Evaluation of Control over the Edge of a Configurable Mid-band 5G Base Station

Haorui Peng*, William Tärneberg*, Emma Fitzgerald*†, Fredrik Tufvesson*, 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, fredrik.tufvesson, maria.kihl}@eit.lth.se

***Abstract*—Mission-critical applications such as industrial control processes are evolving towards a new development paradigm by offloading their heavy computations to the edge of the emerging Fifth Generation Wireless Specifications (5G) network. In this manner, the applications can gain the economical and efficiency benefits of cloud computing, as well as reliable communication from the 5G network. However, the limited access to a configurable infrastructure of the 5G network and its edge computing infrastructure has restrained academic researchers from experimenting and validating their mission-critical application design under reasonable communication and computation scenarios. In this paper, we present a configurable mid-band 5G Stand-Alone (SA) deployment and demonstrate a control process that is running over the edge of the 5G network. We show in this paper a complete system setup for Control over the Edge (CoE) of the 5G network, and validate the feasibility of deploying similar mission-critical applications over the edge of 5G network.**

***Index Terms*—Control over edge, 5G, Mission-critical application, Mobile edge computing**

I. INTRODUCTION

As 5G is emerging and brings promising network characteristics such as ultra-low latency and extremely high bandwidth, Mobile Edge Computing (MEC) brings the advantages of cloud computing to the vicinity of mobile users. By migrating computations to the edge of 5G network, time-sensitive and mission-critical applications gain both mobility and computational offloading, which benefits especially Internet of Things (IoT) devices that have limited computational resources.

Time-sensitive and mission-critical applications such as feedback controllers are outgrowing their traditional production platforms and are migrating to the cloud where they can reap the benefits of economies of scale, and take advantage of offloading, collaboration, and efficient software development practices. But the cloud is a problematic environment for such applications because of the time-varying execution and communication delays that are inherent properties of cloud deployments and the intermediate Internet. This is a key challenge when rolling out the 4th industrial revolution [1], in which cloud technology is important. The edge is the notion that shared computing resources are placed at the edge of the network, proximal to the end-users, and are shared and orchestrated in a cloud-like manner. Deploying controllers at the edge allows them to take advantage of the cloud platform, but without the need to traverse a significant portion of the Internet, which incurs high latency and uncertainties.

Control over the Edge/Cloud (CoE/CoC) is an application category relying heavily on both client processes as well as network and computing infrastructure. The mission-critical nature of these kinds of applications necessitates research in the area to take real-life conditions into account, with special emphasis on Internet-level scalability and low latency under heavy, real-time usage. Experimental facilities deployed in isolated laboratory environments tend to be poorly suited for evaluating such a networked system's performance realistically [2], [3]. For conducting research on these applications, access to configurable commercial network infrastructures is preferable. Unfortunately, this is usually not possible, and it becomes a major obstacle for reliable experiments and validation of novel ideas.

Over the past decade, researchers have been developing systems with test environments corresponding to 4G/5G. For instance, the NITOS experimental portal [4], developed at University of Thessaly, Greece, offers open and remote access to experimental 4G networks, based on commercial and open-source Long Term Evolution (LTE) components. In their paper [5], the developers of NITOS point to the need for open, programmable, experimental facilities to implement and evaluate novel ideas in real-world settings, and summarize their efforts to build such a testbed. In [6], the authors integrate non-3GPP access to a disaggregated heterogeneous BS complying with the Cloud-RAN concept, and serve end users with two different link technologies, WiFi and LTE. In [7], the authors suggest a novel placement of MEC services and apply their setup to a vehicular environment to improve the network performance in terms of latency for mission-critical data exchange. In both papers, the proposed systems are deployed on the NITOS testbed for evaluation.

