Massive MIMO Pilot Scheduling over Cloud RAN

Peng, Haorui; Tärneberg, William; Fitzgerald, Emma; Kihl, Maria

2020

Massive MIMO Pilot Scheduling over Cloud RAN

Haorui Peng*, William Tärneberg*, Emma Fitzgerald*†, Maria Kihl*
*Department of Electrical and Information Technology, Lund University, Lund, Sweden
†Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
{haorui.peng, william.tarneberg, emma.fitzgerald, maria.kihl}@eit.lth.se

***Abstract*—Cloud-RAN (C-RAN) is a promising paradigm for the next generation radio access network infrastructure, which offers centralized and coordinated base-band signal processing. On the other hand, this requires extremely low latency fronthaul links to achieve real-time centralized signal processing. In this paper, we investigate massive MIMO pilot scheduling in a C-RAN infrastructure. Three commonly used scheduling policies are investigated with simulations in order to provide insight on how the scheduling performance is affected by the latency incurred by the C-RAN infrastructure.**

***Index Terms*—Cloud-RAN, Massive MIMO, latency Constraint fronthaul, MAC scheduling**

I. Introduction

Cloud-RAN (C-RAN) is a competitive candidate Radio Access Network (RAN) architecture for the fifth generation of mobile communication networks. It is envisioned to support softwarization and resource centralization in radio access networks and promises to provide mobile Internet access with low cost and high efficient network operations.

The basic concept of C-RAN is to detach the Base-Band processing Unit (BBU) from multiple legacy radio base stations and centralize them into a BBU pool. The remaining Remote Radio Heads (RRH) are only equipped with basic radio-frequency functionalities like transmitting, receiving and analog/digital convention. The BBU pool allows base-band signal processing in a cooperative way for multiple RRHs beyond sites.

However, as of now various challenges remain to be solved in order to deploy the C-RAN infrastructure for the next generation mobile networks [1], [2]. One important challenge is to establish the fronthaul links that enable the communication between BBU pool and RRHs. These fronthaul links must comply with the stringent bandwidth and latency requirements for C-RAN. Ethernet, because of its high flexibility and cost-efficiency, has been considered as an attractive solution for the fronthaul links [3], [4].

On the other hand, massive Multiple Input Multiple Output (MIMO) is another essential enabler for the next generation RAN that significantly increases the system capacity to handle the rapid growth of traffic in mobile networks. However, this large scale antenna system requires a huge amount of computational power for base-band signal processing. Therefore, it would be beneficial to adopt massive MIMO to C-RAN architecture and to split part of processing functionalities to a remote BBU pool. However, offloading the computational resources of such large antenna systems to a remote BBU pool implies that it may suffer from the capacity and latency limitations of the fronthaul [5].

In [6], [7], the functionality split in massive MIMO RRH C-RAN system are addressed to tackle the bandwidth fronthaul limitation. Instead of offloading the whole base-band function chain to the BBU, the authors keep part of the function blocks in the RRH ans allow them to be processed locally.

Other solutions to the limited-fronthaul in massive MIMO C-RAN system are investigated as well. A-prefiltering C-RAN architecture is proposed in [8] to compress the link data rate over the fronthaul and to keep the RRH structure as thin as possible. In [9], pilot contamination and imperfect channel estimation are considered as the impacts of the limited fronthaul.

In [10], the authors employed a decision-theoretic framework to tackle the issue of outdated Channel State Information (CSI) caused by the delay in a C-RAN and mobile cloud computing system from the perspective of channel estimation.

To the best of our knowledge, in regard to the research on massive MIMO with C-RAN, pilots scheduling problem has not drawn much attention. Likewise, few have considered the latency as the main constraint in the fronthaul for their problems, however under which the scheduling performances and user experiences are significantly affected.

