Is Cloud RAN a Feasible Option for Industrial Communication Network?

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

Original scientific article

***Abstract*—Cloud RAN (C-RAN) is a promising paradigm for the next generation radio access network infrastructure, which offers centralised and coordinated base-band signal processing in a cloud-based BBU pool. This requires extremely low latency responses to achieve real-time signal processing. In this paper, we analysed the challenges to introduce cloud native model for signal processing in C-RAN. We studied the difficulties of achieving real-time processing in a cloud infrastructure by addressing its latency-constraint. To evaluate the performance of such a system, we mainly investigated a massive MIMO pilot scheduling process in a C-RAN infrastructure under a factory automation scenario. We considered the stochastic delays incurred by the cloud execution environment as the main constraint that has has impact on the scheduling performance. We use simulations to provide insights on the feasibility of C-RAN deployment for industrial communication, which has stringent criteria to meet Industry 4.0 standards under this constraint. Our experiment results show that, concerning a pilot scheduling problem, the C-RAN system is capable of meeting the industrial criteria when the fronthaul and the cloud execution environment has introduced latency in the order of milliseconds.**

***Index Terms*—Cloud RAN, Massive MIMO, Latency Constraint fronthaul, MAC scheduling, Industry 4.0.**

I. Introduction

As it is depicted in [1], Cloud RAN (C-RAN) is an intriguing candidate Radio Access Network (RAN) architecture for Fifth Generation Wireless Specifications (5G) that enables softwarization and resource centralisation in radio access networks and promises to provide mobile Internet access with low cost and highly efficient network operations. In the context of Industry 4.0, the communication networks are expected to evolve towards wireless communication, but with characteristics of high reliability, high capacity, large throughput and low latency. All these features are expected to be furnished by 5G. Additionally, by introducing C-RAN infrastructure, industrial communication networks can benefit by the advantages of cloud computing, such as low installation and maintenance cost, scalable service delivery, et cetera.

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

C-RAN aims to exploit the IT cloud computing technology in telecommunication network operations. The cloud computing features like load balancing, scaling and parallelism could be highly beneficial for the coordinated network operation. Likewise, deploying RAN processing in a cloud infrastructure significantly reduces the Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) for Mobile Network Operators (MNOs).

Massive Multiple Input Multiple Output (MIMO) is another essential enabler for the next generation RAN that significantly increases the system capacity in order to handle the rapid growth of traffic in mobile networks. The key feature of massive MIMO system is that it has more number of antennas than the User Equipments (UEs) to be served simultaneously. However, these large scale antenna systems require a huge amount of computational power for base-band signal processing. Therefore, it would be beneficial to adopt massive MIMO in C-RAN and to split part of the processing functionalities to a remote BBU pool.

As of now various challenges remain to be solved in order to deploy C-RAN for the next generation mobile networks as explained in [2], [3]. One important challenge is to establish low-latency communication between the BBU pool and RRHs. Considering massive MIMO as RRH infrastructure of C-RAN, offloading the computational resources of such large antenna systems to a remote BBU pool implies that, the fronthaul links may suffer from bandwidth and latency limitations while transmitting enormous amount of data [4].

Furthermore, instead of using Digital Signal Processors (DSPs) as computational units, C-RAN systems build on cloud-native technologies that makes use of General-purpose Processors (GPPs), which may also have adverse effect on real-time signal processing. Likewise, the virtualisation technology that enables cloud computing introduces more layers on the data path along the processing chain. These characteristics of the C-RAN system incur more uncertainties in the RRH-BBU communication. All these could also introduce catastrophic interruptions in the real-time signal processing

Manuscript received January 20, 2021; revised April 16, 2021. Date of publication April 29, 2021. Date of current version April 29, 2021.

The authors are with the Department of Electrical and Information Technology, Lund University, 22100 Lund, Sweden. Emma Fitzgerald is also with the Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland (e-mails: {haorui.peng, william.tarneberg, emma.fitzgerald, maria.kihl}@eit.lth.se).

Part of this work was presented at the International Conference on Software, Telecommunications and Computer Networks, SoftCOM '20, Hvar, Croatia, September 17-19, 2020.

Digital Object Identifier (DOI): 10.24138/jcomss-2021-0017

[5]. Therefore, the fronthaul links and computational components of C-RAN must comply with the stringent latency requirements to support the signal process in RANs.

In order to deploy C-RAN infrastructure with massive MIMO for industrial automation networks and meet the stringent performance requirements, we must show that the latency in the system does not collapse the network function performance and that the impact of the delay can be mitigated with simple strategies. This system is only viable if the massive MIMO signal processing chain can guarantee that the performances in terms of communication reliability and connectivity still meet the industrial criteria when the data transmission between two functions are delayed.

A number of studies have been performed to achieve the latency requirements of various industrial applications, as well as to take advantages of cloud computing. In [6], the authors proposed a dynamic switching solution between local computers and edge cloud according to network conditions of a multitier control system. A time-sensitive model to deploy machine learning applications for industrial cyber-physical system was proposed in [7]. The system was deployed in a fog architecture, so that the machine learning model can be executed in the vacinity of end-users.

