Latency-aware Radio Resource Allocation over Cloud RAN for Industry 4.0

Peng, Haorui; Tärneberg, William; Kihl, Maria

Published in:
2021 International Conference on Computer Communications and Networks (ICCCN)

DOI:
10.1109/ICCCN52240.2021.9522200

2021

Document Version:
Early version, also known as pre-print

Latency-aware Radio Resource Allocation over Cloud RAN for Industry 4.0

Haorui Peng, William Tärneberg, Maria Kihl
Department of Electrical and Information Technology, Lund University, Lund, Sweden
{haorui.peng, william.tarneberg, maria.kihl}@eit.lth.se

***Abstract*—The notion of Cloud RAN is taking a prominent role in narrative for the next generation wireless infrastructure. It is also seen as a mean to industrial communication systems. In order to provide reliable wireless connectivity for industrial deployments, by conventional means, the cloud infrastructure needs to be reliable and incur little latency, which however, is contradictory to the stochastic nature of cloud infrastructures. In this paper, we investigate the impact of stochastic delay on a radio resource allocation process deployed in Cloud RAN. We proceed to propose a strategy for realizing timely cloud responses and then adapt that strategy to a radio resource allocation problem. Further, we evaluate the strategies in an industrial IoT scenario using a simulated environment. Experimentation shows that, with our proposed strategy, a significant performance improvement on timely responses can be achieved even with noisy cloud environment. Improvements in resource utilization can be also attained for a resource allocation process deployed over Cloud RAN with this strategy.**

***Index Terms*—Cloud RAN, Latency-constraint network, Resource allocation, Industry 4.0**

I. Introduction

The Fifth Generation Wireless Specifications (5G) is shaping the narrative for the Industry 4.0 era. With high reliability, high throughput and low latency, 5G is enabling many new applications in that domain. Further, Cloud RAN is an intriguing candidate Radio Access Network (RAN) architecture for 5G and beyond, as it promotes softwarization and resource centralization in RANs.

The basic concept of Cloud RAN is to detach the Base-Band processing Units (BBUs) from multiple legacy Radio Base Stationss (RBSs), and centralize them into a BBU pool built on cloud-native techniques. The remaining Remote Radio Heads (RRHs) are only equipped with basic radio-frequency functionalities, while the BBU pool allows for cooperative base-band signal processing for multiple RRH sites. In addition, more elaborate decisions and system-wide optimizations can be made when more of the system is orchestrated from the same point, such as the case in Cloud RAN.

In a typical cloud service, a set of dynamic worker nodes are deployed to support its workloads. Then a load-balancer, distributes incoming requests to those workers. The worker nodes share virtualised resources and are subject to a resource management strategy. Consequently, clouds and the extension Cloud RANs, are stochastic and dynamic systems in their own right. This so called *cloud delay* incurred by clouds includes not only the network delays, but also the admission time and execution time. From a Cloud RAN perspective, the stochastic nature of clouds incurs detrimental delays in between signal processing functions, which essentially introduce interruptions to the signal processing function chain [1].

In many future wireless systems aimed at 5G and beyond, for example, Massive Multiple Input Multiple Output (MIMO) [2], the radio resource allocation is performed at the RRH. There is a scheduler deciding how to allocate the available radio resources to the User Equipments (UEs) according to a policy. Often, the objective of the allocation policy is to mitigate resource starvation, collision and congestion. When deploying an allocation process over Cloud RAN, the decisions are performed in the BBU pool and then actuated by the RRH.

However, the stochastic properties of the Cloud RAN environment will cause uncertainties in such an allocation process. As the message exchanged between the BBU pool and RRH will be delayed and may arrive out-of-order. In radio resource allocation, this delay may cause false allocation to the UEs. We presented in [1] the trade-offs between resource utilization and transmission reliability over the communication system when deploying a naive massive MIMO radio resource scheduler over Cloud RAN. Therefore, there is a need of purpose-built schedulers that can cope with the disturbances caused by the Cloud RAN environment.

Some work addressed the resource allocation problem for low-latency communication services in Cloud RAN under different scenarios. In [3], the authors focused on an energy consumption minimization problem for computation tasks for a mobile edge cloud enabled Cloud RAN system. Also, in [4], an energy efficient joint resource scheduling scheme was proposed for a Cloud RAN system. There are some works like [5] [6], which utilized distributed allocation algorithms to minimize the response time or the computation latency in Cloud RAN systems.

Apart from the studies on resource allocation, the characteristics of the fronthaul link delay and the jitter of the delay in Cloud RAN systems were investigated in [7] [8]. Some works proposed solutions that compensate the communication

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, 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). The authors are part of the Excellence Center at Linköping-Lund on Information Technology (ELLIIT), and the Nordic University Hub on Industrial IoT (HI2OT) funded by NordForsk.

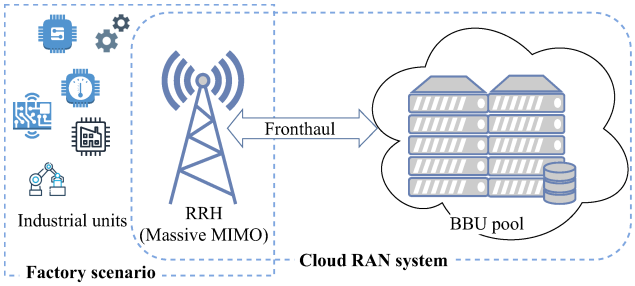

Fig. 1. Target system architecture.

delays or reduce the impact of delays for different networked systems, for example [9] and [10].