In our work, we target a C-RAN system in which a BBU pool and a massive MIMO RRH are connected by a fronthaul link that would introduce relatively long delay to the system. In this paper, we describe how a pilot scheduling function on Medium Access Control Layer (MAC) layer of massive MIMO is affected as it is implemented in the BBU pool of the addressed C-RAN system and has a long delay to the RRH. We apply a scheduling strategy with three well-known scheduling policies in the BBU to improve the system performance. In the end, we implement a simulation to provide the insights on the performances of different schedulers as the latency in the C-RAN system increases.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Maria Kihl and Emma Fitzgerald are also partially supported by the SEC4FACTORY project, funded by the Swedish Foundation for Strategic Research (SSF), and the 5G-PERFECTA Celtic Next project funded by Sweden's Innovation Agency (VINNOVA). Maria Kihl and Emma Fitzgerald are part of the Excellence Center at Linköping-Lund on Information Technology (ELLIIT) strategic area.

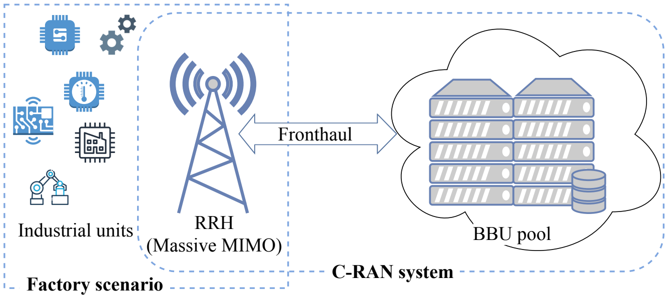

Fig. 1: Target system architecture

II. Target System

In this paper, we focus on a C-RAN architecture that includes one BBU pool and one massive MIMO RRH, connected with a fronthaul link, shown as Fig. 1. We assume that the fronthaul bandwidth limitation is neglectable for the reason that all the Physical Layer (PHY) functionalities are operated on the RRH and no raw base-band data block are transmitted over the fronthaul link.

We focus on an indoor factory automation scenario, where the numerous sensors, controllers and actuators, here called industrial units. We assume that the distance between the industrial units is small enough that they can all be covered by the radio range of one RRH. The actuation units and controllers are distributed and their feedback loops are connected via the mobile network. Sensors are also deployed to measure the states of the actuators. We thus categorise communications for the control loops as Ultra-Reliable and Low-Latency Communication (URLLC) type traffic, which has strict deadlines. As a transmission request is sent and once the deadline is passed without without being allocated the channel resource, the transmission attempt is failed and the data to be transmitted in this attempt is considered to be lost. Meanwhile, there are also other units in the plant with communication needed, they require massive connections but have higher tolerance on the latency, thus the priority is lower than the control units.

The time-frequency space of a single massive MIMO system can be divided into coherence blocks, which is the largest time interval during which the channel can be viewed as time-invariant and channel frequency response is approximately constant. A coherence block is shared by uplink data, downlink data and uplink pilot transmissions. The uplink pilots are used by the base station to estimate each User Equipments (UE)'s CSI, which is for precoding needed to process the input and output data. Thus a pilot is needed for a given UE to transmit data successfully and we will consider the uplink pilots as the resources that the UEs require before a transmission can start.

URLLC type requests (coming from control units and sensors) are prioritised. In the meantime, the number of pilots in a coherence interval are limited. Therefore, a MAC scheduler is deployed in the BBU pool that assigns pilots to requests in each coherence interval. The objective of the resource scheduler is to schedule as many requests as possible within their deadlines. We address a scheduling function only for the URLLC type of requests as they have more stringent latency requirements. The traffic generated by all the other units is considered as background traffic, which gets assigned if there are pilots left in each coherence interval after the URLLC requests are scheduled.

The massive MIMO RRH follows the decisions made by the remote scheduler and sends the status of all active requests frequently to the BBU. Due to the geographic separation of the actuator (the RRH) and the controller (the scheduler in the BBU pool), there will be a delay between a control decision and its actuation. Comparing to a base station with all functionalities executed locally, the target system should be able to make a decision that adapts to the active requests at a future moment. If the decision under-estimates the number of active requests, the URLLC traffic might be time out due to the long waiting time and the transmission is dropped; likewise, if the decision over-estimates the number of active requests, the pilots resource could be wasted since they are assigned to nonexistent requests, which means that the less prioritised traffic would get much less resource and will have longer waiting time or eventually drop the transmission.