As both massive MIMO and C-RAN are the most competitive candidates for building up the infrastructure of future mobile radio access networks, investigations on the combination of the two techniques have received a lot of interest. In [8], [9], the functionality split in massive MIMO RRH C-RAN system is 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 and 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 [10] to compress the link data rate over the fronthaul and to keep the RRH structure as thin as possible. In [11], pilot contamination and imperfect channel estimation are considered as the impacts of the limited fronthaul. In [12], the authors proposed a decision-theoretic framework to tackle the delayed Channel State Information (CSI) for a rate allocation problem in C-RAN and optimize the end-to-end TCP throughput performance for the mobile edge cloud users. In their formulation, the TCP response latency experienced by the users is considered as a constraint and only a low mobility scenario is addressed.

To the best of our knowledge, very little research has addressed on the feasibility of establishing Massive MIMO C-RAN for industrial communication, especially not from the perspective of a MAC layer function, such as pilot scheduling. Likewise, few have considered the cloud execution environment of C-RAN as the main constraint in their problem, which, however, significantly affects both function performance and user experience.

This is an extended paper of [1], in which we targeted the performance evaluation of a C-RAN system with delays under an industrial scenario. In this paper, we first give a detailed discussion about the aspects that introduce the delays in the system, by analysing the path of workloads in a cloud execution environment and presenting our measurements on the delays from different cloud environments. In this context, we implemented the pilot scheduling function at the Medium Access Control Layer (MAC) layer of massive MIMO in the cloudified BBU pool of the C-RAN system. We focus on the feasibility of deploying such a system under industrial automation requirements from the perspective of the scheduling performance, which is affected by several factors of the system. For the investigation, we applied a commonly used Earlist Deadline First (EDF) strategy on the pilot scheduling problem to evaluate a latency constrained system using simulations. Our investigations show that C-RAN is capable of providing a reliable communication infrastructure that meets the criteria of industrial automation.

II. Background and Challenges

In this section, we briefly introduce the background of C-RAN and address the challenges of deploying C-RAN for industrial communication regarding the cloud executing environment of the BBU pool.

It takes a phased approach to softwarize and fully integrate cloud technologies to C-RAN systems as envisioned by [13]. At the first phase of C-RAN, the traditional distributed base-band processing units (BBUs) are detached from the radio-frequency processing units (RRHs). The remaining RRHs are co-located with the antenna while the centralised BBU is responsible for the base-band processing of the RRHs but is not pooled virtualised. The centralised BBU is connected to the target RRHs by fronthaul links.

At the final phase of C-RAN, the computational resources of BBU are pooled and becomes a so-called BBU pool. This phase of C-RAN will take advantages of Software Defined Radio (SDR), leverage virtualisation technologies, make use of GPPs instead of DSPs and approach towards real-time base-band processing in the fashion of cloud computing model. All things considered, C-RAN is a candidate architecture for future Radio Base Stations (RBSs), in which the signal processing functions are partially or fully deployed in the centralised BBU pool building on virtualisation and cloud technologies. However, the cloud execution environment of the BBU pool is usually non-deterministic and unpredictable, which is contrary to low-latency and ultra-reliable virtue we long for from 5G RBSs and industrial automation network. Deploying a real-time pilot scheduling function in the cloud may suffer from the negative impacts of its execution environment.

Unlike DSPs, the computing resources in cloud-based BBU are pooled to serve the customers with a multi-tenant model, which can introduce uncertainties to computing performances. The client of a cloud service has no control on the location of the provided resources, neither the knowledge of other tenants sharing the same physical resource.

Resource pooling of the BBU are achieved by virtualisation, which is the key technology leveraged by cloud computing. Virtualisation brings the benefits of higher flexibility, faster resource provision, cost reduction and higher resource utilisation, however, at the cost of performance suffering. As it adds abstraction layers on top of physical machine, yielding

