Cloud-RAN Resource Scheduling based on real traffic model

Hatem Khedher, Sahar Hoteit, Patrick Brown, Véronique Vèque, Ruby Krishnaswamy, William Diego, Makhlouf Hadji

Cloud-RAN Resource Scheduling based on real traffic model

Hatem Khedher[1], Sahar Hoteit[1], Patrick Brown, Véronique Vèque [1], Ruby Krishnaswamy[2], William Diego[2], Makhlouf Hadji [3]

[1] *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France. (hatem.ibnkheder@l2s.centralesupelec.fr, sahar.hoteit@u-psud.fr, veronique.veque@u-psud.fr)*
[2]*Orange Labs, Chatillon, France (ruby.krishnaswamy@orange.com, william.diego@orange.com)*
[3] *Institut de Recherche Technologique SystemX, 8 Avenue de la Vauve, 91120 Palaiseau (makhlouf.hadji@irt-systemx.fr)*

Cloud-Radio Access Network is a promising mobile network architecture that centralizes the computing resources in the Baseband Unit pool which adds more flexibility and increases network performance. However, as computing resources are shared among the Radio Remote Heads (RRH) connected to the Baseband Unit pool, efficient scheduling algorithms should be explored to meet the deadlines requirements of RRHs' subframes and to increase the network throughput. In this paper, we propose optimal scheduling algorithms for computing resources along with three heuristics. We test the different algorithms as a function of three metrics. The evaluation is performed under a real traffic model and the results highlight the importance of choosing the appropriate scheduling algorithm to increase network performance.

Mots-clefs : Cloud-Radio Acess Network, radio parameters, BBU processing time.

1 Introduction

Cloud Radio Access Network (C-RAN) is a promising paradigm which consists in centralizing certain Base Band Unit (BBU) processing functions in order to add more flexibility and to increase the overall network performance [Mob11]. However, as resources in Cloud-RAN are shared and virtualized, the problem of allocating them to users becomes more complex and challenging [LAL16]. In this paper, we focus mainly on scheduling the Radio Remote Heads (RRHs) subframes on the computing resources (i.e., a set of CPU cores in the shared BBU pool), taking into account the real time constraints of the subframes and considering a real traffic model. In particular, we propose two optimal scheduling algorithms namely OSA 1 and OSA 2 that are based on a linear programming model for deciding the optimal locations to place RRHs' subframes on the CPU cores of the shared BBU pool. Besides, we propose three heuristics and we show, using a real traffic model and for each performance metric, the one that is closer to the optimal solution. We give straightforward recommendations for mobile network operators on the best scheduling algorithm that should be used in order to improve user experience and network performance. These recommendations were somehow missing in the literature. For instance, authors in [RG17] study the problem of optimal C-RAN design and dimensioning. They propose a novel BBU execution strategy using parallel programming techniques on channel decoding BBU-functions in order to minimize the runtime of BBU functions. Despite the relevance of their work, the proposed models do not consider real traffic fluctuations and radio parameters.

Moreover, in [GFS16], authors propose a C-RAN scheduling algorithm, namely RT-OPEX that chooses a partitioned scheduler with 2 CPU cores per RRH and schedules even and odd subframes on these cores in a round robin fashion. Real traffic modeling is missing in their proposal. Moreover, optimal scheduling of computing resources is not considered. In [BPCKJ$^+$], the authors try to reduce the run-time of RAN functions. They propose a framework that splits the set of BBUs into groups that are simultaneously processed on a shared computing platform. They propose a static round robin scheduling algorithm to schedule the set of base stations and to meet their real-time processing requirements. Optimal modeling of the scheduling problem is not provided in their work. We refer the reader to [KHB$^+$19] and [KHB$^+$20] for a more elaborated version of the state of the art. In this paper, we propose an exact algorithm that can quantify the scheduling cost in terms of novel performance indicators taking into account real traffic models.

