Accepted for presentation in: IEEE INFOCOM WKSHPS: The 11th International Workshop on Computer and Networking Experimental Research using Testbeds (INFOCOM CNERT 2024)

Data-Driven Proactive Uplink Slicing enabling Real-Time Control within an Open RAN Testbed

Dennis Overbeck[1], Niklas A. Wagner[1], Robin Wiebusch[1], Jens Püttschneider[2], Timm Faulwasser[2] and Christian Wietfeld[1]
[1]Communication Networks Institute (CNI)
[2]Institute of Energy Systems, Energy Efficiency and Energy Economics (IE3)
TU Dortmund University, 44227 Dortmund, Germany
Email: {dennis.overbeck, niklas.wagner, robin.wiebusch, jens.puettschneider, christian.wietfeld}@tu-dortmund.de, timm.faulwasser@ieee.org

Abstract—**In terms of flexibility and scalability, network slicing enables the realization of contradicting services on public mobile radio infrastructure – providing robustness and reliability for critical services. Combined with the new paradigm of Open RAN, network operators can utilize Commercial of-the-Shelf (COTS) hardware to build a fully softwarized radio network with sophisticated resource management. Using xApps built on top of a near Real-Time RAN Intelligent Controller (RIC) operating as a virtual entry point, the allocation of resources can be optimized to provide low latency and reliability to critical services while yielding high spectral efficiency. In this paper, we showcase the capability of our developed SAMUS xApp relying on Machine Learning (ML) methods in a real-world experimental setup assessed by an inverted pendulum operated with Model Predictive Control (MPC). Compared to traditional scheduling, we reduce the mean uplink one-way delay by 65% using our SAMUS xApp, while simultaneously increasing spectral efficiency by 61% compared to static proactive allocation, maintaining similar control performance.**

I. Introduction

Future energy networks require robust communication infrastructures, however, dedicated infrastructure tends to be cost-intensive. With the current fifth generation of mobile radio networks (5G), the key technology of network slicing enables reliable data transmission over existing public communication infrastructure. Here, dedicated resources can be allocated for critical services, e.g., smart grid management and microgrids. In contrast to the conventional best-effort-based scheduling, these resources are only available to the devices within a specific network slice, further enhancing previous Quality of Service (QoS)-based approaches and enable virtually isolated communication services. The 3rd Generation Partnership Project (3GPP) defines two procedures to allow the allocation of resources without the need for scheduling requests, drastically decreasing the experienced latency for Ultra-Reliable Low Latency Communications (URLLC) devices. The first one is to allocate resources using configured grants that are communicated to the User Equipment (UE) via Radio Resource Control (RRC) signaling and focus on strictly recurrent transmissions over longer time periods. However, the second approach, proactive grants, enables more flexibility by allocating resources proactively, i.e., without the need to send a scheduling request. Using Downlink Control Information (DCI), the base station can also announce specific reserved resources towards the UE without the device initially requesting them. Thus, this approach opens the possibility of improving the scheduling by means of machine learning or heuristics to optimize both latency and overall network performance by predicting the resource demand for low-latency communications as well as other critical services. In our previous work [1], we examined the impact of the Open RAN concept's introduced additional interfaces in terms of latency, which showed to be relatively small compared to the total latency. This work extends the framework by considering a networked MPC as a latency-critical application that relies on robust communication for optimal performance. Co-designing the communication system with the control system is another aspect where channel disturbance plays a vital role. As delays induced by wireless communication have a significant impact on the performance of the control algorithm, the underlying communication system needs to address the expected latency beforehand. Within this work, we present a functional experimental testbed, providing latency guarantees for mission-critical control via a prioritized network slice using proactive scheduling based on Open RAN specifications for an inverted pendulum control benchmark. Furthermore, we examine the trade-off between spectral efficiency and control quality of an MPC controller evaluating

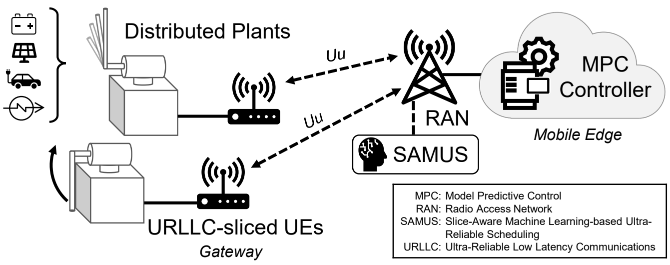

Fig. 1: Overview of the demonstrator system for reliable, networked Model Predictive Control (MPC) applications using an inverted pendulum as control benchmark. The control data exchange is performed via a fully softwarized, open-source wireless communication link with Open Radio Access Network (RAN) backend.

different resource allocation schemes. The sections of this work are structured as follows: We provide a brief overview of related work in Sec. II, then describe the key enablers for our demo testbed in Sec. III, providing an overview of the technologies employed. Subsequently, Sec. IV describes the testbed leveraged for the use case and outlines the key results. Finally, Sec. V summarizes our findings.

