Communication-Control Co-design for Robotic Manipulation in 5G Industrial IoT

Arvind Merwaday, Rath Vannithamby, Mark Eisen, Susruth Sudhakaran, Dave A Cavalcanti and Valerio Frascolla
Intel Labs, Intel Corporation
Email: {arvind.merwaday, rath.vannithamby, mark.eisen, susruth.sudhakaran, dave.cavalcanti, valerio.frascolla}@intel.com

Abstract*—Industrial IoT use cases have stringent reliability and latency requirements to enable real-time wireless control systems, which is supported by 5G ultra-reliable low-latency communications (URLLC) over cellular networks. However, extremely high quality-of-service (QoS) requirements in 5G URLLC causes huge radio resource consumption and low spectral efficiency limiting network capacity in terms of the number of supported devices. Industrial control applications typically incorporate redundancy in their design and may not always require extreme QoS to achieve the expected control performance. Therefore, we propose communication-control co-design and dynamic QoS to address the capacity issue for robotic manipulation use-cases in 5G-based industrial IoT. We have developed an advanced co-simulation framework that includes a network simulator, physics simulator, and compute emulator, for realistic performance evaluation of the proposed methods. Through simulations, we show significant improvements in network capacity (i.e., the number of supported URLLC devices), about 2x gain for the robotic manipulation use-case.

Keywords—dynamic QoS, 5G URLLC, industrial robotics, 5G network simulator, robotic manipulation

I. INTRODUCTION

In recent years, the Industrial IoT sector has been transforming towards fully connected, flexible and intelligent autonomous systems. The Fourth Industrial Revolution or Industry 4.0 vision is driving the transformation in manufacturing towards flexible and re-configurable systems that can adapt to the production demand in real-time. Industrial IoT use cases have stringent requirements such as ultra-low latency, and extremely high reliability [1]. The 3GPP 5G specifications (Releases 15-17) [2-4] have adopted ultra-reliable and low latency communication (URLLC) technologies. Industrial automation is one of the key applications for URLLC features in 5G. Certain industrial processes have extremely tight performance requirements for ultra-low latency and high reliability communications across control loops connecting sensors, actuators and controllers [1][3][4].

The 3GPP Release 15 [2] specification can support 99.999% reliability with 32 kB packets while Release 16 [3] specification can support 99.9999% reliability with 64 kB packets. The adopted flexible numerology for 5G frame structure can support sub-ms latencies. These features are attractive for supporting industrial IoT use cases over 5G networks. However, supporting such extreme performance metrics requires a lot of spectrum resources.

In a white paper by 5G Americas [1], it has been shown that the capacity of 5G system supporting URLLC features is greatly impacted with the target requirements, i.e., as latency requirements are tightened, the URLLC capacity quickly drops to low values. High reliability requirements also reduce capacity as it comes with the cost of lower spectrum efficiency (due to lower order modulation and coding schemes). A 3GPP 5G Technical Requirement [5] also shows that the capacity is very low when supporting stringent requirements, e.g., a maximum of 25 UEs (with periodic traffic, 32 bytes every 2ms) can be supported per cell with ~98% satisfaction rate for latency<1ms and reliability>99.9999% within 20MHz bandwidth.

Industrial networks need to support a large number of devices such as robotic arms, autonomous mobile robots (AMRs) and sensors including multiple cameras on the factory floor. Supporting URLLC connectivity for a large number of sensors and actuators needs large amounts of spectrum resources. An alternative approach is to develop intelligent techniques to reduce the overall amount of resources needed without compromising the target application goals. Towards the latter approach, a communication-control co-design method is proposed in [6], whereby both control and communication domains adapt to each other. It proposes the concept of application availability and application reliability as the quality of control requirements of the application based on an example of automated guided vehicle (AGV). An application-adaptive resource management is then proposed to adjust the success probability for the next transmission by dynamically adding or removing one or multiple links based on these application dependability metrics. Therein, the solution assumes availability of multiple independent links or multi-connectivity.

