The present protocols are not adaptive and have very limited learning capabilities. More advanced cognitive methods could be applied to improve the intelligence of the network protocols. These should adopt more proactive knowledge generation instead of reactive to perceive changes in network state and forecast future network states in real-time.

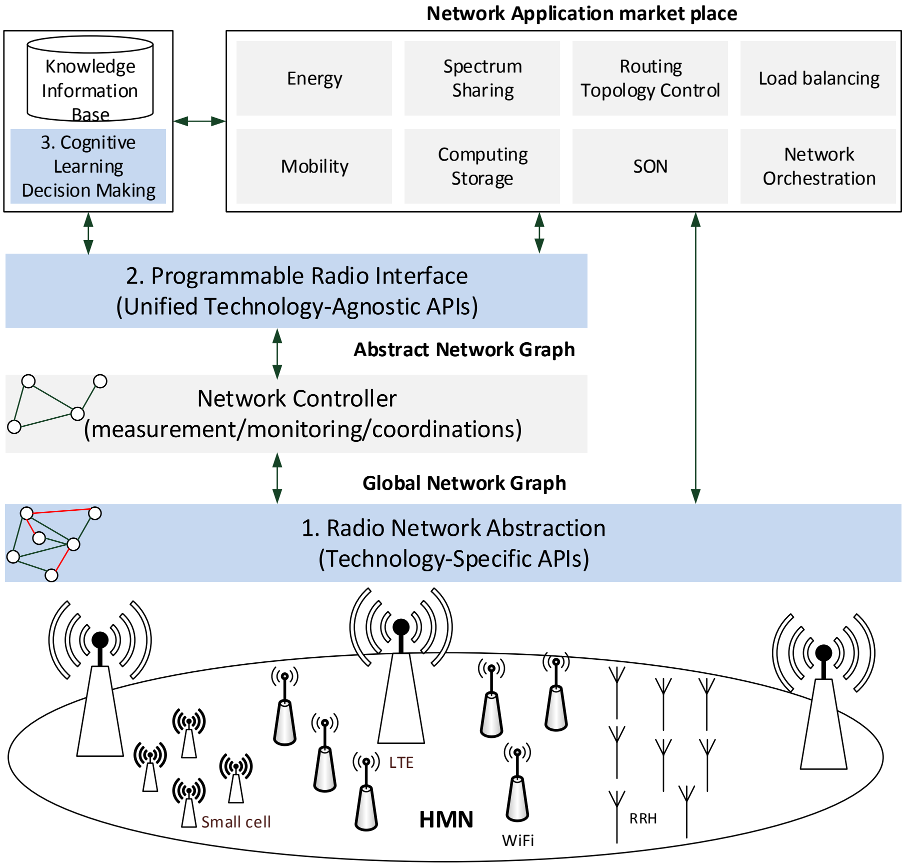

Network Application market place
Knowledge Information Base
3. Cognitive Learning Decision Making
Energy
Spectrum Sharing
Routing Topology Control
Load balancing
Mobility
Computing Storage
SON
Network Orchestration
2. Programmable Radio Interface (Unified Technology-Agnostic APIs)
Abstract Network Graph
Network Controller (measurement/monitoring/coordinations)
Global Network Graph
1. Radio Network Abstraction (Technology-Specific APIs)
LTE
Small cell
HMN
WiFi
RRH

Figure 7.2: *Example SDN architecture with cognition.*

7.4 Mobile Cloud and Edge Computing

Mobile cloud and edge computing provides IT and cloud-computing capabilities within the radio access network, offering personalized services to users, better performance, and better experience, and to businesses more information about consumers, and greater flexibility for provisioning new services [63]. This brings content, services and applications closer to the user increasing responsiveness from the cloud and edge (c.f. Fig. 6.1). Furthermore, radio network are evolving into modular and open environment for the deployment of network applications and services that are able to integrate in an ecosystem of changeable components.

Mobile cloud and edge computing opens the research era on a local and personalized computing, networking, and storage at the mobile edge. This breaks the assumption of the today cellular systems in that it opens the wireless infrastructure to provide carriers, users, and applications control over network traffic. This makes complex network services as pieces of software running on the network and has the potential to open a new market place for the network application (c.f. Fig. 7.2). The following research directions will be pursued, content storage at local cloud, all IP base station, local routing/switching capability at the edge, sensing device agent at local cloud. Relevant example of the latter case can be found in machine-type communication where the required processing on behalf of small devices is done by an agent at the local cloud in order to reduce the packet latency.

7.5 Dynamic Meshing of the Base stations

Chapter 5 presented mobile base station backhauling as one of the main enablers for many use cases found in moving cell, government and public safety scenarios. In the current cellular system, direct base station connectivity is provided through the logical X2 interface. However, planning of the underlying X2 interface may be infeasible or too costly to be established between moving and/or static cells. This calls for a more dynamic meshing and (local) routing among the base stations.

While non-ideal X2 backhauling is under consideration by 3GPP, mobile backhauling remains an open question. Possible research directions in this area include advanced multiple access schemes and new waveforms in support of inter-base station connectivity, and leveraging UE and/or eNB as relays through different component carriers.