2 Context and problem formulation

Consider a BBU pool composed of $\mathcal{C}$ homogeneous CPU cores that execute the BBU functions of the $\mathcal{N}$ RRHs. Each RRH sends a subframe every Transmission Time Interval (TTI). Each CPU core can process at most one subframe that can not be split among different cores. Our objective is to derive the most efficient CPU cores allocation for the subframes using three performance criteria : the number of correctly processed subframes, the throughput of correctly processed subframes and the amount of residual CPU core times. The RRHs are operating using the 20 MHz bandwidth so that the number of physical resource blocks per subframe is equal to 100. The processing time of the subframe $i \in \mathcal{N}$ is denoted by t_i and is determined based on its modulation coding scheme (MCS) index [KHB+19]. Each subframe is characterized by its length (in bytes) b_i that is also determined according to its MCS index [KHB+19]. The subframes have a processing time deadline d that should not be missed, otherwise they have to be re-scheduled.

2.1 Optimal Scheduling Algorithm (OSA) for computing resources in C-RAN

We formulate the scheduling problem of computing resources (CPU cores) in C-RAN using an Integer Linear Programming (ILP) optimization technique. We consider two different objectives : the first one consists in maximizing the total number of correctly decoded subframes and the second maximizes the throughput. The first proposed centralized optimization problem (OSA 1) is defined as follows :

$$\text{maximize} \quad \sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}} x_i^c \tag{1}$$

$$\text{subject to} \quad x_i^c \in \{0,1\},\ \forall i \in \mathcal{N}, c \in \mathcal{C} \tag{2}$$

$$\sum_{c\in\mathcal{C}} x_i^c \leq 1,\ \forall i \in \mathcal{N} \tag{3}$$

$$\sum_{i\in\mathcal{N}} x_i^c t_i \leq d,\ \forall c \in \mathcal{C} \tag{4}$$

where x_i^c is a single binary variable for the subframe scheduling decision. It is defined as follows :

$$x_i^c = \begin{cases} 1 & \text{if the subframe } i \in \mathcal{N} \text{ is assigned} \\ & \text{to the CPU core } c \in \mathcal{C} \\ 0 & \text{Otherwise} \end{cases}$$

The proposed objective function in *OSA 1* (Eq. (1)) maximizes the number of correctly decoded subframes. It reflects the efficiency of a BBU pool, measured by its ability to host a large number of subframes. Our optimal scheduling algorithm has the following constraints :

— Non-negative decision variable constraint (Eq. 2) : the decision variable x_i^c should always be positive.
— Single core constraint (Eq. 3) : each subframe should be assigned to at most one CPU core c.
— Deadline constraint (Eq 4) : each core c must be able to finish the processing of the subframe assigned to it before the deadline d.

Moreover, as one of the major objectives in 5G networks is to guarantee a high throughput to mobile users, a slight modification in the previous objective function can tackle this issue while keeping exactly the same constraints as in *OSA 1*. The corresponding problem (OSA 2) maximizes the total throughput; its objective function is given by : maximize $\sum_{i\in\mathcal{N}}\sum_{c\in\mathcal{C}} x_i^c b_i$, where the throughput of each subframe i is the ratio between the subframe's length b_i and TTI duration.

2.2 Proposed heuristics for scheduling computing resources

Instead of using ILP programming for scheduling resources in Cloud-RAN, it is important to determine the heuristics that are compatible with the real time constraints in C-RAN. We propose three heuristics :

— Round Robin (RR) : this heuristic consists in scheduling the incoming subframes from the RRHs to the CPU cores of the BBU pool in a round robin fashion (e.g., cyclically).
— Shortest Time First (STF) : it consists of sorting, in increasing order, the incoming subframes to the BBU pool according to their processing time requirement then applying the round robin heuristic.
— Highest Throughput First (HTF) : this policy first sorts the subframes according to their throughput in a decreasing order, then schedules the sorted list to the CPU cores in a round robin fashion.