Another communication-control co-design scheme to reduce wireless resource consumption based on the state of the control system has been proposed in [7], whereby a real-time wireless control process is divided into two phases to adopt extremely high communication quality of service (QoS) for one phase and lower communication QoS for another phase. Simulation results in [7] show a reduction in communication energy consumption while maintaining good control performance even for 2 levels of dynamic QoS adjustment. Control-communication co-design has been studied primarily in the model-based context for simple control and communication systems that do not well model complex industrial use cases [7-10]. These methods have been further extended to Wi-Fi and 5G systems [11][12]. A control-communication co-design was introduced for industrial environment in [13] that leverages data driven reinforcement learning methods to handle the complex models present in industrial systems.

In this paper, we propose dynamic QoS to enable communication-control co-design that leverages dynamic control state information to address the capacity issue in 5G-based industrial IoT use-cases. The dynamic QoS provides a way to adapt the network resource utilization to the dynamic variations in application QoS requirements that is determined by the communication-control co-design algorithm. Another novel contribution of this paper is the use of an advanced co-simulation framework to model an accurate and scalable factory environment by considering physics, network, and compute models. Through simulations, we show significant improvements in network capacity (i.e., the number of supported URLLC devices), about 2x gain for the robotic manipulation use-case. This paper is organized as follows: in Section II, we introduce the concept of communication-control co-design. In Section III, we provide the details of our co-simulation framework and the simulation scenario. We present the simulation results and analysis in Section IV, followed by some key discussions and future research directions in Section V. Finally, we conclude the paper in Section VI.

II. Communication-control Co-design

The principle of communication-control co-design is based on the use of dynamic control state information to inform the communication system and, vice versa, to use dynamic communication state information to inform the control system. A variety of model-based [7-10] and data driven [13][14] techniques have been used to co-design control and communication systems, with varying degrees of complexity and performance.

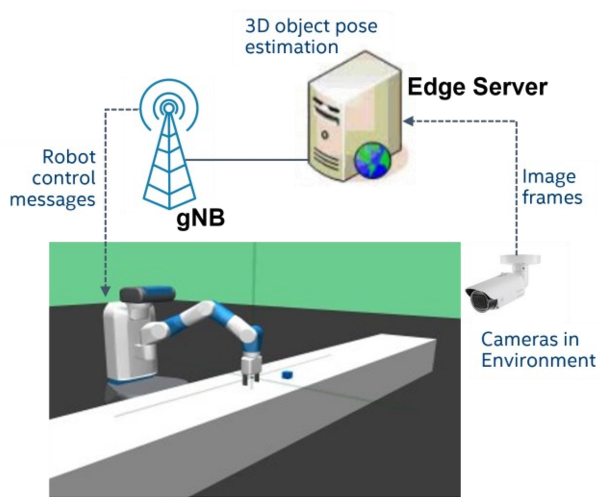

Fig. 1. Industrial robotic manipulation use-case.

To understand the co-design procedure, consider an example industrial use case of a robotic manipulator picking objects from a moving conveyor belt, as illustrated in Fig. 1. The robotic manipulation system features a robotic arm agent, an Edge compute system, and a set of cameras placed around the conveyor belt. The robot arm and the cameras are connected over a wireless 5G network, with the gNodeB (gNB) co-located with the Edge compute system.

A robotic arm needs to know real-time location information of its nearby objects on the conveyor belt to successfully complete the manipulation tasks. However, the robotic arms are assumed to have no sophisticated sensing capabilities, and hence rely on Edge computing for this critical information. The state information of the object, namely its position and velocity, is obtained via the processing of camera data at the Edge. In particular, camera frames containing image and depth data are sent over the 5G network to the Edge compute system, which runs object pose estimation workloads over the set of camera frames to obtain object state, which is sent to the robotic arm to be used in the computation of its local control action. Thus, the closed loop control system is obtained over the 5G network, with an uplink between camera and Edge, and downlink between Edge and robot.