Therefore, we propose the two following performance metrics for investigating how a massive MIMO pilot scheduling is affected by a C-RAN architecture. The first performance metric is the *loss probability* (L) of a request. A loss occurs every time a request is not scheduled within its deadline. The average loss probability can be calculated as the ratio between the dropped transmissions and the total number of requests. The second performance metric is the *pilots utilization* (U). A pilot is wasted every time it is allocated by the scheduler, but there is no request that can use it when the decision arrives at the RRH. The average utilization of pilots can be calculated as the ratio between the pilots that are successfully assigned to requests and the total number of pilots that the decisions allocate.

III. Proposed Scheduling Strategy

In this section, we propose a pilot scheduling strategy for massive MIMO using a C-RAN architecture. We assume that the RRH keeps an arrival queue of all transmission requests and that the BBU is able to keep track of the status of this queue. Every time the BBU gets updated the queuing information, it sends new scheduling decisions so that the RRH could apply the updated decisions on the latest status of the queue.

The RRH periodically wraps the status of the queue and sends it as a “Report” to the BBU. The report would include the UE id, the arrival time and the deadline of each request. Once the BBU receives a new “Report”, it inspects the number of requests from each UE in the queue and makes “Decision” that picks which UEs will be assigned the pilots and how many pilots will be assigned to each chosen UE. If the total number of requests in the queue is less than the number of available pilots P in a coherence block, the decision simply assigns equal number of pilots as the pending requests to each

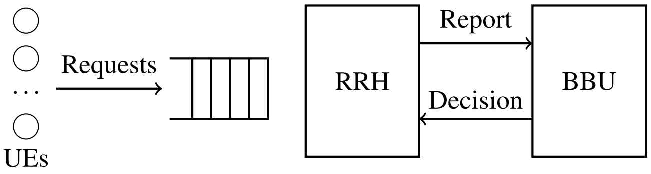

Fig. 2: Simulation model

UE per time slot. Conversely, if the queue length is too large that P pilots are not sufficient to meet all the requests at once, the decision would choose the UEs and assign the pilots via one of the following scheduling policies:

- First Come First Served (FCFS): Only the UEs that sent the first arrived P requests in the arrival queue are allocated the pilots.
- Earliest Deadline First (EDF): Only the UEs that sent the first P requests in the arrival queue with the earliest deadlines are allocated the pilots.
- Proportional Division (PD): The weight value of UE k is $\omega_k = N_k/N$ where N is the total number of requests in the reported queue and N_k is the number of requests sent by UE k in the queue. Thus number of pilots allocated to this user is $\lfloor \omega_k P \rceil$

In this way, as the assignment is not to a specific request but to an UE that sent out the request, it is still applicable if the UE is consistently in a transmission state when the decision is conducted by the RRH, even if this request becomes inactive in the queue at this moment.

IV. Evaluation

We implemented a simulation program in Simpy[1] to evaluate the impact of latency on the performance of the remote scheduling over C-RAN. The simulation model is shown as Fig.2, in which the BBU, RRH and the UEs are concurrent processes driven by time and message (the "Report" and "Decision") exchange. In the simulation, we deploy an arrival queue to buffer all the active transmission requests from the UEs to the RRH. Each request is stamped with its arrival time, the UE id and the deadline at which it is supposed to be expired if no pilot has been allocated.

A. Traffic Generation

For this paper, each UE sends requests according to an ON/OFF process. However, we have investigated the system using other arrival processes as well, and the general results are not dependent on this specific arrival process. During the OFF period t_{off} of an UE, it is in sleep mode and no transmission requests are sent. When the UE is awake during the ON period t_{on}, it follows a Poisson distribution of rate λ to send out the pilot requests. Each request is then followed by a data transmission if the pilot is granted. The length of each t_{on} of an UE can also represent the data size to be transmitted during this awake period and follows a Pareto distribution with tail index α . Denoting that the minimum ON duration is d_{on}, the mean of ON duration is thus:

$$\bar{t}_{on} = \frac{\alpha d_{on}}{1-\alpha} \tag{1}$$

We consider t_{off} to be nearly constant but varies with a normal distribution around the mean $\bar{t}_{off}$. The ON/OFF model thereby leads to the average arrival rate $\bar{\lambda}$ of UE k and the offered load $\bar{\rho}$ of the whole system with $K+1$ customers are:

$$\bar{\lambda}^k = \frac{\bar{t}^k_{on}\lambda^k}{\bar{t}^k_{on} + \bar{t}^k_{off}} \tag{2}$$

$$\bar{\rho} = \frac{\sum_{k=0}^{K} \bar{\lambda}^k}{P/T_{slot}} \tag{3}$$

B. Performance Metrics

In the simulations, we investigated how the system is affected by fronthaul latency using the aforementioned performance metrics: *loss probability (L)* and *pilots utilization (U)*. During each coherence interval, P pilots can be allocated. The coherence interval will hereinafter be noted as the time slot T_{slot} in the resource allocation problem. In a time slot j, the RRH takes a decision that $\hat{P}^k_j$ pilots should be assigned to UE k waiting in line, where $k = \{0, 2, ...K\}$ and $\sum_{k=0}^{K} \hat{P}^k_j \leq P$. The actual number of active requests from UE k in the queue is N^k_j. Leading that in a time slot j, the number of wasted pilots W^k_j for UE k is

$$W^k_j = \max(\hat{P}^k_j - N^k_j, 0) \tag{4}$$

It yields the resource utilization in slot j:

$$U_j = 1 - \frac{\sum_{k=0}^{K} W^k_j}{\sum_{k=0}^{K} \hat{P}^k_j} \tag{5}$$

Taking that the length of one simulation is T, the resource utilization during the whole service period is:

$$U = 1 - \frac{\sum_{j=1}^{T/T_{slot}} \sum_{k=0}^{K} W^k_j}{\sum_{j=1}^{T/T_{slot}} \sum_{k=0}^{K} \hat{P}^k_j} \tag{6}$$

And the overall loss probability of the system during T is given by:

$$\bar{L} = \frac{\sum_{k=0}^{K}(\bar{\lambda}^k T - \sum_{j=1}^{T/T_{slot}} S^k_j)}{\sum_{k=0}^{K} \bar{\lambda}^k T} \tag{7}$$
$$\text{where } S^k_j = \min(\hat{P}^k_j, N^k_j)$$

We noted that $\bar{L}$ is the mean loss probability calculated from the mean arrival rate $\bar{\lambda}^k$ of each UE. In the simulation experiments, we measured the actual number of arrivals in the system to calculate the loss probability L.

In the experiments, we run each simulation with system parameter set as indicated in Table I and increases the delay variable from 0ms to 20ms. Given that the total number of UEs is K, from Eq.(1-3), the offered load in our experiments is $\bar{\rho} = K/36$. In the next section, we show our simulation results

[1] https://simpy.readthedocs.io/en/latest/

TABLE I: System parameters used in the simulation.

Parameter name	Value	Symbol
Simulation length	300000 ms	T
Slot time	0.5 ms	T_{slot}
Available pilots per slot	12	P
Minimum ON duration	100ms	d_{on}
Mean OFF duration	600ms	$\bar{t}_{off}$
Pareto distribution tail index	1.5	α
Poisson distribution rate	2 requests/ms	λ
Repetitions per simualtion	20	-

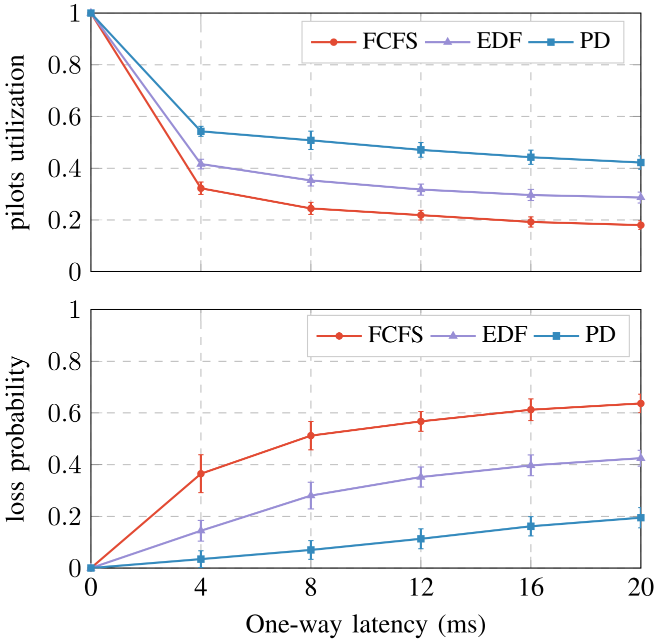

Fig. 3: The pilots utilization ans loss probability with three scheduling policies when the offered load $\bar{\rho} = 0.5$

when the offered load $\bar{\rho} = 0.5$ under different system delays and discuss how the two performance metrics are affected by the fronthaul latency.