Another framework called TRIANGLE [8] has been developed within the EU H2020 project "5G Applications and Devices Benchmarking" [9]. The purpose of the framework is to provide application developers and device manufacturers with a realistic 5G testbed on which to test and evaluate their products and services. TRIANGLE exploits existing FIRE facilities [10], adding 5G facilities as necessary. It provides a high-level test platform for application developers, device designers and researchers working with protocols at the higher layers of the network protocol stack. It consists of LTE small cells, WiFi access points, a virtual Software Defined Networks (SDN)-powered Evolved Packet Core (EPC) emulator, another emulator spanning from Radio Access Network (RAN) to the LTE core, and a traffic generator. In [11], the author claimed that the increasing popularity of Virtual Reality (VR) combined with the possibilities provided by the anticipated deployment of 5G would create a high demand by end-users in terms of quality of experience, which would be challenging. To address these challenges, VR applications must exploit MEC and 5G technologies. The paper reports an experiment to test such a VR mobile application together with a new MEC application server conducted on the TRIANGLE testbed.

To the best of the authors' knowledge, as of now, when telecommunication systems are deployed commercially worldwide, research on mobile edge application development and verification still relies on emulation or simulation test-beds. Limited access to a complete and configurable 5G Base Station (BS) has restrained academic research from discovering potential problems in realistic 5G network systems. The poorly understood characteristics of real 5G systems also put obstacles in the way of application design for academic researchers.

Therefore, we see the need for researchers to be provided access to the edge infrastructure of 5G to conduct experiments and validate their designs for mission-critical applications. In this paper, we present a configurable mid-band 5G SA deployment, evaluate its application-level latency characteristics, and demonstrate a mission-critical application running over the edge of the 5G network. The application is a control process with a plant and its controller. The plant can be seen as a client and the controller as a service. The edge of the presented system is co-located with the SA core network. The communication between the client and service at the edge are over HTTP, which is widely accommodated by contemporary web and cloud applications. Further, we use the application's end-to-end response time as a performance metric to show the feasibility of deploying mission-critical applications at the edge of 5G network. We mainly examine the typical features of an application that affect network traffic, as well as radio parameters such as Discontinuous Reception (DRX) timers, which have a large impact on the power consumption and communication latency of IoT applications.

II. CONTROL OVER THE EDGE

A conventional control system, as shown in Fig. 1, consists of a controller and the plant to be controlled. A system that is controlled over the cloud or edge has the same system components, but separated by a network and has its controller deployed at the edge of the network as a cloud service. The process operates at a frequency appropriate for it. In each period, the plant sends a request to the controller, at the edge, with its current state, which the controller uses to compute a control signal. In return, the controller synchronously replies with the control signal, which is then actuated at the plant.

The difference between a CoC/CoE process and a local control process is the network latency, which may cause obsolete control signals to be applied to the process under control. This can be detrimental to the performance and safety of the process. The request frequency of control processes depends on the models of different plants, which also affect the design of the controller. A critical control process requires a higher frequency to update its plant states and to get control signals to actuate. Thus, these critical processes require lower network latency as well as efficient computation in the cloud to meet the process dynamics. With the low latency and reliable 5G network described above and the adjacent edge server, we can get close to what is desired.

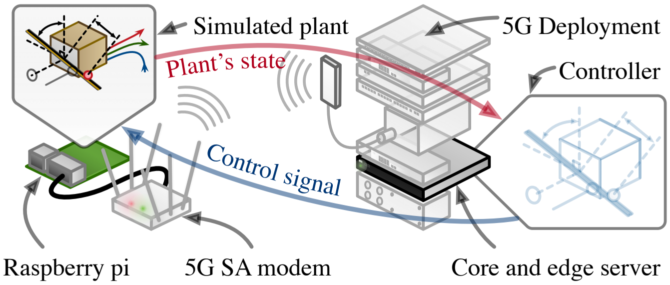

Fig. 1: Overview of evaluation system

III. SYSTEM DESCRIPTION

In this section, we describe the system we setup to evaluate the performance of CoE processes assisted by 5G. The evaluation system is shown in Fig. 1, which includes a 5G deployment that provides both the communication network and edge computing resources. The system also contains a User Equipment (UE) for running the plant of the CoE process.

A. 5G deployment

The evaluation platform used in this paper is a 5G NR SA mid-band Radio Base Station (RBS), equipped to operate within band n3 (1800 MHz) with a single antenna. The deployment is provided by Ericsson through a collaboration. We have access to reconfigure the radio and due to scalability, we have opted for an open-source core network deployment.