Co-design in this use-case is critical in improving the efficiency of the 5G network. The 5G network should adapt to dynamics of the robotic system in a manner that improves efficiency. Using control theory, we can identify performance requirements for the system, such as minimum object lifting percentage, and dynamically adapt the QoS to meet such requirements at minimum cost. The QoS parameters at discrete time index r are given by packet delivery probability PDP_r and latency L_r. These dynamic parameters are computed at the Edge as a function of the current state of the object and the robotic system, and communicated to the access network to be used in network scheduling of packets associated to various control loops. In particular, with a current application state x_r, the minimum QoS parameters can be determined as,

$$PDP_r, L_r = \arg\min_{PDP,L}[Cost(PDP, L) - \lambda\, Perf(x_r, PDP, L)] \quad (1)$$

where $Cost(.)$ and $Perf(.)$ characterize the cost of achieving given QoS parameters and the resulting application performance, respectively. Moreover, the hyperparameter λ tunes the tradeoff between lower QoS and the effect on application-level performance. For the Edge node to be able to determine packet delivery probability and latency via (1), it needs access to the application state x_r, which includes the current object and robot states. Note in Figure 1 that the former is already computed at the Edge (see Figure 1), while the latter is known to the robot but can be periodically communicated to the Edge node with small communication overhead. This dynamic computation of QoS parameters contrasts with conventional URLLC design, in which high PDP_r and low L_r is expected to be maintained at all times to protect against worst-case scenarios. In the co-designed system, these parameters can be loosened during non-critical periods of the task, e.g., when the object is far from the robotic arm or when location prediction accuracy is high.

As previously mentioned, model-based methods for computing the control state-aware dynamic QoS parameters in (1) are often difficult to implement in robotic systems, which feature complex dynamics that may not be well modelled. For industrial systems, we leverage the use of physics and a robotic application simulator, MuJoCo [15] to find data driven solutions. The co-design policy is represented with deep neural networks (DNNs), whose parameters are optimized for performance by simulating a large number of object manipulation tasks and optimizing via deep reinforcement learning methods—see [13] for details on this approach.

III. Simulation Methodology

This section presents the details of an advanced co-simulation platform that models a complete end to end industrial system and is used for the performance evaluation of co-design and dynamic QoS techniques. The co-simulation platform implements an end-to-end simulation methodology which includes not only a 5G network simulator, but also includes a physics simulator, real application workloads, and compute emulator. Having an end-to-end simulation platform is important to evaluate the benefits of our co-design algorithm and quantify the impact on the overall system and application performance. As co-design relies on the dynamic QoS to achieve network capacity gain, it should be demonstrated that the performance of the use-case is not negatively impacted by the co-design algorithm. Therefore, a physics simulator is used to realistically model the details of use-case itself (robotic manipulation is used as an example use-case in this paper) by considering the laws of physics. Also, real application workloads required for the use-case are used in the simulator to achieve the most realistic representation of the use-case. For example, artificial intelligence (AI) based implementation of object detection and pose estimation algorithms are used at the Edge module, actual images of a virtual world in the physics simulator are rendered by the camera modules, etc. Since, the compute nodes in an industrial scenario (Edge, cameras, etc.) are characterized with diverse computing resources and capabilities, a compute emulator is used to accurately model distributed computing on different nodes in the simulation. The compute emulator orchestrates various containers to run the application workloads and also controls the execution of each container appropriately to match the compute capability of its respective compute node.

To summarize, the simulation method makes use of a 5G network simulator, physics simulator, real workloads, and compute emulator, all working synchronously for an accurate, end-to-end performance evaluation of the use-case. Since the co-simulation platform needs to run several detailed models and application workloads, it may be time-consuming to run large-scale scenarios with large numbers of devices and sensors. We will discuss some technics to overcome these challenges in Section III.A. In the following paragraphs, we discuss details of the co-simulator architecture.

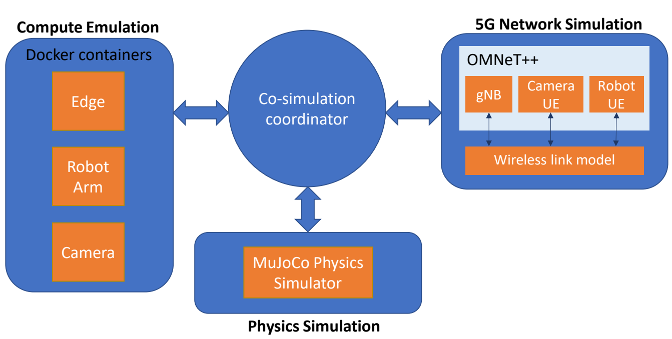

Fig. 2. Overview of co-simulation platform.

