Coordination de l'ordonnancement radio et de calcul dans Cloud-RAN

Mahdi Sharara, Sahar Hoteit, Patrick Brown, Véronique Vèque

Coordination de l'ordonnancement radio et de calcul dans Cloud-RAN

Mahdi Sharara[1], Sahar Hoteit[1], Patrick Brown and Véronique Vèque [1]

[1] *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France*

Cloud-Radio Access Network (Cloud-RAN) est une architecture prometteuse de réseau mobile qui consiste à découpler les unités de bande de base des unités radio et à centraliser le traitement en bande de base de plusieurs Radio Remote Heads (RRH) dans un BBU pool. Dans Cloud-RAN, les RRHs partagent des ressources limitées de calcul, cela rend plus difficile la planification du traitement des données des utilisateurs, en particulier pour les BBU pools surchargés, tout en respectant les contraintes de temps imposées par le mécanisme HARQ. Étant donné que le temps de traitement des données des utilisateurs dépend des paramètres radio tels que l'indice MCS (Modulation Coding Scheme), nous proposons de permettre la coordination entre les ordonnanceurs de ressources radio et de calcul ce qui permet à l'ordonnanceur radio d'ajuster les indices MCS des utilisateurs, pour que l'ordonnanceur de calcul puisse traiter leurs données dans le pool BBU en respectant la date limite imposée par le mécanisme HARQ. Nous proposons deux algorithmes basés sur la programmation linéaire entière ainsi que deux heuristiques de faible complexité. Nous évaluons leurs performances avec différentes métriques. Les résultats révèlent les avantages des solutions de coordination ; notamment en termes de réduction de la puissance gaspillée, et indiquent pour chaque métrique le meilleur algorithme qui pourrait être adopté.

Mots-clefs : Cloud-RAN, 5G, RRH, BBU Pool, Radio and Computing Resources, Scheduling

1 Introduction

Cloud Radio Access Network (Cloud-RAN) is a key pillar in future Mobile Networks, it consists in decoupling Base Band Units (BBUs) from Radio Remote Units (RRUs), and centralizing the baseband processing of many Radio Remote Heads (RRHs) in a shared BBU pool [Mob11]. The latter processes some virtualized functions such as fast Fourier transform, demodulation, decoding, etc. Decoupling baseband processing from radio modules leads to multiple advantages as it reduces CAPEX and OPEX of network operators, increases energy efficiency, and improves user experience [He19]. However, as computing resources of the BBU pool are shared among a large number of RRHs connected to the BBU pool, it is necessary to efficiently manage the BBU pool, especially when it is overloaded, to make sure it maintains the ability to process users' data before passing the deadline imposed by Hybrid Automatic Repeat Request (HARQ) mechanism. As the processing times of users' data strongly depend on radio parameters such as the Modulation Coding Scheme (MCS) index [Ke19][Ke20], the coordination between radio and computing schedulers raises as a candidate to efficiently manage resources of an overloaded BBU pool. In this work, we propose different coordination algorithms between radio and computing schedulers in Cloud-RAN that permit adjusting users' MCS indexes to ensure the processing of their data on the BBU pool, instead of dismissing them in case of overloaded BBU pool. In fact, the processing time of data increases as the MCS index increases [Ke20]. Thus, when the BBU pool gets overloaded, it will not be able to process all users' data while respecting the HARQ-deadline. Users of non-processed data should re-transmit the data and such re-transmission turns out to be energy-inefficient phenomenon that reduces network performance, and wastes radio resources. For that, we propose to employ coordination that allows the radio scheduler to assign users' MCS indexes not only based on the radio quality, but also on the availability of computing resources in the BBU pool. Authors in [Ke19] studied the processing times of BBU up-link functions and showed that the decoding function is the largest consumer of computational resources. Besides, they found that the processing time of decoding function increases with the MCS index and that the uplink processing

time is at least 7 times larger than in downlink. Considering HARQ deadline, authors in [RG17] studied the effect of applying parallelism on the decoding function. Authors in [Ke20] evaluated the performance of two computing scheduling algorithms that aim at increasing the number of correctly decoded sub-frames and the system throughput, respectively. In this context, we investigate two Integer Linear Programming (ILP) coordination solution: Maximize Total Throughput (MTT) and Maximize Users' Satisfaction (MUS). We evaluate their performance with different metrics (total throughput, number of admitted users, fairness, and wasted power). As the ILP problems are NP-Hard, we propose 2 low-complexity heuristics that approximate the performance of ILP solutions.