The deployment's components are laid out in Figure 2, in the order in which they are mounted in the deployment's rack. At the top is a power distributor that is fed from the power supply mounted at the bottom of the rack. Second from the top is a switch that interconnects the base-band units and the Operations, Administration and Management (OAM) server below, which manages the base-band and radio units. The OAM server is a local management server, through which the radio parameters in the base-band unit can be configured. The gNodeB consists of the base-band unit (Ericsson Baseband 6630) and an NR Frequency-Division Duplexing (FDD) radio (Ericsson Radio 2219). Attached to the radio is a passive indoor antenna. Second from the bottom is the core network server, which is served by the switch above it. For this paper, we have installed a C-lang implementation of 5G Core (5GC) and EPC (release 16). The core server also functions as our edge. With our own 5GC, the deployment is self-contained and connects directly to the Internet with an Ethernet cable.

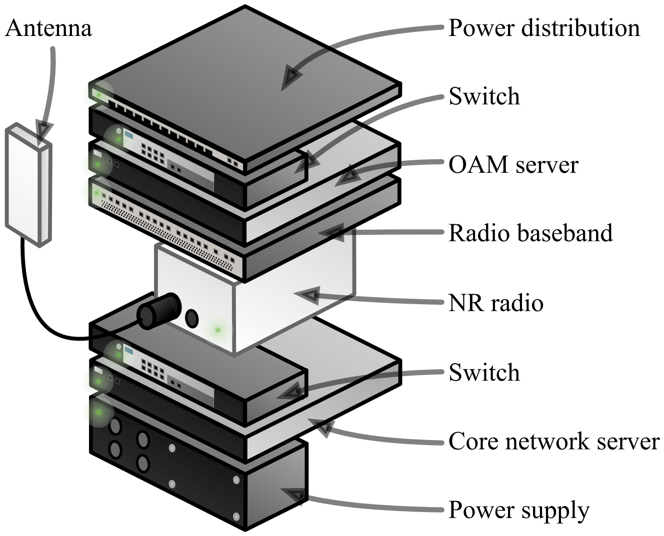

Fig. 2: The anatomy of the 5G deployment

B. Evaluation System Setup

The evaluation system of this paper consists of two agents, the *client* and the *service*. The UE where the client process resides is a Raspberry Pi 400 with Raspberry Pi OS, a lightweight Debian-based system. Since Raspberry Pi does not have embedded 5G connectivity, it is connected to a 5G mobile modem (WNC SKM-5xE) via its network interface. The modem is running in bridge mode, so that the Raspberry Pi can be assigned an IP address directly by the core network.

The service is running on the same machine which hosts the core network, so that it is deployed at the edge of the 5G network: no further network delays in transmission and processing will be introduced in the application. The response time of the CoE process in this case will only be dependent on the latency of the 5G network, application execution time, and higher layer protocol processing time.

IV. EXPERIMENT SETUP

To evaluate the expected performance of a mission-critical control process over the edge of a 5G network, we evaluate the application layer performance between the client and the service over our deployment. In this paper, the performance metric is the HTTP response time in milliseconds, because:

1) HTTP is one of the main deployed application layer protocols for communication with cloud services.
2) Compared to pure radio/core network benchmarking, end-to-end communication delay is the primary concern in the domain of control over the cloud/edge. Both software processing time and network delays contribute to the overall performance of an application.
3) The impact from the application layer on end-to-end delays is not negligible for many real-time processes; the computation time of a consuming process may even constitute a major part of the response time, especially in edge cloud, where computational resources are considered limited compared to a centralized cloud.

TABLE I: DRX parameters

Parameter name	**Value**
DRX long cycle	160ms
DRX inactivity timer	100ms
DRX on-duration timer	50ms

The evaluation of application performance is three-fold:

1) HTTP response time over 5G under different application sampling rate and payload size.
2) HTTP response time over 5G under the impact of DRX ON duration timer.
3) Performance of a control process running over 5G.