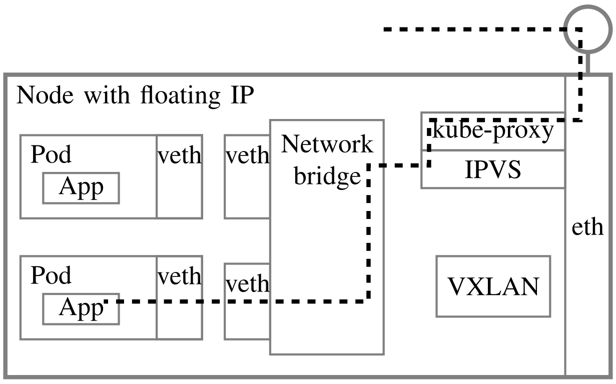


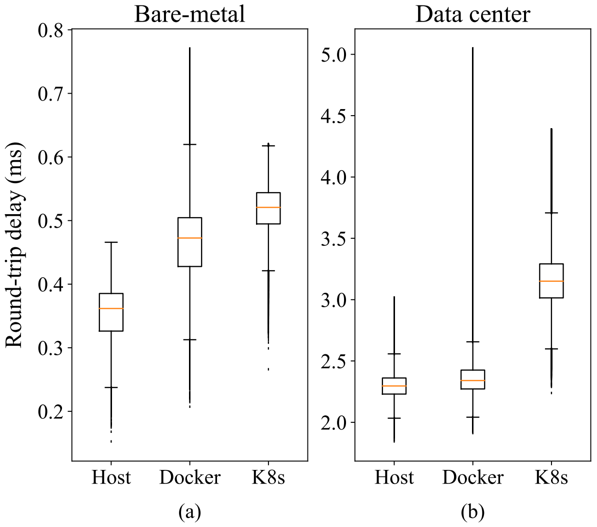
Fig. 1. An example of workload path to reach a service in a typical cloud environment through Kubernetes NodePort deployment. Figure from [17].

a longer path of the workloads than that in a bare-metal server. Researches and experiments have shown performance degradation on I/O, networking and memory access with virtualisation [14]–[16].

Besides the virtualisation technology, the cloud-native deployment also brings overheads on the processing function. We hereby take into account of an application running on a standalone Docker[1] container. As a container is the smallest necessary computing unit in cloud-native deployment and it shares the Linux kernel with the host machine. The inbound/outbound traffic of a container is forwarded by a software bridge. However, a container is unaware of itself running on an abstraction layer, but sees itself as a normal machine with full network stack. Thus the host machine of Docker container needs to encapsulate incoming packets with IP headers in order to direct them to the destined container.

If we consider Kubernetes [2] deployment, which is a typical and most popular orchestration platform for container deployment, management and scaling. Briefly speaking, Kubernetes wraps containerised applications into Pods on Nodes, where a node is virtual machine or physical machine. When deploying cloud service on top Kubernetes, where the networking functionalities are provided by Container Network Interface (CNI), more hops are added along the path of workloads to reach the end-point of a service. Fig. 1 is an example given by [17] on how the workload traffic is directed to the service end-point (the scheduling function) in a Kubernetes node when having a NodePort[3] type of service. Additionally, if several functions are in the chain, which are deployed but hosted by different nodes in a cluster, the workload will be directed over the Virtual Extensible LAN (VXLAN) to the next end-point.

This overall brings overheads on performances in terms of networking by adding extra layers on the path of workloads, which is detrimental to the real-time signal processing function chain over C-RAN. All this would lead to, from the perspective of a client (the RRH), a longer and uncertain response time from the application (the pilot scheduling function), and further cause performance degradation on the UEs as they relies on the pilot allocation from the system for their data transmission.



(a)

(b)

Fig. 2. The UDP latency measurements on response time to different type of services deployments, 1.5% outliar removed

To give an intuitive illustration of the delay incurred by the cloud execution environment of C-RAN, We show in Fig. 2 our measurements on response time of a simple "UDP ping" application when it is deployed in (a) a bare-metal cluster in the same Local Area Network (LAN) with the client machine, and (b) in a cloud date centre residing in the same city. Compared to the C-RAN system architecture, the client machine in this set up represents the RRH, and where the the application end-point is running on is the BBU pool. The network between the "RRH" and the "BBU pool" are (a) 1Gps Ethernet and (b) 1Gps Ethernet and Wide Area Network (WAN). Under each scenario, We compared the application as running in a host machine, in a stand-alone Docker container and in a Kubernetes Pod. As depicted in Fig. 2, for the same application, the mean response time and its variances have increased as it is running on a host, a docker container and a Pod in Kubernetes cluster. We also see that the Kubernetes deployment in a cloud data centre has more overheads than in a bare-metal cluster, as we don't know about the networks between two node.

In short, introducing the cloud deployment for the pilot scheduling function would have more challenges, as extra abstraction layers are added to the path of the workload traffic between the RRH and the BBU pool. That is when the scheduling decisions made by the BBU pool arrive at the RRH, it may not be applicable for the state of the Critical Unitss (CUs) due to the outdated information exchange caused by the delays. In this paper, we aimed to investigate whether the C-RAN deployment is still feasible when such delay is incurred by the cloud execution environment.

III. Targeted System

In this paper, we target a C-RAN architecture described in [1], which includes one cloud-based BBU pool and one

[1] https://www.docker.com

[2] https://kubernetes.io

[3] https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types

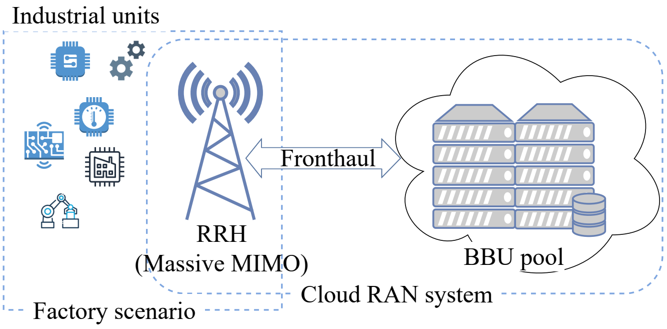


Fig. 3. Target system architecture. Figure from [1].

massive MIMO RRH, connected with a fronthaul link, shown in Fig. 3. As the MAC layer scheduling function is the main focus of our problem, we assume that the Physical Layer (PHY) functionalities are operated on the RRH and no raw base-band data blocks are transmitted over the fronthaul link.