Bibliography

[1] I. Latif, F. Kaltenberger, N. Nikaein, and R. Knopp. Large scale system evaluations using phy abstraction for lte with openairinterface. In *Workshop on Emulation Tools, Methodology and Techniques*, 2013.

[2] N. Nikaein and S. Krco. Latency for real-time machine-to-machine communication in lte-based system architecture. In *17th European Wireless Conference, Sustainable Wireless Technologies*, 2011.

[3] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato, and M. Rupp. A comparison between one-way delays in operating hspa and lte networks. In *8th International Workshop on Wireless Network Measurements (WinMee)*, 2012.

[4] Mobile FP7-METIS and wireless communications Enablers for the Twenty-twenty Information Society. Deliverable d2.2 novel radio link concepts and state of the art analysis, 2013.

[5] 5th Generation Non-Orthogonal Waveforms for Asynchronous Signalling FP7-5GNOW. Deliverable d3.1 5g waveform candidate selection, 2013.

[6] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi. Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development*, 2010.

[7] C-Lin I, J. Huang, R. Duan, C. Cui, J. Jiang, and L. Li. Recent progress on c-ran centralization and cloudification. *IEEE Access*, 2014.

[8] B. B. Romdhanne, N. Nikaein, R. Knopp, and C. Bonnet. Openairinterface large-scale wireless emulation platformand methodology. In *roceedings of the 6th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, 2011.

[9] OpenBTS Project. www.openbts.org/.

[10] N. Nikaein, R. Knopp, F. Kaltenberger, L. Gauthier, C. Bonnet, D. Nussbaum, and R. Gaddab. Demo: Openairinterface: an open lte network in a pc. In *Mobicom*, 2014.

[11] A. Bhamri, N. Nikaein, F. Kaltenberger, Jyri J. Hämäläinen, and R. Knopp. Preprocessor for MAC-layer scheduler to efficiently manage buffer in modern wireless networks. In *WCNC 2014, IEEE Wireless Communications and Networking Conference*, 2014.

[12] A. Bhamri, N. Nikaein, F. Kaltenberger, Jyri J. Hämäläinen, and R. Knopp. Three-step iterative scheduler for qos provisioning to users running multiple services in parallel. In *IEEE Vehicular Technology Conference*, 2014.

[13] K. Zhou, N. Nikaein, R. Knopp, and C. Bonnet. Contention based access for machine-type communications over lte. In *Vehicular Technology Conference*, 2012.

[14] K. Zhou, N. Nikaein, and R. Knopp. Dynamic resource allocation for machine-type communications in lte/lte-a with contention-based access. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2013.

[15] FP7-CONECT Cooperative Networking for High Capacity Transport Architectures. Deliverable d4.5 final report on integration of signal and packet level cooperation schemes for efficient video multicasting, June 2013. .

[16] FP7-LOLA, Achieving Low-Latency in Wireless Communications. Deliverable 5.7: Validation Results of WP4 Algorithms on Testbed 3, March 2013. .

[17] A. Apostolaras, N. Nikaein, R. Knopp, A. M. Cipriano, T. Korakis, I. Koutsopoulos, and L. Tassiulas. Evolving ues for collaborative wireless backhauling. In *submitted to SECON*, 2015.

[18] The OpenAirInterface Initiative. http://www.openairinterface.org/.

[19] Navid Nikaein, Raymond Knopp, Florian Kaltenberger, Lionel Gauthier, Christian Bonnet, Dominique Nussbaum, and Riadh Ghaddab. Demo: OpenAirInterface: an open LTE network in a PC. In *MOBICOM 2014, 20th Annual International Conference on Mobile Computing and Networking*, 2014.

[20] N. Nikaein, M. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet. Openairinterface: A flexible platform for 5G research. *ACM Sigcomm Computer Communication Review*, 2014.

[21] K. Tan et al. Sora: High-Performance Software Radio Using General-Purpose Multi-Core Processors. *Communications of the ACM*, 54(1):99–107, Jan 2011.

[22] S. Shearman and J. Kimery. Software Defined Radio Prototyping Platforms Enable a Flexible Approach to Design. *IEEE Microwave Magazine*, 13(5):76–80, Jul 2012.

[23] K. Amiri et al. WARP, a Unified Wireless Network Testbed for Education and Research. In *Proceedings of IEEE MSE*, 2007.

[24] Amarisoft. http://www.amarisoft.com/.

[25] N. D. Nguye, R. Knopp, N. Nikaein, and C. Bonnet. Implementation and validation of multimedia broadcast multicast service for lte/lte-advanced in openairinterface platform. In *International Workshop on Performance and Management of Wireless and Mobile Networks*, 2013.

[26] nwEPC - EPC SAE Gateway. http://sourceforge.net/projects/nwepc/.

[27] M Dickens, Brian P Dunn, and J Nicholas Laneman. Design and implementation of a portable software radio. *IEEE Communications Magazine*, 2008.

[28] Ettus usrp. http://www.ettus.com.

[29] Bilel Ben Romdhanne, Navid, Nikaein, Knopp Raymond, and Bonnet Christian. OpenAirInterface large-scale wireless emulation platform and methodology. In *PM2HW2N 2011, 6th ACM International Workshop on Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks*, 2011.