2 Context and problem formulation

We consider a set of RRHs connected to a BBU pool composed of homogeneous CPU cores. As the uplink processing time is at least 7 times larger than that in downlink [Ke19], it is a dominating issue for the BBU pool's bottleneck. Thus, we focus on the uplink direction where users connected to each RRH share the available resource blocks (RBs). The RRHs transmit users' data to the BBU pool which has to process all the incoming data from the RRHs' users in $2ms$, as instructed by (HARQ) [†] mechanism, and the acknowledgement should be delivered to users in $8ms$ [RG17]. We further consider that user's MCS index is determined by jointly considering the channel conditions of all the RBs in the associated RRH. This permits the radio scheduler to attribute the same MCS index to a given user over all its allocated RBs. It is worth mentioning that a maximum allowed MCS index is attributed by the radio scheduler to a given user by considering its radio conditions measured by user's equipment. More specifically, the Channel Quality Indicator (CQI), which is related to the Signal-to-Noise-and-Interference ratio, is sent by the user equipment (UE), and it carries information on how good/bad the communication channel quality is [Ke19]. Based on CQI, the radio scheduler determines for each user, its maximum allowed MCS index that can be used. As shown in [Ke19], the processing time of the BBU sub-functions (more particularly, the decoding function) strongly depends on the MCS index ; it increases with the increase of MCS index. Hence, if the BBU pool is overloaded, and if all users use their maximum allowed MCS index, the BBU pool will fail to process all the incoming users' data in the specified deadline. Next, we present two ILP coordination solutions, each with a different objective function, and two low-complexity heuristics that serve as alternatives to ILP solutions. Let $\mathcal{R}$ be the set of RRHs, $\mathcal{U}_r$ the set of users of RRH r, $\mathcal{M}$ the set of possible MCS indexes that can be used for the radio transmission, and $\mathcal{C}$ the set of homogeneous CPU cores in the BBU pool. For each RRH r, the coordination policy attributes to user $u \in \mathcal{U}_r$ an MCS index $m \in \mathcal{M}$ lower or equal to the maximum allowed one which was initially chosen by the radio scheduler $M_{r,u,max}$. Based on the chosen index m, user u transmits an amount of data equal to $b_{r,u,m}$ which is determined according to [ETS14] that maps the transport block size (TBS) (i.e., the payload that can be carried by the physical layer) to the MCS index and the number of RBs. Besides, the time required for processing user's u data on the BBU pool is equal to $t_{r,u,m}$ and determined using the simulation results of [Ke19]. We suppose that each user transmits its data with a constant power P_t. The coordination ILP solutions and their heuristic counterparts are:

1. *Maximize Total Throughput (MTT)*: As one of the objectives in 5G networks is to provide a high throughput, this solution tackles this issue by solving the following ILP optimization problem:

$$\text{maximize} \quad \sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}}\sum_{c\in\mathcal{C}} x^c_{r,u,m} b_{r,u,m} \tag{1}$$

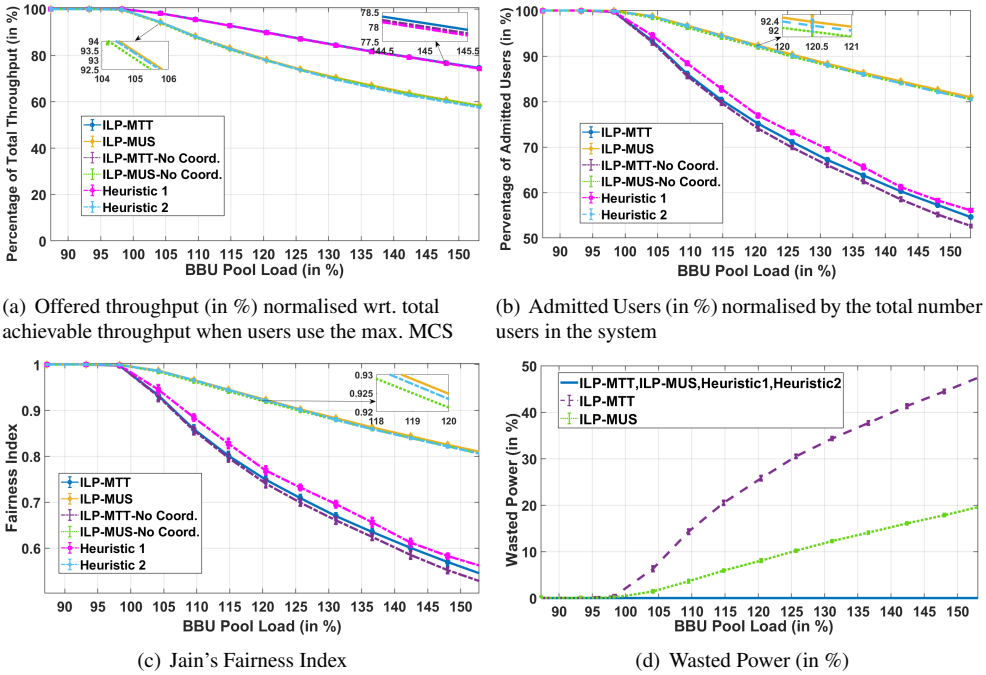
$$\text{subject to} \quad x^c_{r,u,m} \in \{0,1\},\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, m \in \mathcal{M}, c \in \mathcal{C} \tag{2}$$