A. C-RAN System

In this paper we address a C-RAN system that provide wireless communication for manufacturing processes. The RRH is co-located with the manufacturing plant and connected to the BBU pool via fronthaul link. For a manufacturing process, the communication distance is normally less than 100m [18], thus we assume that all the units can be covered by the radio range of one RRH in our target scenario.

We mainly address the inevitable delays incurred by the GPP cloud execution environment of the BBU pool. In Section II and Section V-1, we show that the round-trip delay caused by the cloud execution environment could be up to milliseconds. This is much larger than the fronthaul link latency requirement of C-RAN, which is 5µs to 400µs [19]. Therefore, the cloud environment would impose more interruptions in the processing chain of a function that is deployed across C-RAN.

B. Massive MIMO and Radio Resources

We use massive MIMO as the RRH of our target system. 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 where channel frequency response is approximately constant for a UE . 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 the CSI of each UE, which is needed for precoding to process the input and output data [20]. Thus, in every coherence interval, a new pilot is needed for a given UE to transmit data successfully. In this paper, we consider the uplink pilots as the resources required by the UEs in an industrial automation scenario before a transmission can start.

The length of a coherence interval mostly depends on the UEs' mobility when the carrier frequency is fixed [20]. A UE with lower moving velocity yields a longer interval, therefore it requires fewer pilots to transmit the same amount of data compared to one with higher mobility.

C. Industrial Communication Network

We address an indoor industrial automation scenario, where there are numerous sensors, controllers and actuators, here called CU, which are part of a dynamic control system and are interconnected by a wireless industrial network. The traffic generated by the control operations with these units has key requirements such as less than 10ms latency, availability within the range of 95%-99.999% and density of 10000 devices per km^2, but the mobility of these units are mostly fixed or very low, since there is usually an indoor environment for industrial automation [21].

Because of the processes' low latency requirement, in this paper we assume that each transmission request has a hard deadline. If a unit has not been assigned a channel resource within the deadline, the transmission attempt failed and the data is discarded. Also, as most units have low mobility in the scenario, the coherence interval in the massive MIMO time-frequency space can be relatively long, and thus, a larger number of units can be served by the radio system simultaneously.

To complicate things, there are many types of units in such a system, some with less stringent requirements and thus, the priority of such units is lower than the CUs. The traffic generated by these units is considered as background traffic in the system. Therefore, it is important to optimize the radio resources allocated to the prioritized traffic from CUs, since the remaining resources can be allocated to the low-priority background traffic.

D. Pilots Scheduling Strategy

In our targeted industrial setting, the requests from CUs have strict deadlines but the number of pilots in a coherence interval is limited. In order that the CUs get assigned the pilots for data transmissions within their deadlines, we need to deploy a MAC scheduler to allocate the pilots. In our targeted C-RAN system, the scheduler is located in the BBU pool. The objective of the scheduler is to serve as many requests as possible within their deadlines. The background traffic will be served if there are pilots left in each coherence interval after the requests from CUs have been scheduled.

When allocating pilots to the CUs, the massive MIMO RRH follows the decisions made by the remote scheduler to allocate the pilots to the CUs. In order to investigate the feasibility of C-RAN deployment for industrial automation scenarios, we applied a scheduling strategy with EDF policy on the MAC layer to allocate the pilots to the CUs. The EDF policy guarantees that the CUs whose requests have earliest deadlines get the pilots first.

We propose the two following performance metrics for investigating how massive MIMO pilot scheduling is affected by the C-RAN constraints.

Loss (L): A request is dropped if it is not scheduled within its deadline. The loss can be calculated as the ratio between the dropped transmissions and the total number of requests.

Pilot utilization (U): A pilot is wasted every time it is allocated to a CU that has nothing to send. The utilization of pilots can be calculated as the ratio between the pilots that are

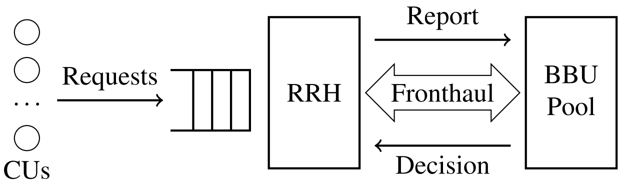


Fig. 4. Simulation model.

successfully assigned for transmission requests and the total number of pilots that are allocated.