To the best of our knowledge, very few studies have addressed the stochastic nature of a Cloud RAN system in a radio resource allocation problem. In our work, we embrace the fact that delays over Cloud RAN systems are unavoidable and has stochastic characteristics. In this paper, we propose a radio resource allocation strategy for Cloud RANs. The proposed solution is then evaluated as massive MIMO pilot scheduler in an Industry 4.0 scenario with simulations. Our contributions in this paper can be summarized as follows:

- We propose a purpose-built radio resource allocation strategy for Cloud RAN that will mitigate the impact of the stochastic cloud delay.
- We develop a simulation model for the system and evaluates the proposed solution in an Industry 4.0 scenario.
- We show that our strategy significantly improves the radio resource utilization of the system without compromising the communication reliability.

II. Targeted System

In this paper, we target a Cloud RAN architecture that provides wireless communications in an industrial Internet of Things (IoT) scenario. A schematic overview is given in fig. 1.

A. Industry 4.0 scenario

In this paper we address an indoor factory automation scenario, where industrial UEs communicate over the network provided by Cloud RAN. In the envisioned industrial IoT scenario [11], the number of UEs can be extensive, with a density of $10,000$ devices per km^2.

We define two main types of UEs, Critical Units (CUs) and non-Critical Units (non-CUs). First, CUs are sensors, controllers, and actuators. The CUs generate control signals, usually periodically, and typically have strict Quality of Service (QoS) requirements. For example, latency less than 10ms and availability within the range of 95%-99.999%. For simplicity, we call all the signals exchanged among the UEs as transmissions via the network. Each transmission from a CU has a hard deadline, the transmission attempt failed If the CU is not assigned radio resource within its deadline. The number of CUs that have transmissions simultaneously must be limited and never overload the communication system, in order to guarantee transmission reliability.

Second, non-CUs represent collectively other types of devices and can be a much larger amount. Characteristically, they have less stringent requirements and usually sporadic transmissions. The traffic generated by non-CUs is considered as background traffic in the system.

B. Cloud RAN system

A Cloud RAN system consists of a set of RRHs connected with a BBU pool over a front-haul link. The BBU pool is deployed in a cloud-native execution environment. Consequently, the functions offered by the BBU pool are subject to stochastic delays. Also, due to opaque cloud management policies, any messages sent between the RRHs and the BBU pool may come out-of-order.

Since we mainly focus on a radio resource allocation process running over Cloud RAN, which is a MAC layer function of RBS, we assume that the Physical Layer (PHY) functionalities are operated on the RRH and no raw baseband data blocks are transmitted over the front-haul link to the BBU pool. For a manufacturing process, the communication distance is generally within 200m [12], thus we assume that all the UEs can be covered by the radio range of one RRH in our target scenario. The radio resource allocation process is adopted for a single-cell system. However, this is not a limiting factor on our work.

C. Radio resource allocation

In this paper, we adopt a *massive MIMO up-link pilot scheduling* as the use case of our resource allocation process deployed across a Cloud RAN. An up-link pilot is the prerequisite for a UE to be permitted to transmit during a *coherence interval* of massive MIMO. The coherence interval is determined by for how long the wireless channel state is considered to be coherent. Pilot scheduling is performed for each coherence interval. If a UE is allocated a pilot, we call its transmissions can be served within the next coherence interval, and it can use the rest radio capacity in this coherence interval for its data transmission. Irrespective of how many transmission a UE have to make, it needs one pilot to be allowed to transmit during an interval.

The scheduler is located in the BBU pool of a Cloud RAN system and determines when and how to allocate the pilots, based on an explicit objective. The RRH allocates the up-link pilots to allow for the UEs transmissions in the network. From this point on, the resource allocation problem is simply referred to as *pilot scheduling*. The scheduling process over Cloud RAN can be divided into the the following processes:

- The *allocation process* on RRH that allocates the pilots to the UEs with pending transmissions.
- The *updating process* that sends the updates about the pending transmissions to the BBU pool by the RRH.
- The *scheduling decision process* that makes the scheduling decisions in the BBU pool and sends the decisions to the RRH.

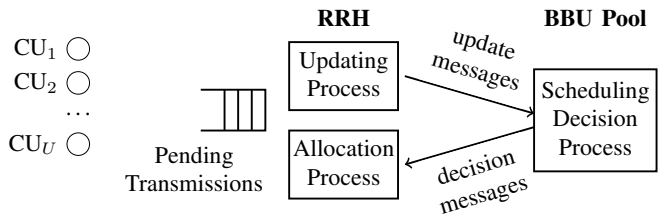

Fig. 2. System model

III. System Model

In this section, we detail a model of the targeted system as presented in section II. The basic components of the system are; a set of UEs, a Cloud RAN infrastructure inclusive of a RRH and a BBU pool. Update messages are sent from the RRH to the BBU in the Cloud RAN, to which the BBU responds with a scheduling decision. Both update and decision messages are delayed due the the stochastic cloud system. An overview of the system and the relationship between those components is shown in fig. 2. In this paper, we consider the CUs in the pilot scheduling problem, since these are the UEs with prioritized traffic. Other UEs, that is the non-CUs, will get the remaining pilots after all CUs have been served in a coherence interval.

