The OpenAirInterface Application Programming Interface for Schedulers using Carrier Aggregation

Florian Kaltenberger, Cedric Roux
Eurecom, France

Mateusz Buczkowski, Maciej Wewior
IS-Wireless, Poland

***Abstract*—Carrier aggregation (CA) is one of the key features of LTE-Advanced. On one hand it allows to boost the system capacity and on the other hand it enables dynamic spectrum access, such as licensed assisted access (LAA) and licensed shared access (LSA). Efficiently exploiting CA is however not a trivial task; the main challenges are scheduling, load balancing, dynamic carrier activation/deactivation, etc. Due to the complexity of these systems, experimentation is needed to evaluate and test these features. OpenAirInterface is a well know open-source software defined radio solution that allows for experimentation with LTE 4G systems. In this paper we are going present two extensions of the current version of OpenAirInterface: Carrier Aggregation and an application programming interface (API) for schedulers that allows a third party to plug-in their own scheduler. These two extensions allow for more control in the experimentation process than with commercial equipment.**

I. Introduction

The current generation of hardware/software for radio access network (RAN) consist of large numbers of proprietary elements that stifle innovation and increase the cost for the operators to deploy new services/application in an ever-changing fast paced cellular networks. Open source software running on general purpose processors (such as x86, ARM) can greatly simplify network access, reduce cost, increase flexibility, improve innovation speed and accelerate time-to-market for introduction of new services.

OpenAirInterface (OAI) is an open-source standard-compliant implementation of a subset of Release 10 LTE on standard Linux-based computing equipment (Intel x86 PC/ARM architectures). The software is split in two projects with two different open-source licenses: *openairinterface5g* implements the RAN (eNB and UE) and uses the OAI public license (see below) while *openair-cn* implements the EPC (MME, HSS, SGw and PGw) and uses the Apache 2.0 license. OpenAirInterface5G makes use of off-the-shelf software defined radio equipment, such as the Universal Software Radio Peripheral (USRP) from ETTUS/National Instruments, blade RF from nuand, LimeSDR from Lime Microsystems, and ExpressMIMO2 from Eurecom.

In this paper we describe two new features of OpenAirInterface5G: Carrier Aggregation (CA) and the femto application programming interface (FAPI) 2.0 that allows a third party to plug-in their own scheduler. CA is one of the key features of LTE-Advanced. On one hand it allows to boost the system capacity and on the other hand it enables dynamic spectrum access, such as licensed assisted access (LAA) and licensed shared access (LSA). Efficiently exploiting CA is however not a trivial task; the main challenges are scheduling, load balancing, dynamic carrier activation/deactivation, etc. This is where the second extension of OpenAirInterface5G comes in: the FAPI 2.0. Initially developed by the femto forum (now know as the small cell forum) the original FAPI [1] specifies an interface between the MAC and the scheduler of an eNB. In the project SOLDER [2], this interface was extended to support carrier aggregation, and implemented in OpenAirInterface5G. Moreover, a scheduler that makes use of this FAPI 2.0 was developed by IS-Wireless and will be presented in this paper.

A. The OAI public license

While OpenAirInterface and the FAPI 2.0 itself are open-source and freely available on our gitlab server[1], the scheduler that uses the FAPI 2.0 is not required to be open and could be distributed in binary form. This is made possible by the OAI Public License V1.0, which is equivalent to the Apache V2.0 License for non-commercial use. For commercial use, the OAI Public License V1.0 however also includes a clause that allows companies/individuals who own significant portfolio of patents to be able to contribute to the OAI source code and still be able to keep their patent rights. The contributing parties however have to agree to make their patent license available to third parties under fair, reasonable and non-discriminatory (FRAND) terms for commercial exploitation, which are already used by 3GPP member companies. We expect that this license will allow more companies/individuals to contribute to the software, while protecting their essential IPR.

II. OAI carrier aggregation features

Carrier aggregation (CA) has been introduced in LTE-Advanced (Rel. 10) as a means to boost capacity by allowing to aggregate up to 5 component carriers (CC) of 20MHz bandwidth each. However, CA is also a powerful tool for dynamic spectrum access, as component carriers can be enabled and disabled dynamically and also their parameters can be reconfigured on the fly. It is also the basis for the licensed assisted access (LAA) introduced in LTE-Advanced pro (Rel. 13), that allows to aggregate LTE in licensed spectrum with LTE in unlicensed spectrum in the 5GHz ISM band.