A high-level overview of the co-simulation platform is shown in Fig. 2, which mainly comprises of physics, network, and compute modules. These modules are configured and synchronized by the co-simulation coordinator. MuJoCo [15] is the physics simulation module used to model a 3D environment of the factory use-case which offers accurate and efficient physics simulation with agents such as industrial robotic arms, conveyor belt, camera sensors, and so on. The computation associated with the agents (Edge, robot arms, and cameras) have been emulated using Docker containers. The containers run real application workloads, for example, image compression algorithms in a camera container, object detection and pose estimation algorithms in an Edge container, and movement control algorithms in a robot arm container. During a simulation run, the status of the agents in the physics simulator are shared with the compute containers appropriately via our developed inter-process communication (IPC) interfaces. The shared status information includes rendered camera images and position coordinates of robot arms, etc.

The communication between the compute nodes over the network is modeled using the 5G network simulator which is developed in Omnet++ [16]. The 5G network simulator models the packet loss rates and packet latencies of the IP packets sent from the compute nodes. The protocol stack models of gNB and UE nodes in Omnet++ mainly contain medium access control (MAC) and physical (PHY) layers which are modeled according to the concepts in 3GPP specifications.
The MAC layer functionalities modeled in the simulator include scheduling tasks such as UE selection, resource allocation, and MCS selection. In the PHY layer, the 5G frame structure supports various slot formats to support diverse use-cases. In the simulator, time division duplex (TDD) with self-contained slot structure (supports low-latency transmissions) is modeled which includes a control channel, data channel, and an acknowledgement channel within the same slot.

The wireless transmission and reception of PHY layer packets are modeled using *wireless link model* as shown in Fig. 3. To simulate over-the-air transmissions of PHY layer packets, the wireless link model uses 3GPP specified Indoor Factory (InF) [17]. The implemented channel model includes large-scale fading (pathloss and shadowing) and small-scale fading models. For every receiver node, the wireless link model uses 3GPP defined procedures to compute different parameters such as received signal power, interference powers, SINR, packet error rate, etc., and then decides whether a PHY packet transmission was successfully received or not at the receiver node.

A. *Simulation Scenario*

This section describes an industrial robotics scenario utilized for the performance evaluation of communications-control co-design. A large-scale indoor factory scenario is simulated in which several robotic arms are operating concurrently, each at a conveyor belt, as illustrated in Fig. 3. The objects on a conveyor belt sequentially arrive towards a robotic arm at a fixed velocity. The task of a robotic arm is to pick the moving objects from its respective conveyor belt and place it on an adjacent stationary platform. Each robotic arm has an associated depth camera to perceive the environment around its vicinity. The camera and robot nodes are assumed to have low compute capabilities, and hence cannot execute complex environment perception algorithms in real time to determine the object locations on the conveyor belt to complete the pick-and-place tasks.

The computationally expensive environment perception tasks are offloaded to the Edge via the 5G network. The cameras continuously stream compressed image and depth frames in the uplink direction to the Edge. The Edge application performs object detection and pose estimation tasks, and sends the objects' pose information in the downlink direction to the respective robotic arms. Several control loops can be included in the simulation scenario, each comprised of a camera sensor, Edge process, and a robot arm.