A. Cloud Delay

Radio resource allocation over Cloud RAN includes information dissemination between the RRH and the BBU, as described in section II. Here we denote "update" message as the information sent by the updating process at RRH to the scheduling decision process resides in the BBU pool. Likewise, a "decision" message originates from the BBU pool to the allocation process at RRH.

The cloud, its opaque management systems, shared infrastructure, and intermediate network incur a stochastic delay. This delay is represented as two independent stochastic variables, d_{update} and d_{decision}, representing the time for making and delivering update and decision messages. The two delays are inclusive of all execution times, admission and queuing delays in the cloud, as well as delays along the path of a message. In the following, we refer both delays to cloud delays incurred by the system.

B. Industrial Applications

We denote the number of active CUs covered by the radio range of the RRH, U. CU$_u$ is the uth active CU, where $u \in \{1, 2, ..., U\}$. Each CU$_u$ triggers transmissions according to a stochastic processes to the RRH. The inter-arrival time between subsequent transmissions from CU$_u$ is denoted c_u. A CU can only have successful transmission in a coherence interval if it is assigned a pilot. Each transmission triggered by CU$_u$ has a deadline D_u. A transmission is discarded and fails if it is not served by a pilot before its deadline.

C. Massive MIMO Pilot Scheduling

For each coherence interval, the Massive MIMO up-link pilot scheduling process allocates pilots to the resident CUs. For general applicability, a coherence interval is now referred to as a *slot*, and the length of a slot if denoted as T_c. Also, we assume that the BBU pool and the RRH are synchronized in time, which means that a slot k represents the same time interval at both the BBU pool and the RRH.

At the beginning of a slot k, the RRH updates the BBU about its current state, that is the number of pending transmissions from each CU$_u$, denoted $Q_u(k)$. In the following, we will call $Q_u(k)$ for the state of CU$_u$. The state of all CUs at slot k is then denoted as $\mathbf{Q}(k) = \{Q_1(k), Q_2(k)..., Q_U(k)\}$.

The BBU pool performs the scheduling decision process and then responds the RRH with the decision message, which is actuated by the RRH. We denote a scheduling decision to be applied at slot k by $\mathbf{P}(k) = \{P_1(k), P_2(k), ...P_u(k)\}$, where

$$P_u(k) = \begin{cases} 1 & \text{allocate pilot to CU}_u\text{at slot } k \\ 0 & \text{not allocate pilot to CU}_u\text{at slot } k \end{cases} \quad (1)$$

At every slot k, the RRH allocates pilots to the active CUs according to the decision $\mathbf{P}(k)$. We define that, in total, p pilots are available per slot. Consequently, at most p CUs can be assigned pilots per slot. If $P_u(k) = 1$, N transmissions from CU$_u$ can be served at slot k. Thus, k is the actuation slot of $\mathbf{P}(k)$.

IV. Problem Definition

In this section, we detail the challenges incurred by the stochastic properties of a Cloud RAN system on the scheduling process. We begin with describing the main obstacles when a *naive pilot scheduling scheme* is deployed to a Cloud RAN, in which the inherent delays is not accounted for in the scheduler. This evaluation of this deployment was performed in [1]. Here, the scheduler is triggered every time an update message is delivered to the BBU pool. Upon completion, a scheduling decision is sent to the RRH. A round trip of an update and decision messages delivery needs be finished within one slot, as the state of the RRH may change at the next slot, and new update will be sent to request for new decisions.

However, without taking into account the stochastic delays of update messages and scheduling decisions, a decision $\mathbf{P}(k)$, which is a response to an update message $\mathbf{Q}(k)$, may fail to be actuated at slot k timely. This would lead to false allocation in pilot scheduling, as the state may change at the RRH, and further yields unwanted performances as we discovered in [1].

We herein mainly consider two performance metrics: *timely applied decisions*, and *pilot utilization*, to examine whether a scheduling process performs properly across Cloud RAN system. The prerequisite of taking into account the utilization performance is when the reliability requirements from the CUs are meet. We have investigated the reliability performance of the system in [1] and showed that, the availability range of this system can achieve over 95%, which meet the industrial requirements depicted in section II-A. Thus, we note that,

in this paper, the reliability performance is not presented. If it is not indicated explicitly, all the performance evaluations are within availability range of over 95%. Below we define timely decision and utilization performances, as well as a set of challenges based on the two properties.

A. Timely Applied Decisions (R)

The cloud delay may cause a stale decision to actuate allocation. In this case, the decision is considered *not timely applied*. Conversely, at slot k, the decision $\mathbf{P}(k)$ is applied, we call this *timely applied decision* at k.

The ratio between timely applied decisions and all decisions is denoted R. Also, we denote $R_{k_i:k_j}$ as the ratio of timely applied decisions from slot k_i to k_j. When a decision is applied at a slot it is not intended to be, the state of pending transmissions may deviate. Further, false allocation may occur, leading to performance degradation in *pilot utilization* in pilot scheduling problem.

B. Pilot Utilization (β)

We use *pilot utilization* to to evaluate the performance of the pilot scheduling strategy. Under the industrial scenario described in section II-A, the non-CUs will get the remaining pilots in a slot when all transmissions from the CUs have been served. Therefore, unused, or wasted, pilots is highly undesirable. A pilot is wasted every time it is assigned to a CU that has nothing to transmit. This can occur if a scheduling decision is based on an outdated RRH state and if scheduling decisions are not delivered timely.

The decision $\mathbf{P}(k)$ determines a set of CUs to be allocated pilots at k, as the number of available pilots is p in a slot, the decision should satisfy $\sum_1^U P_u(k) \leq p$.