A. Performance of HTTP application running over 5G

We measure the response time of a simple HTTP application, whose client and service communicate via the 5G network provided by the deployment described in Section III. The response time measurements are collected at the application layer, which is measured at the moment when the HTTP response of each request arrives at the client process. The data analysis of each experiment is based on 50000 requests.

In order to eliminate the impact on the response time of the service execution time as much as possible, for the first two experiments, we run a simple HTTP 'ping' application. Here the client sends out an HTTP request to the service, with predetermined payload size and frequency. As such, at the service, execution time is minimal.

From the perspective of network analysis, both request frequency and payload size affect the data rate for throughput evaluation. But in our evaluation, network throughput is not the only concern, as the frequency is determined by the dynamics of the control process. It is non-trivial to evaluate the application performance under various frequencies for different payload sizes, so that the evaluation covers different application categories. Therefore, we examine the system with request frequencies from 200Hz to 10Hz, representing control processes with sampling rates from 5ms to 100ms. In the remaining part of the paper, we use request intervals varying in the range of $[5, 100]$ms to present the performance evaluation. The payload sizes of each packet are evaluated in the range between 64 bytes and 2048 bytes.

B. Performance of application response time affected by DRX

One group of configurable radio parameters in our 5G deployment is the DRX timers. DRX plays a major role in network latency in telecommunication systems. Enabling DRX is essential in current LTE and 5G systems [12], especially considering that the power utilization of UEs has become one of the main concerns in mobile communication with the advent of cellular IoT. The DRX mechanism controls the transceiver states of UEs with the on-duration timer, inactivity timer and DRX cycles, so that the UE battery life can be saved by entering idle mode when there are no packet transmissions [13]. But network delay in a DRX-enabled system also depends on the inter-arrivals of packet transmissions [13], implying that when a control process is running over a network system that has

DRX enabled, a higher request frequency may incur a lower impact from DRX cycles. Therefore, the DRX parameters need to be adjusted based on the applications that are running over the network.

Therefore, we also evaluated the performance of an HTTP application when DRX is enabled. The DRX parameters used in the evaluation are indicated in Table I. As DRX delays mostly depend on the request intervals and DRX timers, similar to Section IV-A, we run the same HTTP application with request intervals from 5ms to 100ms. The payload size is fixed to 1024 bytes, as it is the closest payload size to that of the requests in one of our deployed control process over the edge.

We compare the application response time to the process performance under the same application parameters but with DRX disabled, so that the impact of DRX on high frequency processes can be revealed by the comparison.

C. *Performance of a control process running over 5G*

We evaluate the performance of a control process running over 5G, with DRX disabled. Instead of a simple HTTP 'ping' application, we deploy a simulated Ball-and-Beam (BnB) control process, and examine the feasibility of running such a dynamical process over a 5G network system and the edge. The BnB process is a classic dynamical control process that keeps a ball balanced on a rotatable beam. In the evaluation, the sampling rate of the BnB process is fixed to 50ms to read the position of the ball and rotation of the beam. The payload size of each request is between 700 and 900 bytes. As for the aforementioned 'ping' process, the BnB plant is simulated at the client of the system, but the controller is a service that is co-located with the 5G core network and listens to HTTP requests. We here show the performance of such a CoE process by comparing it with the same control plant at the client, but only send requests to a service that is running on a machine which is connected to the UE via a wired network.

V. PERFORMANCE OUTCOMES

In this section we present the outcomes of the performance evaluation experiments described in Section IV.

A. *Response time vs. request intervals and payload size*

Fig. 3 shows the changes in mean value of the application response time when running with the request intervals and payload sizes given in Section IV-A. The 95th-percentile and 5th-percentile are also illustrated to show how the jitter of the response time is affected by the examined request characteristics. As we can see from Fig. 3, large payload size does not have a major affect on the average response time of an application, but the 95th-percentile significantly increases as the payload size increase, implying that more jitter was caused by the payload size. We can also see from the figure that lower request intervals cause large average application response times, as the data rate on the uplink grows when the request interval decreases. The 95th-percentile of the measurements also have major increases. Although both smaller request interval and larger payload sizes lead to higher data rate in the uplink, the request interval has more impact on the average response time performance of an HTTP application that is running over the edge of 5G.