This version of the paper is an updated version of the one originally published in the proceedings of the 13th International Symposium on Wireless Communication Systems: Workshop on Spectrum Aggregation and Sharing for 5G Networks (ISWCS'2016–SAS5G Workshop), September 20-23, 2016, Poznan, Poland

[1] https://gitlab.eurecom.fr/oai/openairinterface5g

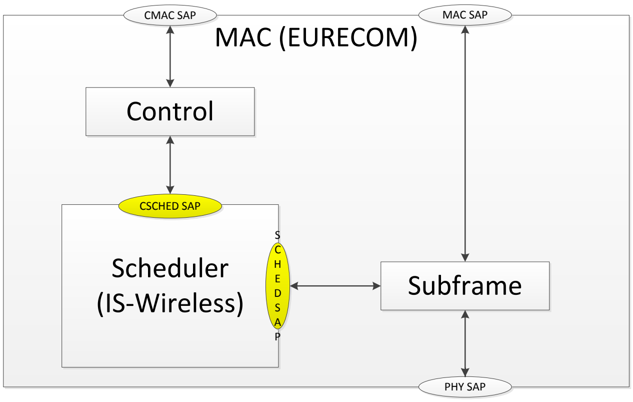

Fig. 1. FAPI 2.0 Interfaces for control plane (CSCHED SAP) and user plane (SCHED SAP).

OpenAirInterface implements Rel. 10 CA with the following features

- FDD, 2 downlink component carriers (CC), 1 uplink CC
- Transmission mode 1 or 2
- UE can connect on either CC (i.e., primary CC and secondary CC are user-specific)
- RRC (re-)configuration of secondary CC
- Dynamic activation and de-activation of secondary CC through MAC signaling

III. FAPI 2.0

Initially developed by the femto forum (now known as the small scell forum) the original FAPI [1] specifies an interface between the MAC and the scheduler of an eNB. In the project SOLDER, this interface was extended to support carrier aggregation, and implemented in OpenAirInterface5G [3].

The interface is defined as a service access point offered by the MAC scheduler to the remaining MAC functionality, as shown in Figure 1. A `_REQ` primitive is from MAC to the MAC scheduler. A `_IND/_CNF` primitives are from the MAC scheduler to the MAC.

The FAPI 2.0 interface is implemented using function calls in one direction and callbacks in the other. The MAC directly calls the `_REQ` primitives. The `_CNF` and `_IND` primitives are callbacks registered when the MAC instantiates the scheduler at startup time. On the MAC side the `_CNF` and `_IND` are also functions. They are called to wait for the corresponding callback to be called by the scheduler. Those functions are called right after the MAC calls the corresponding `_REQ` function. The only exception is for `CSCHED_UE_CONFIG_UPDATE_IND` and `CSCHED_CELL_CONFIG_UPDATE_IND`. Those primitives don't have a corresponding `_REQ` primitive. They are not used in the current implementation. All the synchronization for the MAC side is done in the `_CNF` and `_IND` local functions. A mutex and a cond variable are used to wait for the callback to be called by the scheduler. Callbacks copy the data they receive from the scheduler and then wake up the MAC side which takes this data and serves it to the caller of the function.

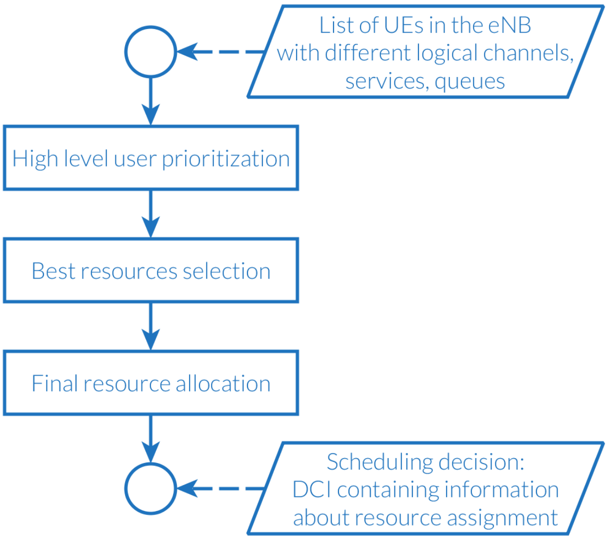
List of UEs in the eNB with different logical channels, services, queues
High level user prioritization
Best resources selection
Final resource allocation
Scheduling decision: DCI containing information about resource assignment