[30] A. Hafsaoui, N. Nikaein, and L. Wang. Openairinterface traffic generator otg: A realistic traffic generation tool for emerging application scenarios. In *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOT)*, 2012.

[31] Nortel Networks. Ofdm exponential effective sir mapping validation, eesm simulation results. Technical report, 3GPP, 2004.

[32] R.B Santos, Jr. Freitas, C. Walter, E. M. G. Stancanelli, and F. R. P. Cavalcanti. Link-to-System Level Interface Solutions in Multistate Channels for 3GPP LTE Wireless System. *XXV Simposio Brasileiro de Telecomunicacoes*, 2007.

[33] Cognitive Radio Experimentation World. http://www.crew-project.eu/.

[34] Thomas Wirth, V Venkatkumar, Thomas Haustein, Egon Schulz, and Rüdiger Halfmann. Lte-advanced relaying for outdoor range extension. In *Vehicular Technology Conference Fall (VTC 2009-Fall)*. IEEE, 2009.

[35] Eric Blossom. Gnu radio: tools for exploring the radio frequency spectrum. *Linux journal*, 2004.

[36] the usb 3.0 superspeed software defined radio. http://nuand.com/.

[37] HackRF. https://github.com/mossmann/hackrf.

[38] K. Mandke, Soon-Hyeok Choi, Gibeom Kim, R. Grant, R.C. Daniels, Wonsoo Kim, R.W. Heath, and S.M. Nettles. Early results on hydra: A flexible mac/phy multihop testbed. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, 2007.

[39] LTEENB: base station software. http://bellard.org/lte/.

[40] gr-lte. https://github.com/kit-cel/gr-lte.

[41] Abdelbassat MASSOURI and Tanguy RISSET. Fpga-based implementation of multiple phy layers of ieee 802.15. 4 targeting sdr platform.

[42] O. Gustafsson and al. Architectures for cognitive radio testbeds and demonstrators an overview. In *Cognitive Radio Oriented Wireless Networks & Communications (CROWNCOM)*. IEEE, 2010.

[43] Zhe Chen, Nan Guo, and Robert C Qiu. Building a cognitive radio network testbed. In *Proceedings of the IEEE Southeastcon*, 2011.

[44] Alessio Botta, Alberto Dainotti, and Antonio Pescapè. A tool for the generation of realistic network workload for emerging networking scenarios. *Computer Networks*, 56(15), 2012.

[45] F. Capozzi, G. Piro, L.A. Grieco, G. Boggia, and P. Camarda. Downlink packet scheduling in lte cellular networks: Key design issues and a survey. *Communications Surveys Tutorials, IEEE*, 2013.

[46] Y. Zaki, T. Weerawardane, C. Gorg, and A. Timm-Giel. Multi-qos-aware fair scheduling for lte. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, 2011.

[47] S. Chaudhuri, D. Das, and R. Bhaskaran. Study of advanced-opportunistic proportionate fairness scheduler for lte medium access control. In *Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on*, 2011.

[48] 3GPP. Ts 36.212, evolved universal terrestrial radio access: Multiplexing and channel coding (release 10). Technical report.

[49] M. Alasti, B. Neekzad, H. Jie, and R. Vannithamby. Quality of service in wimax and lte networks. *Communications Magazine, IEEE*, 2010.

[50] Cisco. Voice over ip - per call bandwidth consumption. `http://www.cisco.com/image/gif/paws/7934/bwidth_consume.pdf`.

[51] Cisco. Implementing qos solutions for h.323 video conferencing over ip. `http://www.cisco.com/image/gif/paws/21662/video-qos.pdf`.

[52] SVN repository of the OAI prototype implementation of contention-based channel access. `https://svn.eurecom.fr/openair4G/trunk/openair2/LAYER2/MAC`.

[53] N. Nikaein et al. Simple traffic modeling framework for mtc. In *10th International Symposium on Wireless Communication Systems (ISWCS)*, 2013.

[54] Markus Laner, Philipp Svoboda, Navid Nikaein, and Markus Rupp. Traffic models for machine type communications. In *Proceedings of ISWCS*, 2013.

[55] M. Laner, P. Svoboda, and M. Rupp. A benchmark methodology for end-to-end delay of reactive mobile networks. In *IFIP Wireless Days*, 2013.

[56] D. P. Bertsekas and R. Gallager. *Data Networks, 2nd Edition*. Prentice Hall, 1992.

[57] 3GPP. Ts 22.368, service requirements for machine-type-communication (mtc), version 12.4.0, 2014.

[58] 3GPP. Ts 25.913, requirements for evolved utra and evolved utran, version 9.0, 2014.

[59] 3GPP. Lte evolved universal terrestrial radio access network (e-utran); requirements for support of radio resource management, 2013. TS 36.133.

[60] 3GPP. Lte evolved universal terrestrial radio access network (e-utran) radio resource control (rrc), 2013. TS 36.331.

[61] A. Helenius. Performance of handover in long term evolution, 2011. Master Thesis, Aalto University.