$$\sum_{c\in\mathcal{C}}\sum_{m\in\mathcal{M}} x^c_{r,u,m} \leq 1,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r \tag{3}$$

$$x^c_{r,u,m} = 0,\ \forall r \in \mathcal{R}, u \in \mathcal{U}_r, c \in \mathcal{C}, m > M_{r,u,max}, \tag{4}$$

$$\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}} x^c_{r,u,m} t_{r,u,m} \leq d, \forall c \in \mathcal{C} \tag{5}$$

†. In HARQ, the data sent from a user need to be transmitted, received, processed, and acknowledged by the BBU, and the sender should receive the acknowledgement in no more than $8ms$. Hence, the deadline for completing the BBU processing of user's data in the uplink is 2ms after deducting the expected latency in fronthaul, transmission, acquirement, etc.



(a) Offered throughput (in %) normalised wrt. total achievable throughput when users use the max. MCS

(b) Admitted Users (in %) normalised by the total number users in the system

(c) Jain's Fairness Index

(d) Wasted Power (in %)

FIGURE 1: Performance evaluation of the different scheduling solutions as a function of BBU pool load

where $x^c_{r,u,m}$ is a binary variable equal to 1 if the data of user $u \in \mathcal{U}_r$ is coded using MCS $m \in \mathcal{M}$ and is processed on CPU core $c \in \mathcal{C}$; and 0 otherwise. The objective function (1) maximizes system's throughput. MTT solution has these constraints: (2) ensures that $x^c_{r,u,m}$ is a binary decision variable; (3) ensures that the data belonging to user $u \in \mathcal{U}_r$ are encoded using at most one MCS index m and are processed on at most one CPU core c; (4) ensures that a user cannot get an MCS index higher than its maximum allowed one and (5) ensures that the data, processed on core c, should finish before the deadline d. Intuitively, MTT favors users with high MCS as they possess higher throughput.

2. *Maximize total Users' Satisfaction (MUS)*: It aims at maximizing total users' satisfaction where user's satisfaction is defined by the ratio of the throughput achieved when operating using a given MCS index to that obtained when using the maximum allowed MCS index. The objective function of MUS is : $\sum_{r\in\mathcal{R}}\sum_{u\in\mathcal{U}_r}\sum_{m\in\mathcal{M}}\sum_{c\in\mathcal{C}} x^c_{r,u,m} \times \frac{b_{r,u,m}}{b_{r,u,max}}$. It assigns for each user an MCS index that does not deviate much from the maximum allowed one. We note that MUS has the same constraints as MTT.
3. *Heuristic 1 - Prioritize High MCS*: This heuristic acts as a low complexity alternative to MTT policy. It applies a 2-level sorting to all users from all RRHs; firstly in descending order of maximum allowed MCS index and then in ascending order of maximum achievable throughput. An adjustment margin variable *AdjMargin* is initialized to 0; it sets a limit on how much user's MCS can deviate from the maximum allowed one. The algorithm loops over the sorted users trying to admit them. After each loop, the algorithm increments *AdjMargin* and loops again over all sorted users. The algorithm stops when all users are allocated, or when the available time is not enough for admitting any other user.
4. *Heuristic 2 - Prioritize Low Throughput*: This heuristic acts as a low complexity alternative to MUS policy. The algorithm acts the same as in Heuristic 1 except in the sorting order; instead of applying a two-level sorting, all users are sorted in ascending order of maximum achievable throughput.