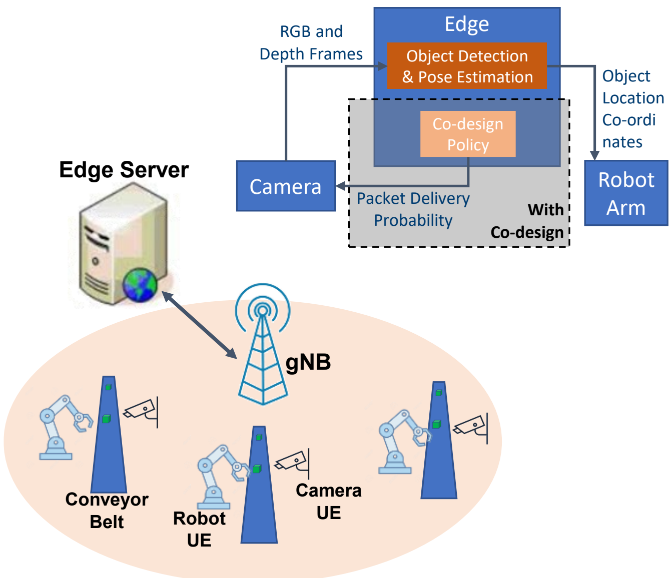

Fig. 3. Simulation scenario of industrial robotics use-case.

For the case with control-communications co-design enabled, a co-design policy is executed for each control loop at the Edge as illustrated in Fig. 3. Using the perception information, the policy periodically generates the required packet delivery probability (PDP_r) parameter and sends it to the camera over the downlink. The cameras receive their respective PDP_r parameter and use it to probabilistically drop the image and depth frames, thereby reducing the amount of traffic injected into the network. Here, the PDP_r is a novel dynamic QoS control signaling that we propose in this paper. In the current work, the periodic signaling of the dynamic QoS parameter was performed in the application layer. However, it could be done more efficiently via the lower layers in 5G protocol stack. We will discuss more about it in Section V.

To evaluate performance benefits of the co-design algorithm on the 5G network and the overall robotic control system, the following key performance indicators (KPIs) are defined:

1) Task Success Rate: A task is considered successful if a robotic arm successfully picks a moving object from the conveyor belt and places it on its adjacent stationary platform correctly. The task success rate is defined as the ratio of the number of successful pick-and-place tasks to the total number of tasks presented to a robotic arm.

2) Network Capacity: In the context of this robotic tasks use-case, the network capacity is defined as the maximum number of concurrent control loops that can supported by the network such that the task success rate is maintained at or above a specified value.

For performance evaluation of the scenario illustrated in Fig. 3, it would be necessary to simulate many control loops (in the order of several 10's of control loops in a cell). Simulation of each control loop involves several computationally expensive tasks such as computation of accurate physics of robotic arm and objects on the conveyor belt, rendering of realistic camera images and depth frames, object detection and pose estimation algorithms, etc. Also, to obtain statistically reliable results for task success rate, each simulation case should be run to cover a sufficient number of tasks (say 100 tasks) which would require several hundreds of simulated seconds (time duration in the simulated world). Simulating a large-scale scenario with these requirements is a compute-heavy task and needs to be simplified. To overcome these challenges, we designated a particular control loop in the scenario as a primary control loop and simulated all the details of its physics, network, and compute models. For the other control loops in the scenario, compute intensive tasks such as object pose estimation, rendering of camera frames, etc. were removed, and the object pose information was directly obtained from the physics simulator.

IV. Simulation Results

This section presents simulation results and evaluates the performance gains of control-communications co-design in 5G Industrial IoT use-case by analyzing the task success rate and network capacity KPIs. The simulation scenario described in Section III.A was run for the two cases: without co-design, and with co-design. The key parameter settings for the 5G network simulator are as shown in Table I.

TABLE I. 5G Simulator Parameter Settings

Parameter	Value
Carrier frequency	3.5 GHz (CBRS band)
Bandwidth, subcarrier spacing	100 MHz, 15KHz
BS and UE transmit power, and antenna configuration	23 dBm, 1x1 SISO
Channel model	3GPP InF-SL (sparse clutter and low BS height)
Channel coding	Polar coding for Control and Ack channels, and LDPC for Data channel

The task success rate versus number of control loops is plotted in Fig. 4 for the cases of with and without co-design. Here, each control loop represents a conveyor belt with a robotic arm, depth camera and Edge communicating mission critical information via the 5G network. It can be noted that with a smaller number of control loops in the system, 100% task success rate is achieved for both of the cases. This shows that the co-design algorithm intelligently drops less critical packets such that it does not impact the task success rate. Furthermore, the results in Fig. 4 show significant gain in network capacity with co-design. For example, assuming 100% task success rate requirement, the network can support up to 16 control loops without codesign. Whereas the network can support up to 31 control loops with co-design, a ~2x gain in network capacity. Similarly, at 98% task success rate requirement, the network can support 17 and 32 control loops without and with co-design, respectively, a 2x gain in network capacity.