[62] GSMA. Prd ir. 34: Inter-service provider ip backbone guidelines (version 4.9). Technical report, 2010.

[63] ETSI ISG MEC. Mobile-edge computing, 2014. White paper, v1.18.

[64] V. Paxson et al. Framework for IP Performance Metrics, http://www.ietf.org/rfc/rfc2330.txt, 1998.

[65] J. Fabini, L. Wallentin, and P. Reichl. The importance of being really random: methodological aspects of IP-layer 2G and 3G network delay assessment. In *ICC'09, Dresden, Germany*, 2009.

[66] F. Baccelli et al. On Optimal Probing for Delay and Loss Measurement. In *IMC'07, San Diego, California*, 2007.

[67] 3GPP. R2-062314, Minutes of the 53rd TSG-RAN WG2 meeting, Shanghai, China, http://www.3gpp.org/ftp/tsg_ran/WG2_RL2/TSGR2_53/Report/.

[68] Achieving Low-Latency in Wireless Communications FP7-LOLA. Deliverable 3.5: Traffic models for m2m and online gaming network traffic, 2012.

[69] Achieving Low-Latency in Wireless Communications FP7-LOLA. Deliverable 3.6: Qos metrics for m2m and online gaming, 2013.

[70] 3GPP. Lte evolved universal terrestrial radio access network (e-utran) physical layer procedure, 2013. TS 36.213.

[71] B. Suard, G. Xu, H. Liu, and T. Kailath. Uplink channel capacity of space-division-multiple-access schemes. *IEEE Transactions on Information Theory*, 1998.

[72] SVN repository of the OAI prototype implementation of contention-based channel access. https://svn.eurecom.fr/openair4G/trunk.

[73] H. S. Park, J. Y. Lee, and B. C. Kim. Tcp performance degradation of in-sequence delivery in lte link layer. *International Journal of Advanced Science and Technology*, 2011.

[74] Stefania Sesia, Issam Toufik, and Matthew Baker. *LTE - The UMTS Long Term Evolution*. John Wiley & Sons, Ltd, 2009.

[75] J.G. Andrews. Seven ways that HetNets are a cellular paradigm shift. *Comm. Magazine, IEEE*, 2013.

[76] A. Damnjanovic, J. Montojo, Yongbin Wei, Tingfang Ji, Tao Luo, M. Vajapeyam, Taesang Yoo, Osok Song, and D. Malladi. A survey on 3GPP heterogeneous networks. *Wireless Comm., IEEE*, 2011.

[77] Sangtae Ha, Soumya Sen, Carlee Joe-Wong, Youngbin Im, and Mung Chiang. TUBE: Time-dependent Pricing for Mobile Data. SIGCOMM, 2012.

[78] Youngbin Im, Carlee Joe-Wong, Sangtae Ha, Soumya Sen, Ted Taekyoung Kwon, and Mung Chiang. Amuse: Empowering users for cost-aware offloading with throughput-delay tradeoffs. In *INFOCOM*, 2013.

[79] S. Sen, C. Joe-Wong, Sangtae Ha, and Mung Chiang. Incentivizing time-shifting of data: a survey of time-dependent pricing for internet access. *Communications Magazine, IEEE*, 2012.

[80] Qixing Wang, Dajie Jiang, Guangyi Liu, and Zhigang Yan. Coordinated multiple points transmission for lte-advanced systems. In *WiCom*, 2009.

[81] 3GPP. Study on coordinated multi-point (comp) operation for lte with non-ideal backhaul, tr 36.874.

[82] 3GPP. Study on small cell enhancements for e-utra and e-utran; higher layer aspects, tr 36.842 v12.0.

[83] Melissa Duarte, Ayan Sengupta, Siddhartha Brahma, Christina Fragouli, and Suhas Diggavi. Quantize-map-forward (QMF) Relaying: An Experimental Study. In *Mobihoc*, 2013.

[84] Yutao Sui, J. Vihriala, A. Papadogiannis, M. Sternad, Wei Yang, and T. Svensson. Moving cells: a promising solution to boost performance for vehicular users. *Comm. Magazine, IEEE*, 2013.

[85] 3GPP. Study on small cell enhancements for e-utra and e-utran, higher layer aspects, tr 36.842 v12.0.0.

[86] T. Doumi, M.F. Dolan, S. Tatesh, A. Casati, G. Tsirtsis, K. Anchan, and D. Flore. LTE for public safety networks. *Comm. Magazine, IEEE*, 2013.

[87] 3GPP. E-utran - x2 data transport, ts 36.424 v11.0.0.

[88] 3GPP. Service requirements for the evolved packet system (eps), ts 22.278, wg sa2.

[89] 3GPP. Relay architectures for e-utra (lte-advanced), tr 36.806 v0.2.0.

[90] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah. When cellular meets wifi in wireless small cell networks. *Comm. Magazine, IEEE*, 2013.

[91] A.M. Cipriano, P. Agostini, A. Blad, and R. Knopp. Cooperative communications with HARQ in a wireless mesh network based on 3GPP LTE. In *EUSIPCO*, 2012.

[92] 3GPP. Physical layer - general description, ts 25.201 v11.1.0.