IV. Simulation Model

In this section, we present the system simulation model shown in Fig. 4, given that in total K active CUs are covered by the radio range of the RRH. The RRH communicates with the BBU pool via the fronthaul link in order to allocate pilots to the CUs.

A. RRH and BBU Pool Model

We consider that each CU only needs one pilot for the base station to estimate its channel state information in order to serve the transmission requests in a coherence interval. We assume that the number of available pilots in an interval is proportional to the length of the interval, which is determined by the mobility of the CUs. Since the end-users can be multiplexed in the spatial domain in massive MIMO, if a given CU gets assigned a pilot, we consider that the number of its requests that can be served is also proportional to the interval length.

We denote the minimum interval length of our system as T_c, during which p pilots are available, implying that maximum p CUs can be assigned the pilots during T_c and one transmission request from each CU is served if it is assigned a pilot. We also denote by T_{slot} the actual length of a coherence interval, as well as as an allocation time slot in our scheduling problem, and there are P pilots available during each slot. When $T_{slot} = T_c$, we call it a high mobility scenario. When T_{slot} increases, it yields that the units in the scenario have lower moving velocity, and the number of available pilots P during T_{slot} increases proportionally.

The RRH keeps an ingress queue of all the active transmission requests. The BBU is able to keep track of the status of this queue. Every time the BBU gets updated queuing information, it sends a new scheduling decision so that the RRH could apply the updated allocation policy to the active CUs.

B. Traffic Model

Each CU$_k$, where $k \in \{1, 2, ..., K\}$, sends out the transmission requests at an average rate λ_k. We take into account the industry and IoT source level traffic models summarized in [18]. We use *Homogeneous periodic traffic* as the arrival process to generate transmission requests. By following this arrival process, each CU sends out requests with a nearly constant period around c but with a normally distributed noise, implying the average arrival rate of CU$_k$ is $\lambda_k = 1/c$. Each request has the following features:

- The CU ID k, indicating it is a request made by CU$_k$.
- The count γ, indicating it is the γth request made by CU$_k$.
- Deadline D_k^γ. The deadline length of CU$_k$ is sampled from an uniform distribution between c and D, where D is the bound of deadline lengths of all the CUs.

The overall average arrival rate to the system is then the sum of all sources $\lambda = K/c$. The offered load to the system only depends on the number of active CUs K in the scenario.

C. The Scheduling Policy

At the beginning of every coherence interval, the RRH sends the information of all the active requests in the ingress queue to the BBU pool. We denote the information sent by the RRH as the *report* in our model, which contains the CU ID and the deadline (k, D_k^γ) of all the active requests in the queue.

When the BBU pool receives a new report, it inspects the active request information in the queue and makes the corresponding *decision*, which is a set of CU IDs $\mathcal{K} \subseteq \{1, 2, 3, ...K\}$ to which the pilots are assigned. If the number of CUs with pending transmissions in the ingress queue is less than the available pilots P, it assigns all the CUs in the queue a pilot. If the number of CUs is greater than P, the EDF algorithm will be applied to allocate pilots to the P CUs whose requests have the earliest deadlines.

D. Fronthaul and Latency Model

The fronthaul link will cause a delay of each message sent over it. The round-trip delay of the fronthaul link is modeled as the duration from when a report departs to when the corresponding decision arrives at the RRH, but neglecting the computation time in the BBU pool for making the decision. The round-trip delay is modeled with a log-Laplace distribution with mean μ milliseconds. Our motivation for this choice is described in Section VI-A.

E. Performance Metrics

In this section, we detail the performance metrics: *loss* and *pilot utilization*. The pilot utilization is calculated as follows. Given a time slot j, the RRH takes a decision that $\hat{P}_j$ pilots should be assigned to the CUs in set $\mathcal{K}_j$ waiting in line, where the length of set $\mathcal{K}_j$ equals to $\hat{P}_j$, $\hat{P}_j \leq P$ and $\mathcal{K}_j \subseteq \{1, 2, 3, ...K\}$. For each CU in set $\mathcal{K}_j$, the number of transmission requests that can be served is T_{slot}/T_c, as it is proportional to the coherence interval length. We denote the actual number of active requests from CU$_k \in \mathcal{K}_j$ in the queue by $N_{k,j}$. This means that in a time slot j, the number of wasted pilots $W_{k,j}$ for CU$_k$ is:

$$W_{k,j} = \begin{cases} 0 & \text{if } N_{k,j} \geq T_{slot}/T_c \\ \frac{T_{slot}/T_c - N_{k,j}}{T_{slot}/T_c} & \text{if } N_{k,j} < T_{slot}/T_c \end{cases} \tag{1}$$

This yields the pilot utilization in slot j:

$$U_j = 1 - \frac{\sum^{\forall k \in \mathcal{K}_j} W_{k,j}}{\hat{P}_j T_{slot}/T_c} \tag{2}$$

Taking the length of one simulation as T, the pilot utilization during the whole service period is:

$$U = 1 - \frac{\sum_{j=1}^{T/T_{slot}} \sum^{\forall k \in \mathcal{K}_j} W_{k,j}}{\sum_{j=1}^{T/T_{slot}} \hat{P}_j T_{slot}/T_c} \tag{3}$$

Denoting the actual number of requests from CU$_k$ being served in time slot j as $S_{k,j}$, the average loss of the system during T is given by:

$$\bar{L} = 1 - \frac{\sum_{j=1}^{T/T_{slot}} \sum^{\forall k \in \mathcal{K}_j} S_{k,j}}{\sum_{k=1}^{K} \lambda_k T} \tag{4}$$

where $S_{k,j} = \min(T_{slot}/T_c, N_{k,j})$.