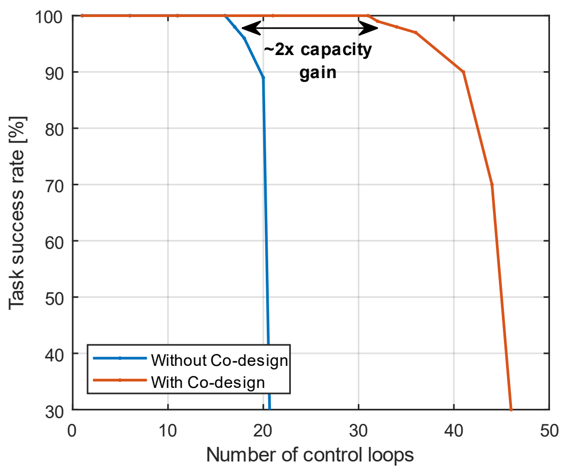

Fig. 4. Task success rate with and without co-design for different number of control loops.

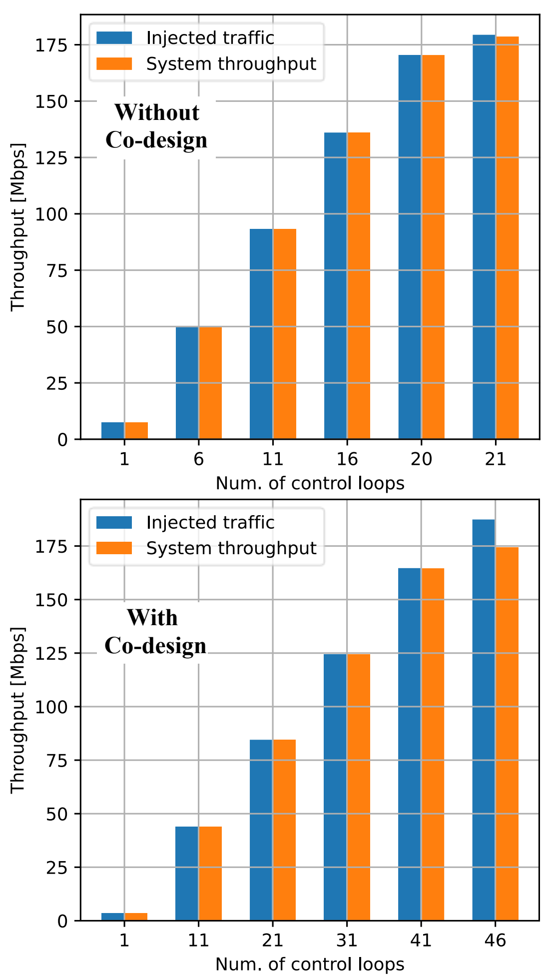

Fig. 5. Injected traffic and system throughput for different number of control loops.

To obtain further insights, consider the injected traffic and system throughput results shown in Fig. 5. As the number of control loops increases in the system, the injected traffic into the network also increases proportionally. For the case without co-design, the injected traffic with 21 control loops (~179.4Mbps) exceeds the system capacity (~178.6Mbps). Hence, the task success rate reduces significantly due to congestion. On the other hand, with co-design, the injected traffic from 21 control loops is only about 84.5Mbps (less than half of system capacity) which can be easily supported by the network.