[93] A. Tyrrell, G. Auer, and C. Bettstetter. Fireflies as Role Models for Synchronization in Ad Hoc Networks. In *Bio-Inspired Models of Network, Information and Computing Systems*, 2006.

[94] Yindi Jing and B. Hassibi. Distributed space-time codes in wireless relay networks. In *Sensor Array and Multichannel Signal Processing Workshop Proc.*, 2004.

[95] SVN repository of the OAI prototype implementation of the proposed architecture. https://svn.eurecom.fr/openair4G/branches/lolamesh.

[96] Shahryar Khan, Jonas Edstam, Balázs Varga, Jonas Rosenberg, John Volkering, and Martin Stümper. The benefits of self-organizing backhaul networks. *Ericsson Review*, 2013.

[97] Oumer Teyeb, Vinh Van Phan, Bernhard Raaf, and Simone Redana. Dynamic Relaying in 3GPP LTE-Advanced Networks. *EURASIP Journal on Wireless Communications and Networking*, 2009.

[98] Sumudu Samarakoon, M. Bennis, W. Saad, and M. Latva-aho. Enabling relaying over heterogeneous backhauls in the uplink of femtocell networks. In *IEEE WiOpt*, 2012.

[99] F. Pantisano, M. Bennis, W. Saad, M. Debbah, and M. Latva-aho. On the impact of heterogeneous backhauls on coordinated multipoint transmission in femtocell networks. In *IEEE ICC*, 2012.

[100] Jian Zhao, T.Q.S. Quek, and Zhongding Lei. Coordinated multipoint transmission with limited backhaul data transfer. *Wireless Comm, IEEE Trans*, 2013.

[101] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i & ii. implementation aspects and performance analysis. *Communications, IEEE Trans. on*, 2003.

[102] Mikio Iwamura, Hideaki Takahasi, and Satoshi Nagata. Relay technology in lte-advanced. *Ongoing Evolution of LTE toward IMT-Advanced, NTT Docomo Journal*.

[103] Steven W. Peters, Ali Y. Panah, Kien T. Truong, and Robert W. Heath. Relay architectures for 3GPP LTE-advanced. *EURASIP J. Wirel. Commun. Netw.*, March 2009.

[104] Chun Nie, Pei Liu, T. Korakis, E. Erkip, and S.S. Panwar. Cooperative relaying in next-generation mobile wimax networks. *Vehicular Technology, IEEE Trans. on*, 2013.

[105] J. Mitola. The software radio architecture. *IEEE Communication Magazine*, 1995.

[106] P. Mell and T. Grance. The nist definition of cloud computing, 2011. NIST special publication.

[107] Mobile Cloud Networking project (FP7-ICT-318109). http://www.mobile-cloud-networking.eu.

[108] iJOIN: an FP7 STREP project co-funded by the European Commission under the ICT theme.

[109] China Mobile Research Institute. C-ran the road towards green ran, 2013. White paper, v3.0.

[110] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud ran for mobile networks - a technology overview. *Communications Surveys Tutorials, IEEE*, 2014.

[111] J.G. Andrews. Seven ways that hetnets are a cellular paradigm shift. *IEE Communications Magazine*, 2013.

[112] S. Samarakoon, M. Bennis, W. Saad, and M. Latva-aho. Backhaul-aware interference management in the uplink of wireless small cell networks. *Wireless Communications, IEEE Transactions on*, 2013.

[113] T. Kolding H. Guan and P. Merz. Discovery of cloud-RAN. *in Cloud-RAN Workshop*, April 2010.

[114] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M.A. Imran, D. Sabella, M.J. Gonzalez, O. Blume, and A. Fehske. How much energy is needed to run a wireless network? *in IEEE Wireless Communications*, 2011.

[115] I. Hwang, B. Song, and S.S. Soliman. A holistic view on hyper-dense heterogeneous and small cell networks. *IEEE Communications Magazine*, 2013.

[116] NGMN. Suggestions on Potential Solutions to C-RAN by NGMN Alliance. Technical report, The Next Generation Mobile Networks (NGMN) Alliance, January 2013.

[117] Bernd Haberland, Fariborz Derakhshan, Heidrun Grob-Lipski, Ralf Klotsche, Werner Rehm, Peter Schefczik, and Michael Soellner. Radio Base Stations in the Cloud. *Bell Labs Technical Journal*, 18(1):129–152, 2013.

[118] NGMN. Further Study on Critical C-RAN Technologies (v0.6). Technical report, The Next Generation Mobile Networks (NGMN) Alliance, 2013.

[119] Scalable and Adaptive Internet Solutions (SAIL). EU FP7 Official Website. *[Online]. Available:http://www.sail-project.eu.*, 2012.

[120] Chih-Lin I, C. Rowell, S. Han, Z. Xu, G. Li, , and Z. Pan. Toward green and soft: A 5g perspective. *IEEE Communications Magazine*, 2014.

[121] Karthikeyan Sundaresan, Mustafa Y. Arslan, Shailendra Singh, Sampath Rangarajan, and Srikanth V. Krishnamurthy. Fluidnet: A flexible cloud-based radio access network for small cells. In *Proceedings of the 19th Annual International Conference on Mobile Computing &#38; Networking*, MobiCom '13, 2013.