For each CU has assigned a pilot, the number of transmissions that can be served is N. However the actual number of pending transmissions from CU_u at this moment is $Q_u(k)$. This will yield the following number of wasted pilots, denoted $\omega_u(k)$, for CU_u at slot k:

$$\omega_u(k) = \max(0, (1 - \frac{Q_u(k)}{N})P_u(k)) \tag{2}$$

which leads to the pilot utilization $\beta(k)$ for all CUs in this allocation:

$$\beta(k) = 1 - \frac{\sum_{u=1}^U \omega_u(k)}{\sum_{u=1}^U P_u(k)N} \tag{3}$$

C. Research challenges

Now that we have a model and defined the performance metrics of the scheduler we can begin the discuss the inherent challenges with radio resource allocation over Cloud RAN. Firstly, we can now define the objective of the scheduler as follows:

- Assign pilots to all CUs with pending transmissions, in a fair manner, before their transmissions have expired.
- While avoiding starving the background traffic, which is a consequence of resource waste.

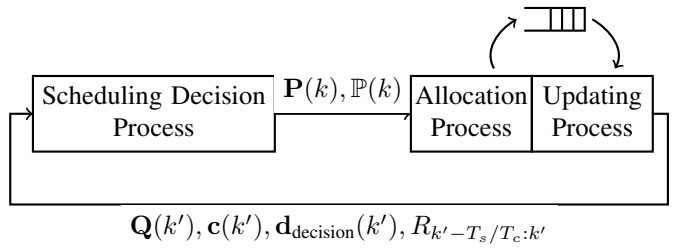

Fig. 3. Overview of the scheduling process over Cloud RAN at slot k', when an update message was sent from the RRH to the BBU pool and allocation is actuated based on an arrived decision $\mathbf{P}(k')$. Meanwhile, the BBU pool performs a decision for a future slot k and sends this decision to the RRH.

We have shown in [1], that a naive scheduling method over Cloud RAN, is feasible when meeting the reliability requirements for industrial standards, by keeping the loss under 5%. However, a naive scheduling would also lead to huge amounts of pilot waste. Therefore, in the next section, given the stochastic nature of Cloud RAN, we propose a new allocation strategy that is focused on improving the resource utilization for time-critical applications without compromising transmission reliability, that is keeping the loss under 5%.

V. Proposed Solution

In this section, we propose an novel scheduling strategy over Cloud RAN that addresses the challenge of timely arrival of decisions, as detailed in section IV. Our proposed solution can handle the delayed and out-of-order messages that is an effect of a stochastic Cloud RAN environment. Thereby, the pilot utilization is radically improved, without compromising reliability performance. An overview of the proposed solution is shown in fig. 3.

To remedy stochastic delays, the *updating process* and *scheduling decision process* are handled asynchronously in our proposed solution. The scheduling decisions are generated periodically at the BBU pool for a future actuation slot k, based on the predicted arrival time of this decision and estimated RRH state at k. In this manner, the actual state $\mathbf{Q}(k)$ may not be delivered at the BBU when the decision is made. The state estimation is made based on all the historical information in previously delivered update messages. Add-on concepts such as message buffering and redundancy are also utilized to guarantee the decisions to be timely applied. Below we describe the details of our proposed scheduling strategy.

A. Updating Process

At each slot k', the RRH sends an update message to the BBU, for the purpose of state estimation of the RRH in the BBU pool. The update message includes the following:

- The number of pending requests $\mathbf{Q}(k')$ for each user.
- The inter-arrival times of the transmissions from each CU $\mathbf{c}(k') = \{\mathbf{c}_1(k'), \mathbf{c}_2(k'), ...\mathbf{c}_U(k')\}$, which arrived during slot $k'-1$. Here, $\mathbf{c_u}(k')$ is the set of all inter-arrival samples of CU_u measured during slot $k'-1$, thus $\mathbf{c}_u(k') = \{c_u[n_1], c_u[n_2], ...\}$.

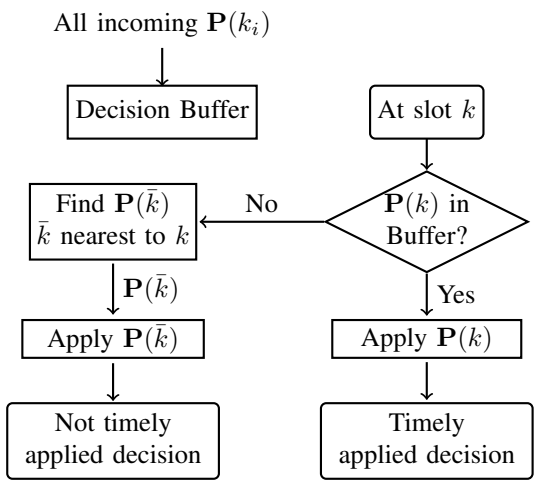

Fig. 4. Allocation process at RRH

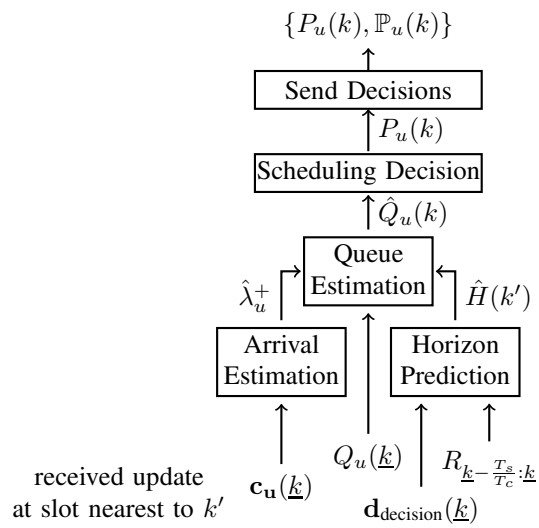

Fig. 5. Scheduling Decision process in the BBU pool at slot k', taking into account the updates sent by the RRH at time $\underline{k}$, which is the nearest slot to k' among all the delivered updates. The process performs a decision expected to be applied by the RRH at slot k, where $k \geq k'$