Fig. 2. Main steps of the IS-Wireless scheduling algorithm

Taking for example the downlink scheduling, a typical flow of events is as follows:

1) MAC calls `SCHED_DL_TRIGGER_REQ`. That triggers the scheduler to start a DL scheduling.
2) MAC calls local function `SCHED_DL_CONFIG_IND` where it waits for the callback from scheduler.
3) scheduler is done and calls the callback for `SCHED_DL_CONFIG_IND`, which copies the data and wakes up MAC.
4) MAC is woken up by the callback and returns from local function `SCHED_DL_CONFIG_IND`, passing the data it received from scheduler.

IV. The IS-Wireless scheduler

Using the FAPI 2.0 described above, anyone can develop and plug in their own scheduling algorithms into OpenAirInterface. As an example we present here the LTE eNB Scheduler developed by IS-Wireless [4].

The scheduler is based on the proportional fair principle [5][6], taking into account the channel state and the historical throughput. Moreover it takes into account the quality of service (QoS) class of each logical channel, which defines the guaranteed bit rate, the maximum packet delay, the priority, and the maximum packet error loss. Example of an algorithm that includes packet delay is Largest Weighted Delay First (LWDF) [7] or its modification M-LWDF [8]. Calculation of the priority is enhanced with the delay of head-of-line (HOL) packet. HOL is first packet in the queue waiting to be scheduled. Last but not least, the scheduler takes into account size of the queue. Modified Queue-Based Exponential Rule (M-QBER) is an exemplary algorithm with such functionality. Users having higher variance in queue length in comparison to average value are given more preference [9].

TABLE I. SCHEDULER OVERVIEW

Algorithm Component	Round Robin	Proportional Fair	Advanced
Channel Conditions	No	Yes	Yes
Historical Throughput	No	Yes	Yes
Packet delay	No	No	Yes
Queue length	No	No	Yes
GBR/non-GBR traffic	No	No	Yes
Carrier Aggregation	No	No	Yes
Wrong CQI correction	No	No	Yes
FAPI 2.0 support	Yes	No	Yes

As shown in Figure 2, the scheduler takes three main steps while assigning radio resources to users.

A. High-level user prioritization and preparation

At the beginning list of users connected to eNB is prioritized. This step allows to sort users according to their data rate requirements and retransmission needs. From all users in each TTI scheduler selects those users whose QoS parameter imposes guaranteed bit rate (GBR). Such users, called GBR users, must get sufficient amount of PRBs, which will allow them to reach their target throughput. GBR users are served with higher priority than regular users—it means that scheduling for non-GBR users is performed on the resources that are not allocated after GBR users scheduling.

During this step scheduler also decides if particular UE can benefit from additional CC and whether it is feasible from eNB point of view. First constraint is UE capability—if UE does not support CA, then scheduling algorithm cannot assign additional CC, even if user has lots of data to transmit. If both UE and eNB support CA, scheduler decides about CC assignment in few steps:

1) Scheduler verifies if all UEs, which have more than one carrier active still need these resources—if not, second carrier will be deactivated for that user
2) Scheduler verifies if any of the carriers can serve more users and marks them as available
3) Scheduler searches for users that do not use “available” CC and verifies if they need additional resources—if yes, then second carrier will be activated for that user

This step also includes correction of real-life UE imperfections. UE measures channel conditions using reference signals transmitted in the downlink frame. Based on this measurement UE sends feedback with channel quality indication (CQI). Different UEs might report slightly different CQI values for the same channel conditions, which will cause wrong selection of modulation and coding scheme (MCS) at the scheduler. To deal with this LTE eNB Scheduler monitors number of successful and unsuccessful transmissions and based on that compensates differences in CQI reporting.