[122] M. Sauer, A. Kobyakov, and A. Ng'oma. Radio over fiber for picocellular network architectures. *IEEE Journal on Lightwave Technology*, 2007.

[123] Common radio public interface.

[124] Intel Newsroom. Chip shot: Intel shows off new switch platform at interop, 2012.

[125] Intel White paper. Smart cells revolutionize service delivery, 2013.

[126] ETSI ORI ISG. Open Radio Interface V4.1. Technical report.

[127] Venkatesh Pallipadi and Alexey Starikovskiy. The ondemand governor: past, present and future. *Proceedings of Linux Symposium*, 2006.

[128] I. Alyafawi, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, and N. Nikaein. Critical issues of centralized and cloudified lte fdd radio access networks. In *ICC*, 2015.

[129] OpenAirInterface Software Alliance. Openairinterface flyer, unleashing the potential of open-source in the 5g arena, 2013.

[130] N. Sapountzis, S. Sarantidis, T. Spyropoulos, N. Nikaein, and U. Salim. Reducing the energy consumption of small cell networks subject to qoe constraints. In *IEEE Global Communications Cnference*, 2014.

Appendices

Appendix A

CBA Protocol Validation

This section presents the modifications to the LTE protocol stacks required to enable CBA, which are implemented in the OpenAirInterface platform.

RRC layer is used to configure the CBA RNTIs at both eNB and UE sides. UEs will use these RNTIs to decode the resource allocated for CBA transmission. For simplicity, we set the total number of CBA group to 2 and number of UEs to 3. Listing Figure 19 shows that: (1) the eNB initializes four CBA RNTIs for possible use and the number of active CBA group is 1; (2) the eNB allocates the RNTI (fff4) to the six UEs in one group. Listing A.1 shows after the RRC connection setup, every UE receives a CBA dedicated RNTI. It can be seen that the eNB correctly configure CBA dedicated RNTI for the 2 UEs in a CBA group and that each UE in the CBA group successfully receives the configured CBA dedicated RNTI. We note that the UE 0 belongs to the CBA group 0 with RNTI fff4 while UE 1 belongs to the CBA group 1 with RNTI fff5.

```
[RRC][eNB 0] Initialization of 4 cba_RNTI values (fff4 fff5 fff6 fff7) num
    active groups 2
...
[RRC][eNB 0] Frame 17: cba_RNTI = fff4 in group 0 is attribued to UE 0
[MAC][eNB 0] configure CBA groups 0 with RNTI fff4 for UE  0 (total active
    cba groups 2)
...
[RRC][eNB 0] Frame 36: cba_RNTI = fff5 in group 1 is attribued to UE 1
[MAC][eNB 0] configure CBA groups 1 with RNTI fff5 for UE  1 (total active
    cba groups 2)
...
[PHY][UE 0 ] physicalConfigDedicated pusch CBA config dedicated: beta
    offset 10 cshift 4
[MAC][UE 0] configure CBA group 0 RNTI fff4 for eNB 0 (total active cba
    group 1)
[RRC][UE 0] State = RRC_CONNECTED (eNB 0)
...
[PHY][UE 1 ] physicalConfigDedicated pusch CBA config dedicated: beta
    offset 10 cshift 4
[MAC][UE 1] configure CBA group 0 RNTI fff5 for eNB 0 (total active cba
    group 1)
[RRC][UE 1] State = RRC_CONNECTED (eNB 0)
```

Listing A.1: *RRC layer signalling to initialize and to configure RNTIs for CBA (UE and eNB sides)*

Listing A.2 show the outcome of the resource allocation for two groups with three active UEs, for the round robin allocation policy.

```
[MAC][eNB 0] Frame 60, subframe 2: cba group 0 active_ues 2 total groups 2
   mcs 16, available/required rb (23/11), num resources 1, ncce (0/21)
   required 4
[MAC]add common dci format 0 for rnti fff4
[MAC][eNB 0] Frame 60, subframeP 2: Generated ULSCH DCI for CBA group 0, RB
    25 format 0
[MAC][eNB 0] Frame 60, subframe 2: cba group 1 active_ues 1 total groups 2
   mcs 16, available/required rb (12/11), num resources 1, ncce (4/17)
   required 4
[MAC]add common dci format 0 for rnti fff5
[MAC][eNB 0] Frame 60, subframeP 2: Generated ULSCH DCI for CBA group 1, RB
    25 format 0
```