- The measured delay samples from scheduling decision messages that have arrived during slot $k'-1$. $\mathbf{d}_{\text{decision}}(k') = \{d_{\text{decision}}[m_1], d_{\text{decision}}[m_2], ...\}$.

Further, every T_s, the RRH includes timely applied decisions during last T_s in the update message. The timely applied decision is noted as $R_{k'-T_s/T_c:k'}$ if sent at slot k'. This information contributes to the horizon prediction of decision arrivals in the scheduling decision process.

B. Allocation Process

At each slot k, after sending the update message, the RRH applies a received decision to allocate the pilots to the active CUs. The allocation process is detailed in fig. 4.

As the decisions performed by the BBU are intended to be actuated in specific slots, there needs to be a solution for decisions that arrive earlier than intended, late or out-of-order. Therefore, we propose that the RRH buffers all arrived decisions and applies them at the intended actuation slot. If one decision fails to be delivered before its intended actuation slot, the RRH takes a buffered scheduling decision $\mathbf{P}(\bar{k})$ that is intended for slot $\bar{k}$, which is nearest to k, and applies this decision instead. This is based on the assumption that the state estimation for the nearest slot will, on average, is the second most accurate.

C. Scheduling Decision Process

The scheduling decision process in our proposed solution can be divided into several sequential sub-processes as presented fig. 5. In the following, we detail every sub-process as illustrated in the figure.

1) Scheduling Decision: The scheduler performs a scheduling decision, $\mathbf{P}(k)$, to be applied at a future slot k. The decision is based on the estimated state of the RRH on all active CUs at slot k.

In this paper, we implement a greedy allocation strategy, however, other scheduling methods can of course be used. The decision for CU_u is, $P_u(k) = 1$ if $Q_u(k)$ is non-zero and has one of the p largest values among the set $\mathbf{Q}(k)$.

The scheduling decision message includes both the newly made decision $\mathbf{P}(k)$ and h redundant scheduling decisions $\mathbb{P}(k) = \{\mathbf{P}(k-1), \mathbf{P}(k-2), ...\mathbf{P}(k-h)\}$, where the intended actuation slots are before k. Using redundant messages means that if a decision intended for slot k is delayed and thereby arrives later than its intended actuation slot, later decision messages may be able to deliver this decision for slot k in time to the RRH. This not only significantly improves the timely applied decisions, but also benefits the utilization performance, as it will be shown in the results.

2) Queue Estimation: The decision $\mathbf{P}(k)$ is indented to be applied at a future slot k. Thereby a state estimation at k needs to be provided, which is denoted by $\hat{\mathbf{Q}}(k)$.

Considering that at slot k', we take the state $Q_u(\underline{k})$ of CU_u from all the received update messages at the BBU pool, where $\underline{k} \leq k'$ and nearest to k'. If the average arrival rate of the requests from CU_u is λ_u, and the predicted time horizon for when a decision should be actuated is $H(k')$, the queue sizes for slot k, $\hat{Q}_u(k)$, can be estimated as follows:

$$\hat{Q}_u(k) = Q_u(\underline{k}) + \lambda_u(k - \underline{k}) - \sum_{\kappa=\underline{k}}^{k-1} P_u(\kappa) \quad \text{where } k = k' + H(k') \tag{4}$$

The term $\sum_{\kappa=\underline{k}}^{k-1} P_u(\kappa)$ corresponds to all decisions that are presumably to be applied from slot $\underline{k}$ to $k-1$.

3) Arrival Process Estimation: In eq. (4), the term $\lambda_u(k - \underline{k})T_c$ is used to predict the number of transmissions for CU_u that have been triggered from $\underline{k}$ to k. We use an Exponential Moving Average (EMA) estimator in order to estimate average inter-arrival time $\hat{c}_u$ of requests for CU_u, which gives:

$$\hat{c}_u^+ = \alpha_c \hat{c}_u^- + (1 - \alpha_c) c_u \tag{5}$$

Here, c_u is taken from the inter-arrival time sample $\mathbf{c_u}(\underline{k})$ informed in the most recent update message. We denote by $\hat{c}_u^+$ the new estimate on c_u. $\hat{c}_k^-$ is the old estimate and α_c is the weight of the EMA estimator. Further, the average arrival rate of CU_u can trivially be derived as:

$$\hat{\lambda}_u^+ = 1/\hat{c}_u^+ \tag{6}$$

4) Predicted Time Horizon: A decision message is performed in slot k' and should be applied in slot k, where $k \geq k'$. $k - k'$ is defined as the predicted time horizon, $\hat{H}(k')$. The predict time horizon is a crucial part of our proposed strategy, since it determines how delayed a decision message can be. A longer predicted time horizon will increase the ratio of timely applied decision, however, at the same time introduce more inaccuracies in the stare estimation.