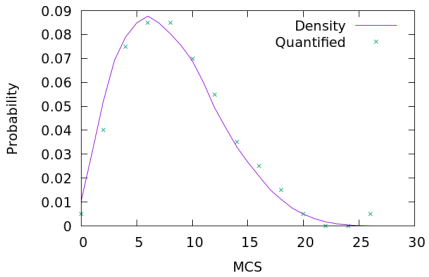


FIGURE 1: Probability density of MCS indexes

3 Performance Evaluation

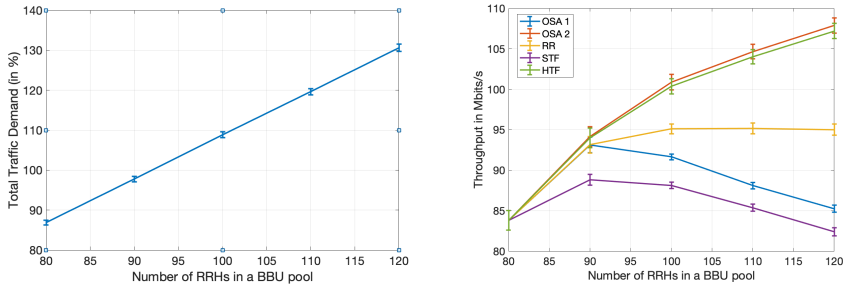
The performance of resource allocation algorithms strongly depends on the traffic pattern. For that, we use a real traffic model based on real measurements [TBWD17], the resulting probability density function is represented in Figure 1. We consider a BBU pool of 4 CPU cores that has to process the uplink subframes of the connected RRHs. The number of RRHs varies between 80 to 120. Fig. 2(a) provides the load expressed by the total traffic demand as a function of the number of RRHs. We clearly focus on scenarios where the load varies between 85% to 130%. We note that a number of RRHs lower than 90 expresses the case of under-load of the CPU cores in the BBU pool ; a number of RRHs between 90 to 100 refers to the case of slightly overload of the CPU cores and finally a RRH number higher than 100 refers to highly overloaded scenarios. In each TTI, the subframe of an RRH is characterized by a randomly drawn MCS index obtained using the distribution presented in Fig. 1. The processing time of each subframe is determined using the measurements in [KHB+19] and the deadline of each subframe is set to 2 ms. We compare the performance of *OSA 1* and *OSA 2* as well as those of the three heuristics *RR*, *STF* and *HTF* as a function of the following performance criteria :

— The **total traffic throughput** that could be decoded before the deadline.
— The **number of subframes** that could **not** be **decoded** in the 2 ms deadline.
— The **CPU cores occupancy** : the total amount of residual CPU core times that could not be used due to the fact that either all subframes were processed or the processing times of the remaining subframes would exceed the deadline.

Fig. 2(b) shows the throughput offered by the different scheduling algorithms . We notice that OSA 2 (i.e., the optimal scheduling algorithm with the objective of maximizing the throughput) has the best performance. Among the three heuristics, the HTF policy is very efficient ; it shows a very close performance to OSA 2. The STF technique that favors the shortest subframes in time (i.e., those having the lowest MCS and hence the lowest throughput) shows the worst performance. Finally, RR policy comes in-between the two other heuristics. Results clearly show that among the three heuristics, HTF policy could be adopted by network operators to maximize the throughput as it shows close performance to OSA 2. Fig. 2(c) shows the results of the second criteria (i.e., the number of incorrectly decodes subframes). We notice that OSA 1 has the best performance and among the three heuristics, the STF is the one that is closer to OSA 1. This is due to the fact that STF gives the highest priority to small subframes (i.e., those having the shortest processing time and smallest MCS index). According to this performance criterion, HTF is not an appropriate technique. We remark also that RR comes in-between the HTF and STF approaches. Fig. 2(d) shows the results of the third criteria (i.e., the time during which the CPU cores are idle). We notice that OSA2 has the best performance. Among the heuristics, HTF outperforms the others. As HTF starts by serving subframes with longest processing times, those with short processing times are left as we get closer to the deadline. These processing times are easier to fit in within the deadline, leaving less unused CPU. As before, RR comes always in-between the other two heuristics. We remark also that OSA techniques converge quickly to zero which proves the efficiency of the exact methods. We note also that STF has the worst performance according to this metric as it starts by serving subframes with the smallest processing times, leaving only subframes with long processing times when we get closer to the deadline.