II. Related Work

The literature relevant to this paper, sharing common ground with our research, focuses mainly on the preliminary work on Machine Learning (ML)-based network slicing, and thus, resource allocation as well as its application in the form of xApps in the Open RAN context. Furthermore, works on the use-case-related MPC and the idea of communication control co-design are considered. Within the survey of [2], the remaining challenges and current state of ML-based resource management for network slicing within 5G and beyond networks are reviewed, highlighting their importance for future networks. Therefore, several works consider ML as a key enabler for efficient resource orchestration in the wireless domain, as well as the capabilities of 5G to reduce latency. Fast uplink grants for Massive Machine Type Communication (mMTC) environments enable to significantly decrease the impact of scheduling requests on the network [3]. The work shows how the combination of ML and proactive resource allocation can lead to decreased experienced latency in scheduling for the devices within the network. The authors of [4] propose an orchestration of resources within a sliced network by multiplexing URLLC and Enhanced Mobile Broadband (eMBB) services in an optimization problem, achieving lower packet blocking probability for bursty URLLC traffic. The achieved results were obtained within a simulation. Focusing on the optimization of resource allocation for URLLC services in coexistence to eMBB transmissions, the authors of [5] propose an optimization-aided Deep Reinforcement Learning (DRL) based framework. Here, mini-slots (or short Transmission Time Intervals (TTIs)) are used in combination with preemptive scheduling within a simulation-based evaluation to show promising results in terms of fulfilling URLLC requirements as well as reliability for eMBB services. Age of Information (AoI) for control algorithms is of crucial importance for networked control algorithms, and thus, the co-design of communication of control is indispensable for high control performance. The work of [6] presents a framework for determining the quality of control based on a given application. Specifically, the packet error rate is highlighted to impact the applications, and thus, adaptive resource allocation is proposed as key to preventing consecutive packet losses. The drive towards open interfaces pioneered by the Open RAN concept is explored in several works, including the work of [7], which showcases an xApp implementing heuristic control for lowering the experienced latency for Virtual Reality (VR) streams. The heuristic is based on the detected frames per second transmitted by the UE and adjusts the allocation of resource block groups accordingly. Furthermore, the authors of [8] propose an orchestration framework based on O-RAN to choose and assign specific trained ML models to different network slices depending on the incentivized Key Performance Indicators (KPIs). In our work, we show a tangible demo use case that demonstrates the importance of low latency for critical control services, while considering MPC including delay compensation, further increasing control quality.

III. ML-driven Proactive Network Slicing based on Open Interfaces

This section describes the utilized approach to proactive uplink resource management based on ML to provide low latency for the critical communications.

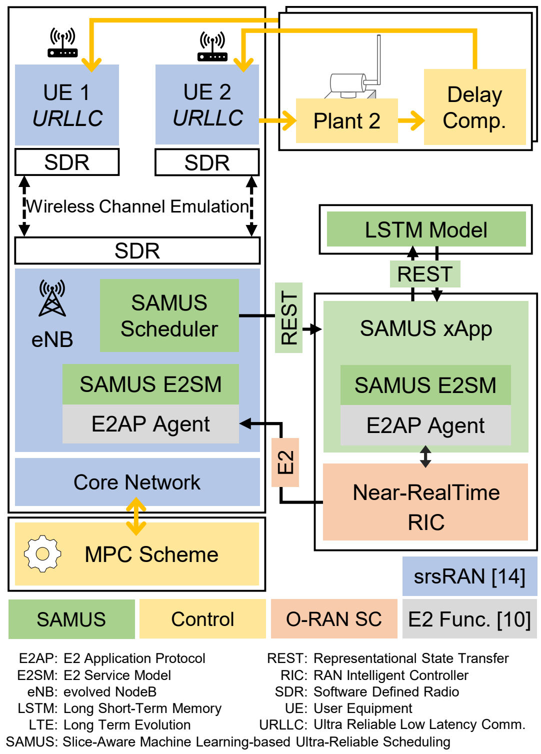

Fig. 2: Depiction of the underlying framework and communication streams for data-driven proactive scheduling by our SAMUS xApp.