Therefore, in this paper, we propose to calculate the predicted time horizon by using an estimate of the average decision delay $\hat{d}_{\text{decision}}$, and adding an offset σ, as follows:

$$\hat{H}(k') = \left\lceil \frac{\hat{d}^+_{\text{decision}}}{T_c} \right\rceil + \sigma^+ \tag{7}$$

Here, $\hat{d}^+_{\text{decision}}$ is the estimation of the average decision delay given by an EMA with weight α_d:

$$\hat{d}^+_{\text{decision}} = \alpha_d \hat{d}^-_{\text{decision}} + (1 - \alpha_d) d_{\text{decision}} \tag{8}$$

Similar to the average inter-arrival time estimator in section V-C3, d_{decision} is a sample of the decision delay, informed in the update message $\mathbf{d}_{\text{decision}}(\underline{k})$. $\hat{d}^-_{\text{decision}}$ is the previous estimate of the average decision delay.

The offset value σ^+ is an output of a step controller via eq. (9) when a new update on the average timely applied decision ratio $R_{\underline{k}-\frac{T_s}{T_c}:\underline{k}}$ has arrived.

$$\sigma^+ = \begin{cases} \sigma^- + 1 & \text{if } R_{\underline{k}-\frac{T_s}{T_c}:\underline{k}} < r \\ \sigma^- & \text{Otherwise} \end{cases} \tag{9}$$

Here, σ^- is the previous offset value and initialized as 0. r is the lower bound reference value for timely applied decisions ratio. A prerequisite of applying eq. (9) based on σ^- is when the average network delay has minor changes or increases. If the estimated mean delay has decreased, σ^- is reset to 0, and the controller searches for a new offset value again. The feedback $R_{\underline{k}-\frac{T_s}{T_c}:\underline{k}}$ should be calculated from a sequence of past slots and the measurements size should be large enough to be confident. We thus define the sampling time of the step controller as T_s, which is much greater than the scheduling time slot length T_c. Therefore, $R_{\underline{k}-\frac{T_s}{T_c}:\underline{k}}$ is collected through every T_s/T_c slots. In this way, the mean estimation on $\hat{d}_{\text{decision}}$ is made every T_c but σ is made every T_s.

With the step controller, if the number of discarded decisions exceeds a set point, the predicted time horizon is extended by increasing the offset value. So that the probability that a decision arrives before its indented actuation time is increased. If a decision is performed at slot k', the indented actuation slot k is given as $k = k' + \hat{H}(k')$.

TABLE I
PARAMETERS OF TRANSMISSION ARRIVAL PROCESS

Parameter name	Value	Symbol
inter-arrival time mean	10 ms	c
inter-arrival time std	0.0005	δ
Number of CUs	20	U
Deadline of a transmission from CU_u	10 ms	D_u

VI. EXPERIMENTS

In this section, we describe our experiments for evaluating the performance of our proposed pilot scheduling strategy over Cloud RAN. In our evaluation, we address the performance metrics described in section IV, timely applied event and pilot utilization. We examine how these performance metrics are affected by the stochastic properties of a Cloud RAN.

We evaluated our proposed strategy in a simulated environment built on SimPy [13] and the system model described earlier. We ran all experiments for a simulated system time of $T = 200$s and the results are based on the average of 20 repetitions. As a result, all confidence intervals are within 10% of the corresponding average value.

A. Simulation Parameters

The system model includes several system parameters that need to be set. These are described below.

1) Arrival process of transmissions: To generate traffic that can correspond to time critical industrial applications, we use the industry and IoT traffic models summarized in [12]. Each CU_u generates transmissions according to a homogeneous periodic stochastic process, with inter-arrival time $c_u \sim \mathcal{N}(c, \delta^2)$. Table I lists all parameters related to the arrival process of the transmissions and the values used in our simulations.

2) Stochastic delay: In this paper, we use the exponential distribution family to generate the two parameters representing the cloud delay, d_{update} and d_{decision}. For all distributions, the average delay was μ. We examine how different delay distributions and μ affect the system performance.

In the simulations, we evaluated the system performance when the cloud delays are deterministic, Erlang distributed, Exponential distributed, and Hyper-exponential distributed. Correspondingly, the coefficient of variance, CV^2, was $\{0, 0.5, 1, 2\}$. The average delay, μ, was varied from 0ms to 4ms, where $\mu = 0$ represents a system with a scheduler co-located with the RRH.

3) Scheduling strategy: The values of the parameters in the allocation process, placed in the RRH, are shown in Table II. The values of these parameters correspond to the radio spectrum parameters of our massive MIMO test-bed [14]. Table III lists the values for the parameters used in the scheduling decision process placed in the BBU pool.

B. Evaluation Methods

The objective of the evaluation is to show that our proposed pilot scheduling strategy efficiently mitigates the negative effects of the stochastic properties of the Cloud RAN, and

TABLE II
PARAMETERS OF THE ALLOCATION PROCESS IN THE RRH

Parameter name	Value	Symbol
Scheduling time slot length	0.5 ms	T_c
Number of available pilots per slot	12	p
Number of requests served by a pilot	1	N

TABLE III
PARAMETERS OF THE DECISION MAKING PROCESS AT BBU

Component	Parameter name	Value	Symbol
Arrival Estimation	EMA weight	0.999	α_c
Horizon Prediction	EMA weight	0.999	α_d
	Lower bound reference	90%	r
	Sampling time	2000ms	T_s
Redundant Decisions	No. of redundancy	2	h

thereby improves the pilot utilization without compromising reliability performance. Since such a strategy needs to mitigate the delayed and out-of-order decision messages, we will in the result section show how our proposed solution performs in comparison with three other methods that do not include the full set of remedy strategies. The strategies we used in the evaluation and their corresponding system parameters are summarized in Table IV. The details for the Naive Scheduling method under the same scenario is studied in [15].