3 Performance Evaluation

We consider a BBU pool composed of 4 CPU cores which has to process the incoming data from the RRHs' users. We vary the number of RRHs connected to the BBU pool from 80 to 140 which in turn varies

the load of the BBU pool from 87% to 153%. Each RRH has 100 RB available for distribution to users, and each user is assigned a random number of resource blocks between 10 and 30. The maximum users' MCS indexes are sampled according to the distribution in [Ke20]. We use the results in [Ke19] based on the Open Air Interface RAN simulator to find the data processing time as a function of MCS index, and number of RBs. Additionally, we refer to the technical specification of ETSI [ETS14] for determining the TBS of a user according to the MCS index and the number of RBs. The throughput is obtained by dividing the TBS by the Transmission Time Interval (TTI) duration. We use MATLAB for the simulation, and CPLEX MILP solver for the ILP problems. To analyze the performance of the coordination policies, we adopt these metrics: *total throughput, number of admitted users, fairness, and wasted power*. We measure the *fairness* using Jain's fairness index [JCH84]: $J_I = \left(\sum_{m\in\mathcal{N}} s_i\right)^2 / (|\mathcal{N}| \times \sum_{m\in\mathcal{N}} s_i^2)$, where $\mathcal{N}$ is the set of users and s_i is the satisfaction of user i (i.e., the ratio of the attained throughput to the maximum achievable throughput). Clearly, a user is most satisfied if it is assigned its maximum allowed MCS index. The *wasted power* metric is defined by the ratio of the power of signals transmitted by UE, but dismissed by the BBU pool to the total emitted power. We limit the study to one TTI and we perform 100 simulations to provide confidence intervals of 95%. We compare the coordination solutions to two other approaches from the literature [Ke20], that do not consider any coordination between radio and computing schedulers. Their objectives are to maximize the throughput and the number of admitted users, respectively. Fig. 1 shows the obtained results for different metrics as a function of BBU pool load. We clearly notice that the different algorithms perform similarly when the BBU is not fully loaded (load$<$100%) because there is enough computing resources for all users without adjusting their MCS indexes. Afterwards, the performance differs among the algorithms. When comparing each of the ILP-coordination (MTT, MUS) to the ILP-no coordination counterparts, we notice: a slight improvement of less than 1% in terms of throughput, up to 2% in the number of admitted users, and up to 0.3 regarding fairness. This slight improvement is a result of the flexibility that the coordination algorithms provide, as they allow to adjust MCS indexes if that helps improving their performance objective. We also notice that the low-complexity heuristics have performance that is very close to the highly-complex NP-Hard ILP counterparts ; so from the operator's perspective these heuristics can serve as good real-time alternatives to the ILP problems which can't output results in real-time. ‡ The high improvement that the coordination brings is in terms of wasted power metric as shown in Fig. 1(d). For both ILP problems with no-coordination, the wasted power increases till it reaches 48% and 20% respectively when the BBU load is 153%. This is because in the no-coordination algorithms, transmission decisions are taken by the radio scheduler alone without even knowing whether the BBU pool will be able to process users' data or not.

4 Conclusion

In this paper, we have evaluated different coordination algorithms between radio and computing schedulers in Cloud-RAN. We considered two objectives that aim at maximizing throughput and users' satisfaction, respectively. Simulation results showed that the coordination brings slight improvement regarding total system throughput, number of admitted users, and fairness, but it significantly reduces wasted transmission power. Moreover, we proposed low-complexity heuristics that perform similarly to the ILP solutions in a much lower run-time. Hence, they can serve as good candidates in practical implementation.

Références

[ETS14] ETSI. LTE ; Evolved Universal Terrestrial Radio Access (E-UTRA) ; Physical layer procedures 3GPP Rel.12, 2014.

[He19] M. A. Habibi and M. Nasimi et al. A comprehensive survey of RAN architectures toward 5g mobile communication system. *IEEE Access*, 7:70371–70421, 2019.

[JCH84] R. Jain, D. Chiu, and W. Hawe. *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems*. DEC Research Report TR-301, Sep 1984.

[Ke19] H. Khedher and S. Hoteit et al. Processing time evaluation and prediction in cloud-ran. In *proc. of ICC*, 2019.

[Ke20] H. Khedher and S. Hoteit et al. Real traffic-aware scheduling of computing resources in cloud-ran. In *proc. of ICNC*, 2020.

[Mob11] China Mobile. C-RAN: the road towards green RAN. *White Paper, ver*, 2:1–10, 2011.

[RG17] V. Rodriguez and F. Guillemin. Towards the deployment of a fully centralized c-ran architecture. In *proc. of IWCMC*, 2017.

‡. We note that the heuristics manage to reduce the run-time (measured in high-level MATLAB) by more than 99%.