B. Best resources selection

Afterwards, for each of the UEs and every PRB temporary priority is calculated based on following formula:

$$\begin{aligned} P_i^{(n)} &= (1 + w_1 \cdot p_{CQI_i}^{(n)}) \cdot (1 + w_2 \cdot p_{delay}^{(n)}) \\ &\cdot (1 + w_3 \cdot p_{historical}^{(n)}) \cdot (1 + w_4 \cdot p_{queue}^{(n)}) \end{aligned} \tag{1}$$

$P_i^{(n)}$	– priority for n-th user for i-th PRB
$p_{CQI_i}^{(n)}$	– n-th user’s channel quality in i-th PRB
$p_{delay}^{(n)}$	– n-th user’s packet delay priority
$p_{historical}^{(n)}$	– n-th user’s historical throughput priority
$p_{queue}^{(n)}$	– n-th user’s queue size priority

Weights $w_1 - w_4$ can be used to increase or decrease importance of specific part of priority. In particular if w_n is set to 0 then n-th component of the priority is not included in priority calculation. Default setting of the weights is 1. It allows to obtain balanced results in terms of achievable throughput, delay and fairness. If operator’s requirements are different it is possible to calibrate algorithm to its needs.

C. Final resource allocation

An important feature of the IS-Wireless (ISW) proprietary scheduler is fact that it is implemented as an eNB-ready scheduler (which means it can be connected to the eNB and scheduling decision is compliant with the standard). It takes into account required parameters and procedures defined by 3GPP in LTE standard such as maximum buffer size, frame structure (reserved slots for other channels), retransmissions, DCI formats and allocation types.

An example of requirement for scheduling decision is that not every combination of PRB is valid. In some cases it is impossible to assign PRBs with highest priorities - based on set of the possible DCI allocation types scheduler must pick best allocation type matching to the best PRBs.

V. RESULTS

Several versions of the scheduler are available for different platforms. For early simulation studies, the scheduler described in the previous section (called “advanced scheduler” henceforth) as well as two well-known scheduling algorithms, namely round-robin (RR) and proportional-fair (PF) have been

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Area size	1.2 km x 1.2 km
Environment	Urban
Number of users	120
Simulation time	50 sec
SINR mapping	MIESM
Number of CC	2
CC Bandwidth	5 MHz
Traffic model	Full buffer

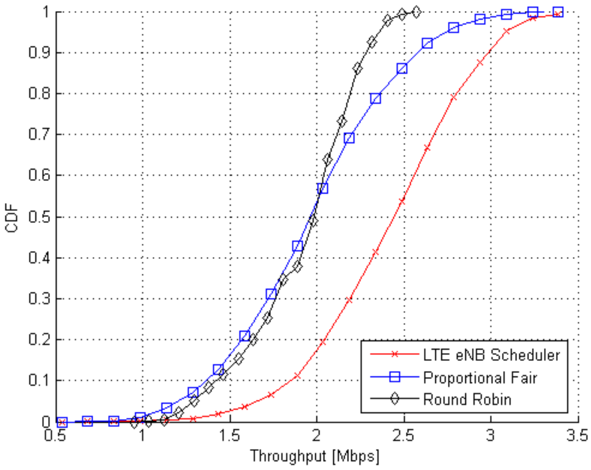

Fig. 3. CDF of Throughput for different schedulers

implemented in Matlab and evaluated using the LTE MAC Lab [2], a system-level simulator capable of simulating Heterogeneous Networks and Carrier Aggregation.

The round robin scheduler and the advanced scheduler have also been implemented in C language using the FAPI 2.0 in order to plug them into the OpenAirInterface eNB. An overview of the different schedulers with their capabilities is given in Table II. In this section we present results that were obtained with all the different versions.