In the experiments, we refer to the 95% availability industrial requirement noted in section II-A and set the maximum permissible loss to 5% for all the transmissions, then examine the pilot utilization performance when this condition is satisfied.

VII. RESULTS

In this section, we present and discuss our simulation results. We show that our proposed scheduling strategy of increasing the number of timely applied decisions, significantly improves pilot utilization, while meeting the industrial reliability requirement as noted in section II-A. Below we first present the results of our strategy for timely applied decisions. Then we present the results of the pilot scheduling process that relies on the ratio of timely applied decisions.

A. Timely Applied Decisions

With a high proportion of timely applied decisions, the scheduling strategy has been able to successfully mitigate the adverse effects of a Cloud RAN system. As a reference point, fig. 6 shows that when a naive scheduler is employed, no decision will be timely applied. This is because, the naive scheduler does not take into account the cloud delays incurred in the system, all decisions will arrive later than their intended actuation slot.

An extended predicted time horizon can improve the ratio of timely applied decisions, as revealed by the comparison between Naive Scheduling, Short Horizon and Single Decision methods in fig. 6, wherein the prediction horizon increases in turn. However this it is rather logical that, a longer predicted time horizon leads to an earlier arrival than the actuation time, and thereby a decision can be timely applied. Fig. 6 also shows

TABLE IV
EVALUATED SCHEDULING STRATEGIES IN THE EXPERIMENTS

Method Name	Parameters
Proposed Solution	Indicated in TABLE III
Single Decision	Same as Proposed Solution but $h = 0$
Short Horizon	Same as Proposed Solution but $h = 0, \sigma \equiv 0$
Naive Scheduling	Described in section IV

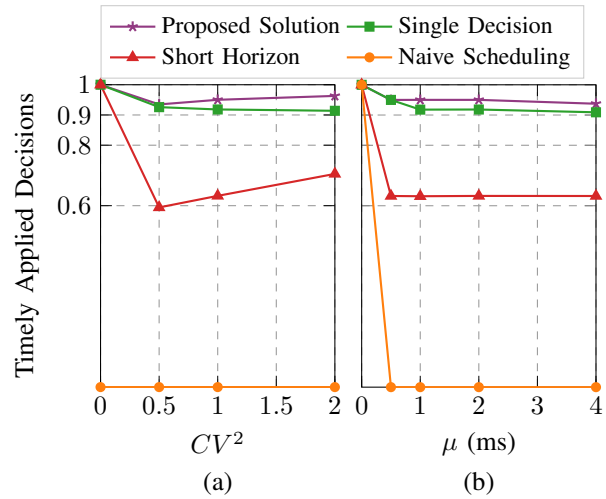

Fig. 6. Timely applied decision for the four methods (a) under different delay distributions when μ=2ms and (b) when μ increases for exponentially distributed delays.

that when adding redundant messages in Proposed Solution, the ratio of timely applied decisions is further improved, for all experiments.

Fig. 6 also reveals that the ratio of timely applied decision is not greatly affected by the length of the average delay, as it is remedied by the estimation on the average decision delays. Furthermore, a larger variance in the distribution may even improve the timely applied decisions. This result is mainly an effect of the different distributions we used for simulating cloud delays. For certain hyper-exponentially distributions, the probability that $d_{\text{decision}} \leq \mu$ is higher than the one in other distributions.

In summary, the ratio of timely decisions is highly correlated to the delay distribution and the perdition horizon. In this paper. We make use of the strategy detailed in section V-C4 to determine the prediction horizon. But it is an open question and various methods can be adopted to make the prediction.

B. Pilot Utilization

Fig. 7 shows the resulting average pilot utilization for our proposed up-link pilot scheduling strategy, compared with pilot utilization when Naive Scheduling. We note that with both our Proposed Solution and the Naive Scheduling, the loss of the transmissions is below 5%, as is required by the industrial standards. The Naive Scheduling method meets the transmission deadlines by keeping assigning redundant pilots to serve a single transmission. Although the Single Decision and Short Horizon methods significantly improved the timely

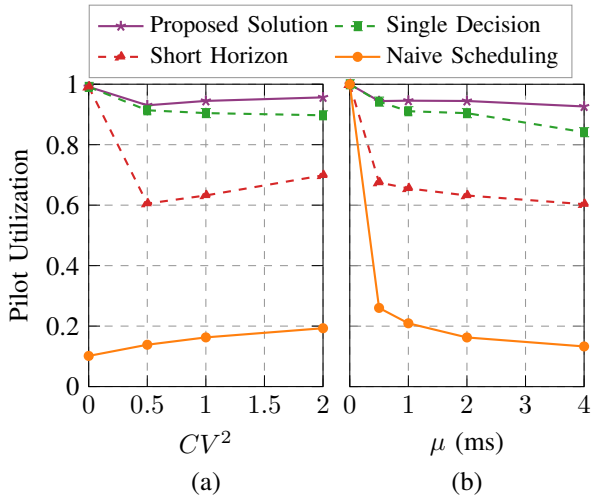

Fig. 7. Pilot utilization (a) under different delay distributions when μ=2ms and (b) as μ increases for exponentially distributed delay. Performances in dashed lines indicate that the methods didn't meet the reliability requirements.

applied ratio comparing to the naive scheduling method, the loss with these two methods does not meet the industrial standards, as the increment in timely applied decisions is not high enough to compensate the inaccuracy in state estimation long prediction horizon.