We denote this as $\bar{L}$ because it is calculated from the mean arrival rate λ_k of each CU. In the simulation experiments, we measured the actual number of transmission requests in the system to calculate the loss L.

V. System Evaluation

In this section, we present the experiment setup and the parameter values we used in the simulations to investigate the feasibility of deploying a C-RAN system in an industrial automation scenario. The simulation is implemented in SimPy[4]. We ran all the experiments to simulate a system time of $T = 200\,000$ms and there are 20 repetitions for each parameter set.

1) Latency: In our simulation, we used a Log-Laplace distribution to generate the round-trip delays, which, as will be shown in Section VI-A, is empirically modelled from our measurements took from the system setup described in Section II. The mean μ of the round-trip delay varies from 0.5ms to 15ms, but the other distribution parameters remain the same for all experiments.

2) Loss: To evaluate if the C-RAN system can meet the minimal requirements from the industrial standards, here, we set the maximum permissible loss to 5% for all the transmission requests.

The loss is highly related to the CUs' tolerance on the waiting time to get a radio resource, and therefore we ran experiments with the objective of investigating the maximum round-trip delay that the CUs can tolerate when they have different deadlines. We set the variables of the CUs arrival process as shown in Table I. We choose a medium mobility scenario in this evaluation and the corresponding variables under this mobility scenario can be found in Table II. The same parameter configurations are used in [1].

To invetigate the maximum number of CUs that the system can serve under different mobility scenarios, we also ran the experiments when all CUs have deadline lengths the same as their arrival intervals indicated in Table I. We set the round-trip delay in this evaluation as 3ms, which, as will be shown in Section VI-A, is slightly larger than our latency measurements from the aforementioned experiment setup.

TABLE I
Arrival Process Parameters for the Evaluation on Tolerable Round-trip Delay. Table from [1].

Parameter name	Value	Symbol
Arrival interval	10 ms	c_k
Number of CUs	20	K
Deadline length bounds	$\{5, 6, 8, 10, 12, 15\}$ ms	D

TABLE II
Parameters Related to Different Mobility Scenarios in the Simulation. Table from [1].

Mobility scenario	High	Medium	Low
Coherence interval length T_{slot}	0.5ms	1ms	1.5ms
Available pilots per interval P	12	24	36

3) Pilot Utilization: The pilot utilization becomes important once the requirement of loss is met. It is obvious that the loss decreases if the scheduler allocates redundant resources to the CUs. However, this could mean that the background traffic, which has lower priority than the CU traffic, may be faced with resource starvation due to pilot waste. Thus we should consider pilot utilization under a low loss case, in which the length of deadlines has very little impact on the utilization, but the length of the coherence interval, or the CUs' mobility, becomes the dominating factor. Thus we ran the experiments under different mobility scenarios but with the parameters of the CUs' arrival processes the same as in Table I, except for the deadline length, which in this case has an upper bound fixed to 15ms. The longest round-trip delay is set to 8ms, in which case there are rare discarded requests in the system for this deadline. loss

VI. Experiment Results

In this section, we show our latency measurements and the simulation results regarding the two performances metrics *loss* and *pilot utilization* by following the evaluation setup. The results under the same system configuration are also discussed in [1].

A. Round-trip Delay

Fig. 5 shows the histogram of our round-trip delay measurements. This also refers to the measurements of the Docker application in Fig. 2(b). We fitted the histogram to a log-Laplace distribution with mean value $\mu \approx 2.38$ms. This is a long-tailed distribution, which is not just incurred by the separation between the RRH and the BBU pool, but also by the cloud execution environment.

B. Loss

Fig. 6 shows the maximum round-trip delay the system can tolerate so that the loss is under 5% when the CUs have the arrival processes indicated in Table I . As we can see from the Fig. 6, the tolerable delay is always 1-3ms less than the deadline length. If one expects each CU to have a deadline the same length as its period, the round-trip delay incurred by the

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

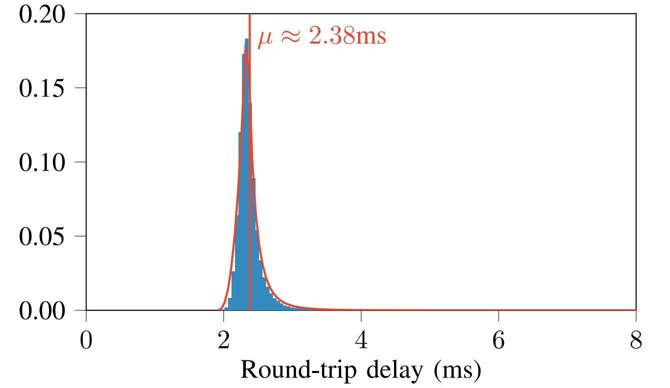


Fig. 5. The histogram of the UDP round-trip delay measurements. The red curve is the probability density function and the mean value fitted from the histogram. Figure from [1].