Listing A.2: *CBA resource allocation*

The resource allocation is sent by eNB using DCI 0 format. With the configured CBA RNTI, a UE can decode the DCI 0 and extract the resource allocation information for CBA. Listing A.3 shows that each UE receives the DCI information for CBA transmission: UE 0 and 2 receives the DCI destined for CBA group 0 with RNTI fff4 while UE 1 receives the DCI destined for CBA group 1 with RNTI fff5.

```
[PHY][UE  0][PUSCH] Frame 199 subframe 8: Found cba rnti fff4, format 0,
   dci_cnt 0
[PHY][UE 0][PUSCH 2] Frame 199, subframe 8 : Programming PUSCH with n_DMRS2
    0 (cshift 0), nb_rb 12, first_rb 1, mcs 16, round 0, rv 0
[PHY][UE  1][PUSCH] Frame 199 subframe 8: Found cba rnti fff5, format 0,
   dci_cnt 1
[PHY][UE 1][PUSCH 2] Frame 199, subframe 8 : Programming PUSCH with n_DMRS2
    6 (cshift 1), nb_rb 10, first_rb 13, mcs 16, round 0, rv 0
[PHY][UE  2][PUSCH] Frame 199 subframe 8: Found cba rnti fff4, format 0,
   dci_cnt 0
[PHY][UE 2][PUSCH 2] Frame 199, subframe 8 : Programming PUSCH with n_DMRS2
    0 (cshift 0), nb_rb 12, first_rb 1, mcs 16, round 0, rv 0
```

Listing A.3: *CBA DCI decoding*

After decoding the resource allocation information with CBA dedicated RNTI, a UE has the CBA transmission opportunity. As the MCS for CBA transmission is not specified by eNB, a UE determines it MCS by itself and transmit this information together with its RNTI using uplink channel information (UCI) to comply with the uplink signaling in LTE, which is originally used to send CQI report. Figure 30 shows that each UE fills the UCI with its C-RNTI and MCS. Listing A.4 shows the CBA transmission opportunity for a UE and UCI preparation.

```
[MAC][UE 0] frameP 201 subframe 2 CBA transmission oppurtunity, tbs 453
[MAC][UE 0] Generate header :bufflen 453  sdu_length_total 105, num_sdus 1,
    sdu_lengths[0] 105, sdu_lcids[0] 3 => payload offset 3,  dcch_header_
   len 0, dtch_header_len 2, padding 0,post_padding 345, bsr len 0, phr len
    0, reminder 345
[PHY][UE 0] ulsch_encoding for eNB 0, harq_pid 4 rnti 6fe1, ACK(1,0)
[PHY]fill uci for cba rnti 6fe1, mcs 16
```

Listing A.4: *CBA transmission opportunity*

As the eNB knows about the CBA resource allocation, it attempts to decode the packets on CBA resource blocks. Listing A.5 shows that the eNB is checking if a CBA packet is sent on the dedicated resource and decoding a CBA transmission from UE 2 with C-RNTI d0d7.

```
[PHY][eNB 0][PUSCH 2] frame 202 subframe 6 Checking PUSCH/ULSCH CBA
   Reception for UE 0 with cba rnti fff4 mode PUSCH
[PHY][eNB 0][PUSCH 2] frame 202 subframe 6 Checking PUSCH/ULSCH CBA
   Reception for UE 1 with cba rnti fff5 mode PUSCH
[PHY][eNB 0][PUSCH 2] frame 202 subframe 6 Checking PUSCH/ULSCH CBA
   Reception for UE 2 with cba rnti fff4 mode PUSCH
[PHY][eNB 0] Found UE with rnti d0d7 => UE_id 2
[PHY][eNB] UCI for CBA : mcs 16  crnti d0d7
[PHY][eNB 0] Frame 202, Subframe 6 : received ULSCH SDU from CBA
   transmission, UE (2,d0d7), CBA (group 0, rnti fff4)
[MAC][eNB 0] Frame 202 : ULSCH -> UL-DTCH, received 120 bytes from UE 2 for
    lcid 3
```

Listing A.5: *CBA reception*

If a collision happens when multiple UEs select the same resource blocks, eNB will allocate resources for the those UEs whose C-RNTI is detected by setting the scheduling request.

```
[PHY][eNB0] Frame 207 subframe 6 : CBA collision detected for UE 3 in group
    1 : set the SR for this UE
```

Listing A.6: *CBA collision*

Appendix B

Virtual Link Validation

In the following figures, we show the log messages form the broadcasting phase to the relaying phase, generated by the OAI system validation platform, to assess the operation of the cooperative MAC. Listing B.1 shows that a bidirectional application traffic from eNB 0 and eNB 1 with the size of 246 and 164 have been received by the PDCP sub-layer, in the broadcasting phase. The radio bearer identity 179 represents the VL#1 for the pair (src 0, dst 1) and (src 1, and dst 0).

```
[APP][I][eNB 0] sending packet from module 0 on rab id 179 (src 0, dst 1)
    pkt size 244;
[PDCP][I]Data request notification with module ID 0 and radio bearer ID 179
     pdu size 246 (header2, trailer0);
...
[APP][I][eNB 1] sending packet from module 1 on rab id 179 (src 1, dst 0)
    pkt size 164;
[PDCP][I]Data request notification with module ID 1 and radio bearer ID 179
     pdu size 166 (header2, trailer0);
```

Listing B.1: *Reception of user application data by the protocol stack for cooperative information forwarding over the VL in broadcast phase*