A. Simulation

Simulation parameters are given in Table II and are the same for the different scheduling algorithms. Two main outcomes of the simulations are presented in the paper, namely eNB throughput CDF and peak/cell edge user throughput. Fig. 3 shows distribution of throughput for different scheduling algorithms. The proposed scheduling algorithm offers higher throughput for all users compared to RR and PF. PF is calibrated in similar way to the proposed algorithm and provides better performance for users with good channel conditions without significant degradation of cell edge users. This is also visible in the Fig. 4, where peak (90-percentile) throughput and throughput for cell edge users (10-percentile) are presented. In comparison to RR, PF offers slightly higher peak and edge throughput—this shows that even simple algorithm can utilize the same resources in more efficient way. Using more advanced algorithms, like the proposed scheduler in this case, gives even better results. Not only peak user data rate is more than 35% higher, but also cell edge users get a bit higher throughput.

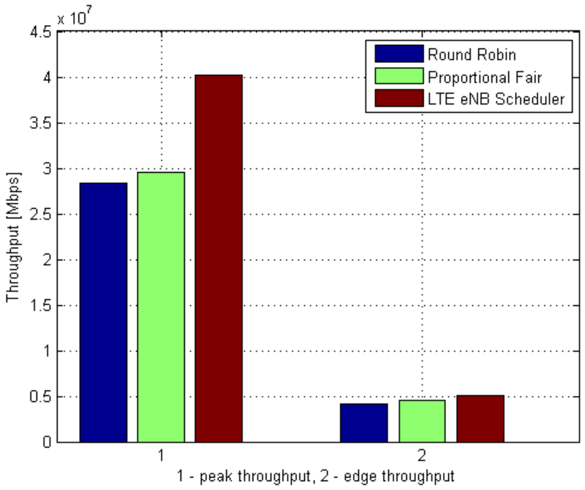

Fig. 4. Peak and Edge Throughput for different schedulers

TABLE III. PARAMETERS FOR THE REAL-TIME TESTS

Parameter	Value
CA configuration	Inter-band, band 4+13
Bandwidth configuration	2 × 5 MHz
Transmission mode	1 (SISO)

B. Real-time

The set-up for the real-time experiments is depicted in Figure 5. It consists of a PC that runs the evolved packet core (EPC) and is connected to the Internet on one side and to an eNB on the other side. The eNB PC uses the ExpressMIMO2 card as a radio transceiver and runs OpenAirInterface together with the IS-Wireless scheduler. Two UEs from Sequans, which are controlled from a laptop, are then connected to the eNB. The basic configuration parameters of the system are given in Table III. For reference, the maximum achievable throughput in this configuration is about 32Mbps in the downlink.

During the trials we first measure the downlink throughput of the two users using the iperf application, which generates UDP traffic at the same PC that runs the EPC. We record one throughput measurement per second and run each trial. In Figure 6 we show results for when both users are in LOS and the offered traffic is 20Mbps per user respectively and the traffic was started with 30sec delay between the users. We can see that the system has no problem serving 15Mbps (approximately half of the total resources) to both users but when the demand is higher than the total channel capacity, buffering occurs. In Figure 7 we show results for the case when one user is in LOS and the other one is in NLOS. The offered traffic is 20Mbps for the LOS user and 10Mbps for the NLOS user. The traffic was started simultaneously. It can be seen that even though the channel is NLOS it is sufficiently good to support 15Mpbs per user. If the demanded traffic is too high, buffering occurs again.

[2] https://www.is-wireless.com/tools/lte-mac-lab/

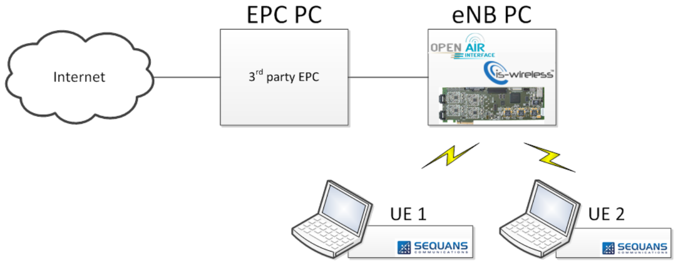

Fig. 5. Real-time test setup.