Comparing to a Naive Scheduling, our Proposed Solution increases the pilot utilization from less than 20% to over 90%. This means that the stochastic delays and out-of-order messages are effectively mitigated, which are the main effects of the Cloud RAN system. When less pilots are wasted on the CUs, the system becomes more capable to serve the traffic from non-CUs, and avoid starvation of these applications.

Comparing fig. 7 and fig. 6, it is clear that the pilot utilization is considerably impacted by the ratio of timely applied decisions. Briefly speaking, when more decisions are timely applied, less pilots are wasted. But we also see that the utilization is not completely decided by the timely applied decisions, but also the mean delays. As longer delay yields longer prediction horizon, which leads to more inaccuracies in the state estimation of the RRH.

VIII. Conclusions

In this paper, we investigated how radio resource allocation can be performed over Cloud RAN. We focused on the stochastic characteristics incurred by the Cloud RAN. We proposed a resource allocation strategy and implemented it for a massive MIMO up-link pilot scheduling problem. The proposed strategy mitigates the impacts of the Cloud RAN, in particular the stochastic delays and out-of-order messages. We have evaluated our proposed strategy with simulations. The effects of the Cloud RAN are mainly mitigated by including a predicted time horizon, estimated RRH state and sending redundant decisions, which are used to perform a scheduling decision for a future time slot. Our experiment results have shown that the proposed strategy significantly improves the pilot utilization by increasing the ratio of timely applied decisions, without compromising the industrial requirements on transmission reliability.

We also note that, there is a trade-off between the length of the predicted time horizon and the accuracy of state estimation. In this paper, we have not presented this trade-off, however, this will of course be performed in future work. Redundancy is introduced to mitigate the impacts from long predicted horizons. In this paper, we have not tried to optimize the number of redundant decisions, just showing the advantageous of including them. However, the optimal number of redundant decisions will of course depend on the available bandwidth of the front-haul link and the cloud delay, which will be further investigated.

References

[1] H. Peng, W. Tärneberg, E. Fitzgerald, and M. Kihl, "Massive MIMO pilot scheduling over Cloud RAN for Industry 4.0," in *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, sep 2020.

[2] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, Feb 2014.

[3] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic Task Offloading and Resource Allocation for Mobile-Edge Computing in Dense Cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282–3299, apr 2020.

[4] K. Wang, W. Zhou, and S. Mao, "Energy Efficient Joint Resource Scheduling for Delay-Aware Traffic in Cloud-RAN," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, dec 2016.

[5] L. Ferdouse, O. Das, and A. Anpalagan, "Auction Based Distributed Resource Allocation for Delay Aware OFDM Based Cloud-RAN System," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, vol. 2018-Janua. IEEE, dec 2017.

[6] H. Mei, K. Wang, and K. Yang, "Multi-Layer Cloud-RAN With Cooperative Resource Allocations for Low-Latency Computing and Communication Services," *IEEE Access*, vol. 5, pp. 19 023–19 032, 2017.

[7] J. Francis, J. K. Chaudhary, A. N. Barreto, and G. Fettweis, "Uplink Latency in Massive MIMO-Based C-RAN With Intra-PHY Functional Split," *IEEE Communications Letters*, vol. 24, no. 4, pp. 912–916, apr 2020.

[8] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee, "5G Fronthaul–Latency and Jitter Studies of CPRI Over Ethernet," *Journal of Optical Communications and Networking*, vol. 9, no. 2, p. 172, feb 2017.

[9] Y. Zong, X. Dai, P. Canyelles-Pericas, K. Busawon, R. Binns, and Z. Gao, "Modelling and Synchronisation of Delayed Packet-Coupled Oscillators in Industrial Wireless Sensor Networks," apr 2020.

[10] D. Xue and N. H. El-Farra, "Optimization-Based Actuator and Communication Scheduling in Networked Distributed Processes with Communication Delays," in *2019 American Control Conference (ACC)*, vol. 2019-July. IEEE, jul 2019, pp. 2558–2563.

[11] ATIS White Papers, "IOT categorization : Exploring the need for standardizing additional network slices," Tech. Rep. ATIS-I-0000075, September 2019, Accessed on April 19, 2020. [Online]. Available: https://access.atis.org/apps/group_public/document.php?document_id=51129

[12] T. Hosfeld, F. Metzger, and P. E. Heegaard, "Traffic modeling for aggregated periodic IoT data," in *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, feb 2018.

[13] "SimPy 4.0.2," Accessed on July 13, 2020. [Online]. Available: https://simpy.readthedocs.io/en/latest/

[14] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. Wong, F. Tufvesson, V. Öwall, and O. Edfors, "The world's first real-time testbed for massive MIMO: Design, implementation, and validation," *IEEE Access*, pp. 9073 – 9088, 2017.

[15] H. Peng, W. Tärneberg, E. Fitzgerald, and M. Kihl, "Massive MIMO pilot scheduling over Cloud RAN," in *16th Swedish National Computer Networking Workshop (SNCNW 2020)*, may 2020.