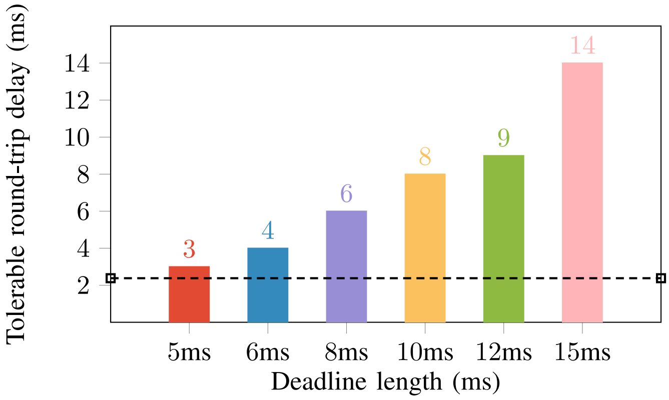


Fig. 6. The tolerable round-trip delay with varying CU deadline lengths. There are in total 20 CUs, all with medium mobility. The dashed line indicates the mean round-trip delay from our measurements shown in Section VI-A. Figure from [1].

C-RAN system can not be longer than the CU's transmission interval.

Fig. 7 shows the maximum number of CUs that the system can serve within the allowable loss of 5%, when all CUs have deadline length of 10ms and the round-trip delay in the system is 3ms. As can be expected, the system can serve more units when the mobility is lower (that is, when the coherence interval is longer). We can conclude from the figure that when the units have low mobility, the system can handle a higher offered load from the CUs without loss than when in a higher mobility scenario.

C. Pilot Utilization

Fig. 8 shows how the pilot utilization is affected by the CUs mobility and the delay. When there is no delay in the system, a short coherence interval can achieve full pilot utilization. But as the interval gets longer, the utilization of the resources drops significantly to only 40% when there is only 0.5ms round-trip delay in the system. This is because when the allocation slot is shorter, the decisions are more frequently made so that they can better follow the dynamics of the ingress

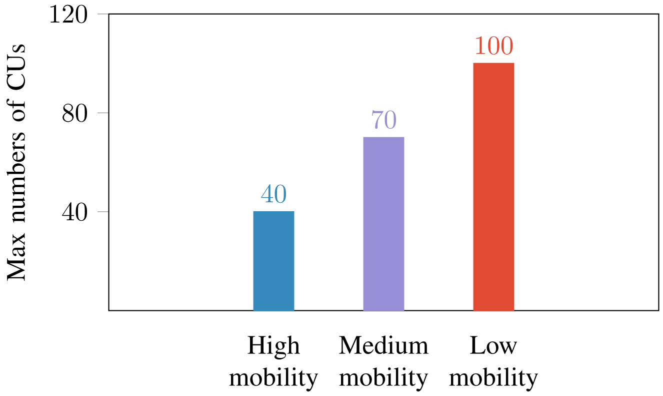


Fig. 7. Maximum number of CUs the system can serve within the allowed loss of 5% under different mobility scenarios. Each CU has a deadline length of 10ms and the system round-trip delay is 3ms. Figure from [1].

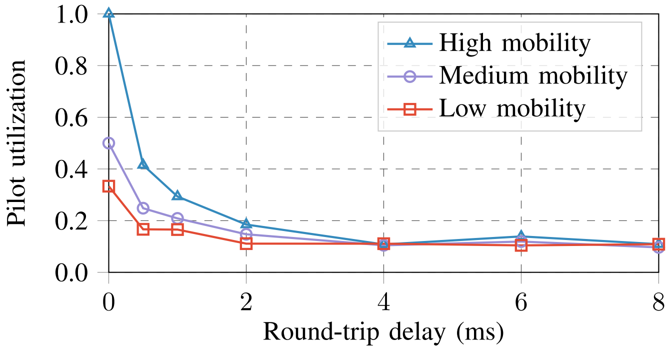


Fig. 8. The pilot utilization when the number of CUs is $K = 20$ and each has a deadline length between 10 and 15ms. Figure from [1].

queue. However, having longer intervals means more time-frequency space resource are reserved for the same set of CUs in each slot. Since the transmission periods of the CUs are usually longer than the length of an allocation slot, this leads to redundant allocations when the number of the pending requests in the queue is less than which can be served by the system. However as the round-trip delay between the RRH and BBU pool increases, which may cause outdated reporting about the queuing status, the pilot utilization converges to only 10%. In this case, the length of coherence intervals has less impact, since the misreporting due to the latency causes faulty allocations, in which case a CU is allocated a pilot according to the latest arrived decision, even though all its requests were already served by previous decisions.