A. *Proactive Resource Management with OpenRAN*

In our previous work [1], we presented our Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS) [9] approach within an open framework for predictive resource allocation. Embedded in the Open RAN architecture, and the E2 agent implementation provided by [10], we enabled the near-Real-Time (RT) RAN Intelligent Controller (RIC) direct access to the distribution of resources via a custom scheduler, providing an entry point for sophisticated scheduling operations using ML methods. The functionalities built on top of the RIC are called xApps, taking advantage of the open interfaces provided by the O-RAN standard. The key elements of this innovation relevant for this work are depicted in Fig. 2. Here, the standardized E2 interface is utilized to provide predicted MAC payload sizes to the scheduler every 20 ms.

The transmission in the opposite direction is realized using a Representational State Transfer (REST) interface towards the prediction server, which is part of the developed xApp. The inverted pendulums are connected to the UEs incorporating the state measurements, which are then transmitted via air interface towards the base station, inheriting the MPC controller. Within the base station, the received transmissions are then sent via the REST interface to the pre-trained prediction model. The predicted TTIs are then communicated back to the scheduler via E2 Control Messages, which are then used to proactively allocate radio resources based on these predictions.

B. Real-Time Prediction of Payloads using Machine Learning

The standard approach for UEs to receive resource allocations within mobile radio networks relies on the concept of actively requesting them via scheduling requests. These inquiries are sent via the Physical Uplink Control Channel (PUCCH) within specified time slots, the Scheduling Request Occasions (SROs), which are communicated by the base station within the DCI packet for each specific UE uniquely to avoid collisions. When packets need to be transmitted, therefore, a wait time is induced based on the next available SRO, followed by the delay of the air interface for the Scheduling Request (SR) to be received by the base station as well as the wait times for the allocated resource for the actual transmission of the data.

The whole process results in a significant delay experienced by the UE, even with small SRO periodicity, and is therefore not suitable for highly critical, latency-sensitive services such as defined for URLLC with requirements below 5 ms. The current 5G standard defines novel methods of assigning resources without requesting them in the first place and, therefore, gives

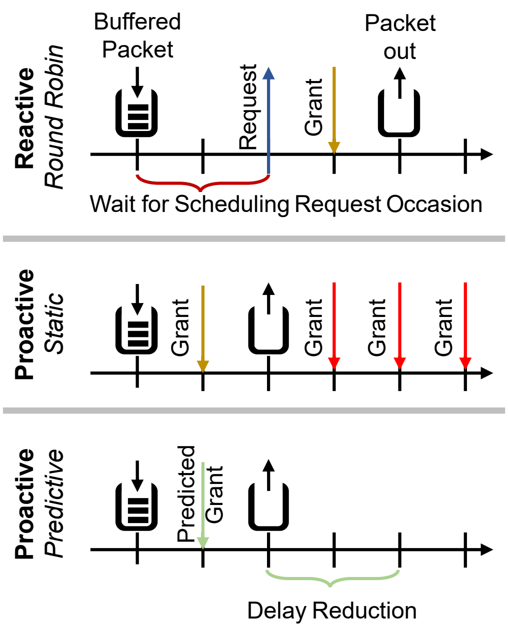

Fig. 3: Overview of the different possible scheduling schemes as standardized by the 3GPP from the UE's perspective.

TABLE I: SAMUS Training Parameter Settings

Learning Rate	10^{-3}
Layer Structure	LSTM (vanilla, 64 Units) Dense Layer (Activation: linear)
Batch Size	4
Epochs	72
Loss	MSE
Optimizer	ADAM

the opportunity to reduce the previously enumerated wait times drastically. Besides the fairly static Configured Grants (CGs), our work concentrates on the more dynamically available Proactive Grants (PGs), which can be used to proactively communicate available resource blocks to the UE via standard DCI packets. Here, the delay from requesting radio resources is virtually none; however, prediction methods, e.g., using ML, are needed to provide them in a spectrally efficient manner. In Fig. 3, the scheduling schemes under investigation are depicted. A static allocation of proactive grants, as shown in the center, fulfills the requirements of reducing latency to a minimum; however, at the cost of limiting available resources and thus bandwidth for other services within the network. Therefore, the idea to utilize ML methods, as shown on the left-hand side, aims to reduce the waste of resources by learning the pattern of unique UEs and assigning radio resources based on previous transmissions. For this, we employ a Long Short-Term Memory (LSTM) model trained offline for our use case of an MPC-operated pendulum.