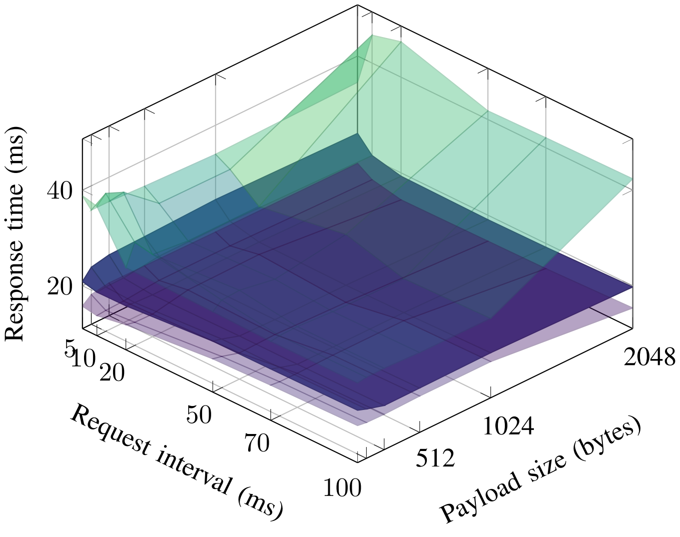

Fig. 3: Impact on HTTP Response time of request intervals and payload size

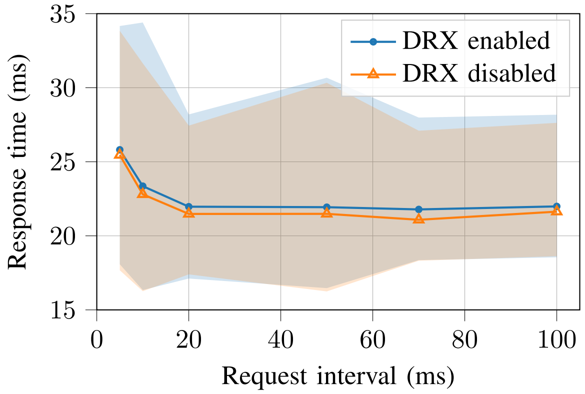

Fig. 4: Mean values of HTTP Response time when enabling and disabling DRX

B. *Application performance with DRX enabled*

In Fig. 4, we test the difference in the response time of an HTTP application as a result of DRX under increasing request intervals. From Fig. 4, we do not see any significant impact on the average response time and its 95th-percentile for the evaluated HTTP application. Similar to the case without DRX, when the request interval is less than 20ms, the average response time has a distinct drop as the interval increases. However, when the request interval is greater than 20ms, the response time of the applic The main reason for the indistinct DRX impact is that most control plants are highly dynamical processes, and the request frequencies of these processes are

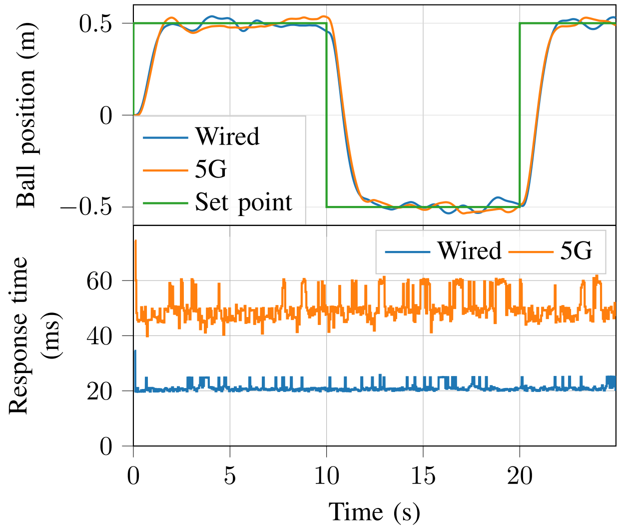

Fig. 5: Step response of BnB process with different controllers

relatively high compared to the DRX cycles in 5G systems. The delays caused by DRX mode depend on both the request frequency and the DRX cycle length. In the evaluation, the HTTP application, which can represent the characteristics of many critical control processes, is able to keep the UE in connected mode with high request frequency, thus the impact caused by DRX is mitigated.