VII. Conclusions

In this paper, we addressed a C-RAN system that is built on the GPP cloud, which imposes inevitable latency issue for a function deployed across the system. We provided an overview on the challenges brought by the cloud execution environment to C-RAN deployment. Based on the characteristics empirically modelled from the response time measurements over a cloud application, we used simulation to demonstrate the feasibility of deploying such a system under industrial criteria.

We considered a pilot scheduling function for industrial critical units that have stringent requirements on the deadlines. The scheduling function is hosted in the BBU pool but the pilots need to be allocated to CUs by the RRH. We focused on two performance metrics *loss* and *pilot utilization* and applied a simple EDF scheduling policy to evaluate if the system can cope with the delay between the scheduling function and the allocation. We performed a simulation to investigate the behavior of the system in different scenarios.

Our experiment results have shown that the C-RAN system is feasible to deploy for the industrial automation scenario, where the CUs can tolerate round-trip delays up to 2ms less than their own deadlines. For a massive MIMO RRH, lower mobility end-users lead to a longer coherence interval and bring lower loss, implying that when the units' mobility is low in the scenario, the system is capable of serving a higher number of CUs simultaneously. On the other hand, both delay and a longer coherence interval lead to a huge amount of resource waste, which may lead to resource starvation of the background traffic. The next step of our work is to develop a new scheduling strategy to avoid redundant and faulty allocation so that the resources can be better utilized and the system can meet more stringent reliability requirements in industrial communication.

Acknowledgement

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.

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)*, 2020, pp. 1–6.

[2] 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.

[3] 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.

[4] 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.

[5] W. Tärneberg, "The confluence of cloud computing, 5G, and IoT in the fog," Ph.D. dissertation, Department of Electrical and Information Technology, Lund University, March 2019.

[6] Y. Ma, C. Lu, B. Sinopoli, and S. Zeng, "Exploring Edge Computing for Multitier Industrial Control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3506–3518, nov 2020.

[7] P. O'Donovan, C. Gallagher, K. Bruton, and D. T. O'Sullivan, "A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications," *Manufacturing Letters*, vol. 15, pp. 139–142, jan 2018.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] China Mobile, "C-RAN: the road towards green RAN," pp. 15–16, 2011, Accessed on Mar. 9, 2020. [Online]. Available: https://www.semanticscholar.org/paper/C-ran-the-Road-towards-Green-Ran/eaa3ca62c9d5653e4f2318aed9ddb8992a505d3c

[14] S.-H. Ha, D. Venzano, P. Brown, and P. Michiardi, "On the impact of virtualization on the I/O performance of analytic workloads," in *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)*. IEEE, May 2016.

[15] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in *2010 Proceedings IEEE INFOCOM*. IEEE, Mar 2010.

[16] U. Drepper, "The cost of virtualization: Software developers need to be aware of the compromises they face when using virtualization technology." *Queue*, vol. 6, no. 1, p. 28–35, Jan. 2008.

[17] L. Larsson, W. Tärneberg, C. Klein, E. Elmroth, and M. Kihl, "Impact of etcd deployment on kubernetes, istio, and application performance," *Software: Practice and Experience*, vol. 50, no. 10, pp. 1986–2007, 2020.

[18] 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, pp. 1–8.

[19] 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.

[20] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge University Press, nov 2016.

[21] 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

Haorui Peng is currently pursuing the Ph.D degree at the Department of Electrical and Information Technology at Lund University, Sweden. She received the M.S. degree in Advanced Robotics from École Centrale de Nantes, France and Università degli studi di Genova, Italy, in 2017. Her main research interests are in cloud computing and cloud-integrated autonomous networked system.

William Tärneberg is a post-doc in networked systems at the Department of Electrical and Information Technology at Lund University, Sweden. His research interests include autonomous system, fog computing, control-over-the-cloud, intelligent networks, and quality-elastic computing. William received his PhD in 2019. and has worked with wireless computing and cloud computing since 2009. in both industry and academia. In addition to research, William is also supervising PhD-students and teaching at the department.

Emma Fitzgerald received her Bachelors degrees in computer engineering and mathematics from the University of Sydney, Australia, in 2008., and her PhD from the same institution in 2013., in the area of vehicular ad-hoc networks. She then joined the Department of Electrical and Information Technology, Lund University, as a postdoc in 2014. for two years, after which she has continued there and currently holds the position of Associate Professor. Her research interests lie in cooperative networking and network performance, with particular focus on the Internet of Things.

Maria Kihl is Professor in Internet worked systems at the Department of Electrical and Information Technology at Lund University, Sweden. Her main research interests are in the fields of performance modeling and analysis of networked systems and applications. Currently, her projects are mainly focused on networked systems in 5G, Industry 4.0, and cloud control.