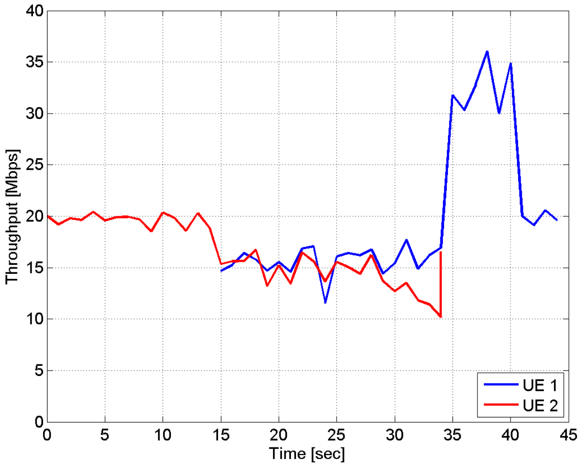

Fig. 6. UDP throughput results for two users with 2 carriers and LOS. The offered traffic is 20Mbps per user respectively and the traffic was started with 30sec delay between the users.

VI. Conclusions and Future Work

We have presented a new framework for experimentation with carrier aggregation based on two extensions to OpenAirInterface: carrier aggregation and the FAPI 2.0. This allows the development and distribution of schedulers on top of the existing OpenAirInterface stack. These extensions could even be closed source thanks to the OAI public license v1.0. We further presented one such example, a new improved scheduler from IS-Wireless, that uses the FAPI 2.0. The real-time tests show proper functioning of the carrier aggregation feature, the FAPI framework and the IS-Wireless scheduler.

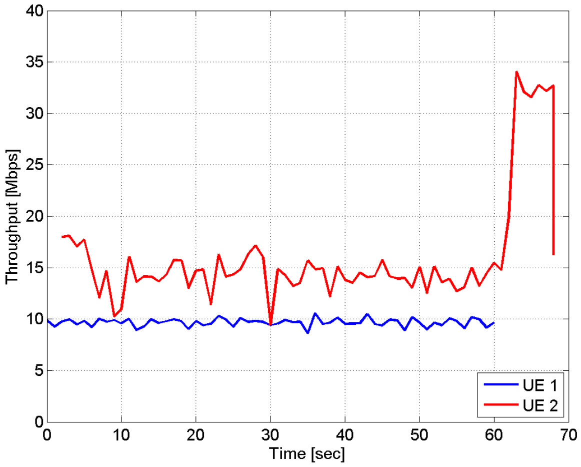

Fig. 7. UDP throughput results for two users with 2 carriers, where one user is in NLOS. The offered traffic is 20Mbps for the LOS user and 10Mbps for the NLOS user. The traffic was started simultaneously.

References

[1] The Femto Forum, “LTE MAC scheduler interface specification v1.11,” Tech. Rep., 2010.

[2] F. Kaltenberger, F. Foukalas, O. D. Holland, S. Pietrzyk, S. Thao, and G. Vivier, “Spectrum overlay through aggregation of heterogeneous dispersed bands,” in *European Conference on Networks and Communications (EuCNC 2014)*, Bologna, Italy, Jun. 2014.

[3] The SOLDER project, “LTE MAC scheduler interface specification v2.0,” http://fapi-2.0.ict-solder.eu, Tech. Rep., 2016.

[4] IS-Wireless, “LTE eNB Scheduler ,” Tech. Rep., 2016. [Online]. Available: https://www.is-wireless.com/small-cells/lte-enb-scheduler/

[5] J. M. Holtzman, “Asymptotic analysis of proportional fair algorithm,” in *Indoor and Mobile Radio Communications*, Sep. 2001.

[6] H. Y. Lee, M. Kang, Y. J. Sang, and K. S. Kim, “The modified proportional fair scheduling algorithms for real-time applications in multiuser multicarrier systems,” in *Military Communications Conference*, Oct. 2009.

[7] A. L. Stolyar and K. Ramanan, “Largest weighted delay first scheduling: Large deviations and optimality,” *The Annals of Applied Probability*, vol. 11, no. 1, pp. 1–48, 2001.

[8] Y. J. Xian, F. C. Tian, C. B. Xu, and Y. Yang, “Analysis of m-lwdf fairness and an enhanced m-lwdf packet scheduling mechanism,” *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 4, pp. 82–88, Aug. 2011.

[9] C. K. Chakravarthy and P. Reddy, “Modified queue-based exponential rule scheduler for improved qos in ofdma systems,” *International journal of UbiComp (IJU)*, vol. 1, no. 2, pp. 34–43, Apr. 2010.