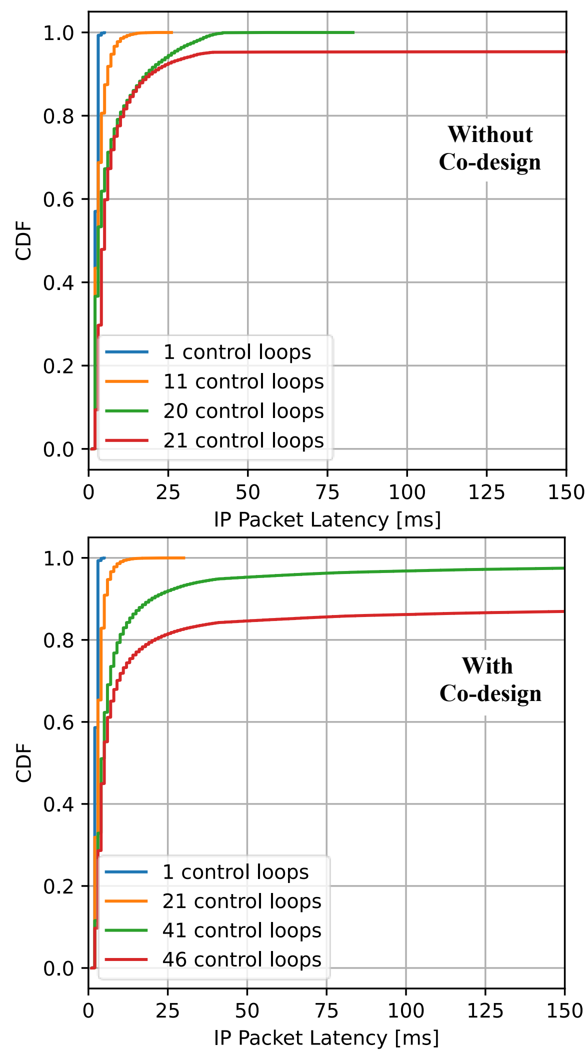

Fig. 6. IP packet latency for different number of control loops.

The cumulative distribution function (CDF) of latency of IP packets within the 5G network is shown in Fig. 6. As expected, the latency increases with the increasing number of control loops. For the case without co-design, the 99-percentile latency is less than 40ms with up to 20 control loops. However, with 21 or more control loops, the latency increases significantly due to the network congestion. On the other hand, with co-design, the packet latency is comparatively smaller for the case of 21 control loops.

V. Disucssion and Future Directions

The simulation results presented in Section IV show significant capacity improvements using the communication-control co-design, with which a greater number of robots can be supported in a given wireless network. In particular, about 2x gain in capacity was observed for the manipulation use-case. This is the result of the co-design algorithm and dynamic QoS technique with which the non-essential camera frames were intelligently dropped by considering the state of control system, resulting in lesser traffic injected into the network per control loop. On the contrary, without co-design and dynamic QoS, the QoS flows were required to be configured to satisfy stringent reliability and latency requirements for all the packets, which eventually resulted in capacity degradation.

In this paper, a robotic manipulation scenario was studied which is characterized by downlink control traffic with small application packet sizes, and the uplink traffic with large application packet sizes from cameras and other sensors. Such a scenario presents greater challenges in network capacity due to the stringent reliability and latency requirements. In that case, dropping uplink packets which are bigger in size would provide higher margin in terms of reduction in traffic injected into the network. However, uplink packet dropping involves additional requirements like signaling of dynamic QoS parameters to the UE, which was performed in application layer in the current work. However, signaling of the dynamic QoS parameters could be performed more efficiently and with low latency via the lower layers in 5G protocol stack. Exploration of those efficient signaling methods is left for the future work.

Another direction to achieve further improvements in network capacity is by using co-design policy for improving the robustness of robotic systems to the delay and packet loss introduced by the network. Here, the co-design policy is based on delay-aware “state correction” [18], in which the robot uses time stamp information to determine time delay of the received downlink state information. The state correction co-design policy can run in complementary to the communications-control co-design policy to provide significant network capacity improvements.

VI. Conclusion

In this paper, we discussed the capacity challenges of URLLC in large-scale industrial use-case scenarios. By taking robotic manipulation as an example use-case, we designed communications-control co-design algorithm and dynamic QoS techniques to address the capacity issue. We have developed an advanced co-simulation framework that includes a 5G network simulator, physics simulator, and emulation of compute workloads, for comprehensive and accurate modeling of the manipulation use-case. The simulation results show that co-design and dynamic QoS can provide significant improvements in network capacity (i.e. the number of supported control loops), about 2x gain for the manipulation use-case.