C. BnB process when running

In Fig. 5, we illustrate the step response of the described BnB process and the response time of the process while running the controller (1) on a machine connected to the plant via a wired network and (2) on the edge of the 5G network. The top figure of Fig. 5 plots the step response of the ball position as the set point is changed between the two ends of the beam. The plot below illustrates the response time of each request sent from the BnB plant while the controller is running. As we can see from Fig. 5, due to the computation of the controller and the processing on the plant, it takes around 20ms for the plant to update its state and for the controller to compute the control signals. The introduction of the 5G network and the edge added around 30ms of delay to the application, as well as much higher variance of the delay. But from the step response on the ball position, the plant is well controlled to reach the set point with the support of the 5G system and its edge, even though delays are added to its response time to get the control signals.

VI. CONCLUSIONS

In this paper, we presented a performance evaluation of a real-time HTTP application that is deployed at the edge of a complete mid-band SA 5G base station. The end-to-end response time of the application was evaluated as to the feasibility of deploying a CoE process over the 5G network system. We found that, although the wireless communication system and computational process in the edge brought additional delays in application response time, the CoE process showed stable performance using the reliable 5G network system. The next step of the authors' work will be to deploy a scalable core network for the system in a cloud-native way, so that both core network services and edge applications for 5G are under the management of the same cloud orchestration system. In this manner, we will evaluate the impact and benefits that native cloud computing bring to the edge of current 5G network systems.

ACKNOWLEDGEMENT

We would like to thank Ericsson Lund for providing all the equipment of the 5G deployment. We also thank Fredrik Brumme from Ericsson Lund for building the 5G deployment for our lab. This work was partially supported by the Excellence Center at Linköping-Lund on Information Technology (ELLIIT), 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 IMMINENCE Celtic Next project funded by Sweden's Innovation Agency (VINNOVA).

REFERENCES

[1] J. Jasperneite, T. Sauter, and M. Wollschlaeger, "Why we need automation models: handling complexity in Industry 4.0 and the Internet of Things," *IEEE Industrial Electronics Magazine*, vol. 14, no. 1, pp. 29–40, 2020.

[2] R. V. de Omena., D. Santos., and A. Perkusich., "An approach to reduce network effects in an industrial control and edge computing scenario," in *Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER,*, INSTICC. SciTePress, 2021.

[3] 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*, nov 2020.

[4] "NITOS experimental portal," accessed: 2021-12-24. [Online]. Available: http://nitos.inf.uth.gr

[5] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassiulas, "Enabling open access to LTE network components; the NITOS testbed paradigm," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 2015.

[6] K. Chounos, N. Makris, and T. Korakis, "Enabling distributed spectral awareness for disaggregated 5G ultra-dense hetnets," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019.

[7] V. Passas, N. Makris, C. Nanis, and T. Korakis, "V2mec: Low-latency MEC for vehicular networks in 5G disaggregated architectures," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021.

[8] A. F. Cattoni, G. C. Madueño, M. Dieudonne, P. Merino, A. D. Zayas, A. Salmeron, F. Carlier, B. S. Germain, D. Morris, R. Figueiredo, J. Caffrey, J. Baños, C. Cardenas, N. Roche, and A. Moore, "An end-to-end testing ecosystem for 5G," in *2016 European Conference on Networks and Communications (EuCNC)*, 2016.

[9] "TRIANGLE project," accessed: 2021-12-24. [Online]. Available: http://triangle-project.eu

[10] "Federation of internet experimentation facilities," accessed: 2021-12-24. [Online]. Available: https://www.fed4fire.eu

[11] S. K. Datta, "Virtual reality mobile application testing in a 5G testbed," in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 2019.

[12] F. Moradi, "Improving DRX performance for emerging use cases in 5G," Ph.D. dissertation, 2021.

[13] C. S. Bontu and E. Illidge, "DRX mechanism for power saving in LTE," *IEEE Communications Magazine*, jun 2009.