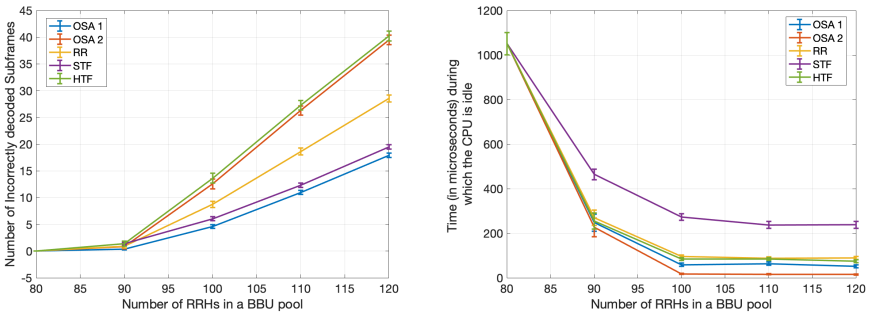
4 Conclusion

This paper presents real traffic based scheduling algorithms for computing resources in Cloud-RAN. We integrate new real-time constraints such as subframe processing times and CPU cores deadlines to our optimization problems. Moreover, for reasons of scalability, three heuristics are proposed. We evaluate the proposed algorithms for several number of RRHs and for different performance metrics and we show for each one, the heuristic that is closer to the optimal solution which brings recommendations to mobile



(a) Total traffic demand as a function of the number of RRHs assigned to a BBU pool

(b) Offered throughput as a function of the number of RRHs assigned to a BBU pool



(c) Number of undecoded subframes as a function of the number of RRHs in the BBU pool

(d) Time during which the CPU cores in BBU pool are unused as a function of the number of RRHs

FIGURE 2: Performance evaluation of the different scheduling algorithms

network operators on the best scheduling algorithm that should be adopted to increase network performance.

Acknowledgment

This research work has been carried out in the framework of IRT SystemX, Paris-Saclay, France, and therefore granted with public funds within the scope of the French Program Investissements d'Avenir.

Références

[BPCKJ+] S. Bhaumik, S. Preeth Chandrabose, M. Kashyap Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo. Cloudiq : A framework for processing base stations in a data center, in proceedings of mobicom 2012.

[GFS16] K.C. Garikipati, K. Fawaz, and K. G. Shin. Rt-opex : Flexible scheduling for cloud-ran processing. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, NY, USA, 2016.

[KHB+19] H. Kheder, S. Hoteit, P. Brown, R. Krishnaswamy, W. Diego, and V. Vèque. Processing time evaluation and prediction in cloud-ran. In *Proceedings of International Conference on Communications ICC*, 2019.

[KHB+20] H. Khedher, S. Hoteit, P. Brown, V. Véque, R. Krishnaswamy, W. Diego, and M. Hadji. Real traffic-aware scheduling of computing resources in cloud-ran. In *International Conference on Computing, Networking and Communications (ICNC)*, 2020.

[LAL16] M. Y. Lyazidi, N. Aitsaadi, and R. Langar. Dynamic resource allocation for cloud-ran in lte with real-time bbu/rrh assignment. In *International Conference on Communications*, 2016.

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

[RG17] V. Q. Rodriguez and F. Guillemin. Towards the deployment of a fully centralized cloud-ran architecture. In *International Wireless Comm. and Mobile Computing Conference*, 2017.

[TBWD17] H. D. Trinh, N. Bui, L. Widmer, J.and Giupponi, and P. Dini. Analysis and modeling of mobile traffic using real traces. In *IEEE PIMRC*, 2017.