References

[1] 5G Americas, “New services & applications with 5G ultra-reliable low latency communications,” White Paper, Nov. 2018.

[2] 3GPP TR 21.915 V15.0.0 (2019-09), 3GPP; Technical Specification Group Services and System Aspects; Release 15 Description; Summary of Rel-15 Work Items (Release 15)

[3] 3GPP TR 21.916 V16.2.0 (2022-06); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Release 16 Description; Summary of Rel-16 Work Items (Release 16)

[4] TR 21.917 V0.7.0 (2022-06), 3GPP; Technical Specification Group Services and System Aspects; Release 17 Description; Summary of Rel-17 Work Items (Release 17)

[5] 3GPP TR 38.824 V16.0.0 (2019-03), “Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC),” Release16.

[6] L. Scheuvens, T. Hößler, A. N. Barreto and G. P. Fettweis, "Wireless Control Communications Co-Design via Application-Adaptive Resource Management," 2019 IEEE 2nd 5G World Forum (5GWF), 2019, pp. 298-303, doi: 10.1109/5GWF.2019.8911713.

[7] B. Chang, G. Zhao, L. Zhang, M. A. Imran, Z. Chen and L. Li, "Dynamic Communication QoS Design for Real-Time Wireless Control Systems," in IEEE Sensors Journal, vol. 20, no. 6, pp. 3005-3015, 15 March15, 2020, doi: 10.1109/JSEN.2019.2957569.

[8] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 2009, pp. 5832-5838, doi: 10.1109/CDC.2009.5400528.

[9] K. Gatsis, A. Ribeiro and G. J. Pappas, "Optimal Power Management in Wireless Control Systems," in IEEE Transactions on Automatic Control, vol. 59, no. 6, pp. 1495-1510, June 2014, doi: 10.1109/TAC.2014.2305951.

[10] Z. Hou, C. She, Y. Li, L. Zhuo and B. Vucetic, "Prediction and Communication Co-Design for Ultra-Reliable and Low-Latency Communications," in IEEE Transactions on Wireless Communications, vol. 19, no. 2, pp. 1196-1209, Feb. 2020, doi: 10.1109/TWC.2019.2951660.

[11] Mark Eisen, et al, “Control Aware Radio Resource Allocation in Low Latency Wireless Control Systems,” IEEE Internet of Things Journal, Vol. 6, No. 5, Oct. 2019.

[12] Mark Eisen, et al, “Scheduling Low Latency Traffic for Wireless Control Systems in 5G Networks,” IEEE ICC 2020.

[13] M. Eisen, S. Shukla, D. Cavalcanti and A. S. Baxi, "Communication-Control Co-design in Wireless Edge Industrial Systems," 2022 IEEE 18th International Conference on Factory Communication Systems (WFCS), 2022, pp. 1-8, doi: 10.1109/WFCS53837.2022.9779178.

[14] D. Baumann, J. -J. Zhu, G. Martius and S. Trimpe, "Deep Reinforcement Learning for Event-Triggered Control," 2018 IEEE Conference on Decision and Control (CDC), 2018, pp. 943-950, doi: 10.1109/CDC.2018.8619335.

[15] “Mujoco,” *MuJoCo*. [Online]. Available: https://mujoco.org/. [Accessed: 12-Jul-2022].

[16] “OMNeT++ Discrete Event Simulator,” *OMNeT++*. [Online]. Available: https://omnetpp.org/. [Accessed: 12-Jul-2022].

[17] 3GPP TR 38.901 V17.0.0 (2022-03), “Study on channel model for frequencies from 0.5 to 100 GHz,” Release17.

[18] A. Baxi et al., "Towards factory-scale Edge robotic systems: challenges and research directions," *in IEEE Internet of Things Magazine*, vol. 5, no. 3, pp. 26-31, Sep. 2022.