V. Simulation Results

From our experiments, we noticed that the system has a relatively robust performance if the incoming traffic has a low variance. The reason for this is that the queuing status would be rather time consistent for these kinds of traffic, which means that the scheduling decisions will be less affected by the fronthaul latency. Therefore, we introduced the more bursty ON/OFF arrival process, so that the effects of fronthaul latency on the scheduling performance becomes more evident.

Fig.3 shows the scheduling performance for the three different policies when the offered load is $\bar{\rho} = 0.5$ and the one-way fronthaul latency increases from 0 to 20ms. We note that Fig.3 gives the mean values of the two performance metrics and 0.95 confidence interval around the mean from 20 repeated runs.

We can see from the figure that although the policies show different robustness and capabilities to handle the bursty traffic, the performances is drastically affected by the latency. Among the three policies, proportional division has the most robust performance. It gives a moderate loss probability in the low latency (4ms) case, which yet becomes unacceptably large for URLLC type of requests as the fronthaul latency increases. Moreover, only 80% of the pilots allocated by the decision are utilized and assigned to the UEs in the best case among our experiments, which means the resource left for the other units would be insufficient to support large number of connections.

VI. Conclusions

In this paper, we have presented a MAC layer pilot scheduling problem over C-RAN. We proposed a scheduling strategy with three commonly deployed policies to allocate the pilots to the transmission requests sent to the radio system. Simulation experiments were performed to study how the scheduling function is affected by fronthaul latency. We used two performance metrics, loss probability and pilot utilization.

The next step of our work is to develop a new scheduling strategy, which is capable to estimate and predict the queuing status and make a decision based on predicted status instead of the reported one. In this way, we expected the performances on both loss probability and pilot utilization would be improved for the C-RAN system.

References

[1] 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,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.

[2] N. Nikaein, “Processing radio access network functions in the cloud: Critical issues and modeling,” in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, ser. MCS ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 36–43.

[3] N. J. Gomes, P. Chanclou, P. Turnbull, A. Magee, and V. Jungnickel, “Fronthaul evolution: From CPRI to Ethernet,” *Optical Fiber Technology*, vol. 26, pp. 50–58, Dec 2015.

[4] T. Wan and P. Ashwood-Smith, “A performance study of CPRI over Ethernet with IEEE 802.1qbu and 802.1qbv enhancements,” in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[5] S. Mikroulis, L. N. Binh, I. N. Cano, and D. Hillerkuss, “CPRI for 5G cloud RAN? – efficient implementations enabling massive MIMO deployment – challenges and perspectives,” in *2018 European Conference on Optical Communication (ECOC)*, Sep. 2018, pp. 1–3.

[6] S. Park, H. Lee, C.-B. Chae, and S. Bahk, “Massive mimo operation in partially centralized cloud radio access networks,” *Computer Networks*, vol. 115, pp. 54 – 64, 2017.

[7] D. M. Kim, J. Park, E. De Carvalho, and C. N. Manchon, “Massive MIMO functionality splits based on hybrid analog-digital precoding in a C-RAN architecture,” in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct 2017, pp. 1527–1531.

[8] W. Chang, T. Xie, F. Zhou, J. Tian, and X. Zhang, “A prefiltering C-RAN architecture with compressed link data rate in massive MIMO,” in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, May 2016, pp. 1–6.

[9] S. Parsaeefard, R. Dawadi, M. Derakhshani, T. Le-Ngoc, and M. Baghani, “Dynamic resource allocation for virtualized wireless networks in massive-MIMO-aided and fronthaul-limited C-RAN,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9512–9520, Oct 2017.

[10] Y. Cai, F. R. Yu, and S. Bu, “Cloud radio access networks (C-RAN) in mobile cloud computing systems,” in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 369–374.