Listing B.2 depicts that the MAC layer is notified about the data, and generates the sequence number for the upcoming packet, and schedule a collaborative transmission.

```
[MAC][eNB 0] Generated DLSCH header (mcs 15, TBS 285, nb_rb 8)
[MAC][D][eNB 0] adding CO_SEQ_NUM CE with LCID 27 and sn 1
...
[MAC][eNB 1] Generated DLSCH header (mcs 14, TBS 193, nb_rb 6)
[MAC][D][eNB 1] adding CO_SEQ_NUM CE with LCID 27 and sn 1
```

Listing B.2: *MAC scheduling and sequence number for buffer synchronization in the broadcast phase*

Listing B.3 displays the reception of the MAC PDU by the eUEs over the VL with the corresponding ingress and egress IDs from source eNB 0 to destination eNB 1 and from source eNB 1 to destination eNB 0.

```
[MAC][I][eUE 0][VLINK 1] Frame 201 :  i_cornti 960b -> o_cornti ceec, src_
    eNB 0-> dst->eNB 1 (282 bytes)
[MAC][D][eUE 0] Frame 201 mac_buffer_data_ind:  PACKET seq_num 1, pdu_size
    282, proccess_ID 0
```

```
...
[MAC][I][eUE 0][VLINK 1] Frame 201 : i_cornti ceec -> o_cornti 960b, src_
    eNB 1-> dst->eNB 0 (190 bytes)
[MAC][D][eUE 0] Frame 201 mac_buffer_data_ind:  PACKET seq_num 1, pdu_size
    190, proccess_ID 0
```

Listing B.3: *eUE Reception on the VL and MAC PDU buffering in the broadcast phase*

The collaborative buffer status (COBSR) reporting from the eUE to the destination eNB is shown in the listing B.4

```
MAC][D][UE 0/960b/0] updating COBSR0 to (level 20, bytes 190) and COSN0
    (1,1) for eNB 0 (nb element1)
[MAC][D][UE 0][SR 7ffc] Frame 202 subframe 2: send SR to eNB 0 (SR_COUNTER/
    dsr\_TransMax 1/4)
...
[MAC][D][UE 0/ceec/1] updating COBSR0 to (level 23, bytes 282) and COSN0
    (1,1) for eNB 1 (nb element 1)
[MAC][D][UE 0][SR 104b] Frame 202 subframe 2: send SR to eNB 1 (SR_COUNTER
    \/dsr_TransMax 1/4)
```

Listing B.4: *Collaborative buffer status reporting to the destination eNB in the relaying phase*

In the relaying phase, we notice in the listing B.5 that the relay gets the scheduling information through a new downlink control information (DCI) format corresponding to the egress ID indicating with the expected sequence number to be transmitted.

```
[PHY][D][UE  0][PUSCH] Frame 203 subframe 8: Found cornti ceec, format 0A,
Format 0A DCI :ulsch (ue): NBRB        9
Format 0A DCI :ulsch (ue): first_rb    1
Format 0A DCI :ulsch (ue): harq_pid    0
Format 0A DCI :ulsch (ue): Ndi         1
Format 0A DCI :ulsch (ue): TBS         2600
Format 0A DCI :ulsch (ue): mcs         15
Format 0A DCI :ulsch (ue): cshift      0
Format A DCI  :ulsch (ue): sn          1
```

Listing B.5: *Scheduling information received by the eUEs indicating the expected Mac PDU SN in the relaying phase*

Listing B.6 shows that the MAC PDU with the requested sequence number is fetched from the MAC buffer and transmitted on the uplink towards the destination eNB.

```
 [MAC][D][eUE 0] Frame 204  mac_buffer_data_req: MAC PDU with sn 1 for eNB
    index 0 and cornti 960b
[MAC][D][eUE 0][vLINK] Generate ULSCH: buflen 217 MAC PDU size 190
...
[MAC][D][eUE0] Frame 204  mac_buffer_data_req: MAC PDU with sn 1 for eNB
    index 1 and cornti ceec
[MAC][D][eUE 0][vLINK] Generate ULSCH: buflen 325 MAC PDU size 282
```

Listing B.6: *Fetch the PDU with the requested sequence number and transmit if found*

The reception of the collaborative transmission at the destination eNBs corresponding the VL#1 (RAB ID 179) is shown in listing B.7.

```
[PHY][I]Found eUE with CO-rnti 960b => eUE_id 0
[MAC][D][eNB 0] for CORNTI 960b sdu num 0, pass the sdu to rlc lcid 179
    with length 168
[PDCP][I]Data indication notification with module ID 0 and radio bearer ID
    179 rlc sdu size 166
[APP][I][SRC 1][DST 0] RX INFO pkt at time 2043: flag 0x ffff, seq number
    0, tx time 2011, size 164
...
[PHY][I]Found eUE with CO-rnti ceec => eUE_id 0
[MAC][D][eNB 1] for CORNTI ceec sdu num 0, pass the sdu to rlc lcid 179
    with length 248
[PDCP][I]Data indication notification with module ID 1 and radio bearer ID
    179 rlc sdu size 24
[APP][I][SRC 0][DST 1] RX INFO pkt at time 2043: flag 0x ffff, seq number
    0, tx time 2011, size 244
```

Listing B.7: *Destination eNBs decode a transmission over a VL and send the packet to the higher layers*

List of Tables

List of Figures