The parameters used for training the model are listed in Tab. I; the framework used for training is Keras with TensorFlow backend. We use a pre-recorded packet trace of a single pendulum controller over a timespan of 875 s for training and validation of the model. The available dataset is split into $\frac{2}{3}$ training and $\frac{1}{3}$ validation subsets while test data is later generated live. Training of the model results in 60% exact matches of the payload sizes per timeslot in the validation subset. For the live prediction of estimated payloads, we employ a lookback size of 100 TTIs as well as a prediction horizon of 100 TTIs. The Medium Access Control (MAC) layer payload sizes per time slot serve as input for the LSTM model. The precise amount of assigned resources is critical since over-allocation results in decreased spectral efficiency. In contrast, insufficient assignment of resources leads to the necessity of reactively scheduling packets and, thus, increasing latency. Therefore, we rely on peak prediction, i.e., we apply a noise gate and fix the allocations to a defined maximum to provide enough resources. Several challenges arise regarding the real-time prediction on *millisecond* timescales. For the UE to utilize the assigned grants, it is crucial to have those grants available within the expected time slot, otherwise the device will request resources using SRs. As the predictions performed by the model are based on the arrival of payloads at the base station, this leads to unexpected additional delays and, therefore, a propagation of deviation between the predicted

TTI and the ground truth of TTI, where the packet should have been scheduled. To counter this problem, we introduce a synchronization phase to evaluate the ground truth timing and provide the model with the information to correct the predicted TTIs towards it. Additionally, we use a broadening of the predicted timestep to compensate for mispredictions. We define the periodicity of evaluation phases $\tau = 2\,\mathrm{s}$, the duration of the evaluation phases $\xi = 100\,\mathrm{ms}$, and the timely broadening of predictions as $\beta = 7\,\mathrm{ms}$ at a data inter-arrival time of $\sigma = 20\,\mathrm{ms}$. Employing the evaluation phase, in comparison to statically allocating resources, results in the spectral efficiency given by the equation

$$\#AllocatedPRBs[\%] = \frac{\xi}{\tau} + \left(1 - \frac{\xi}{\tau}\right)\frac{\beta}{\sigma}.$$

Therefore, the theoretical relative resource efficiency gain by employing the data-driven scheduling method is 66.75%.

C. Model Predictive Control

MPC is an optimization-based control method that calculates feedback actions based on the repeated solution of an open-loop Optimal Control Problem (OCP) [11]. At each sampling time, the state of the plant is measured, and an optimal open-loop prediction of states and inputs is calculated over a finite horizon based on the minimization of a cost functional while satisfying the system dynamics and constraints. Then, the first part of this trajectory is implemented until the next sampling time.

Here, we consider the control task of stabilizing an inverted Furuta-Pendulum in the upper equilibrium position. The system is a fourth-order system with state vector $x = (\theta, \alpha, \dot{\theta}, \dot{\alpha})^\top \in \mathbb{R}^4$, where θ is the rotary arm angle with the angular velocity $\dot{\theta}$ and α represents the pendulum arm angle with velocity $\dot{\alpha}$. The control input $u \in [-7.5, 7.5]$ is the voltage applied to the motor. For the system model, we refer to [12]. The continuous time dynamics $\dot{x} = f(x, u),\ x(0) = x_0$ are discretized using a fourth-order Runge-Kutta discretization scheme with a fixed stepsize of 20 ms which gives $x_{k+1} = f^{\mathrm{d}}(x_k, u_k),\ x_0 = x_0$.

Within the control loop, the plant measures its states $x(t_k)$ at time t_k and sends it to the controller in the uplink channel. There, the varying uplink delay $\tau_{\mathrm{UL}}(t_k)$ is determined. The combined calculation time and downlink delay τ_{DL} is assumed to be constant and measured before the experiment and measured in a preparatory step. With this assumption, the total delay $\tau(t_k) = \tau_{\mathrm{UL}}(t_k) + \tau_{\mathrm{DL}}$ is available at the controller and can be compensated by forward integration of the plant model

$$\hat{x}(t_k + \tau(t_k)) = x(t_k) + \int_{t_k}^{t_k + \tau(t_k)} f(x(t), u(t))\mathrm{d}t.$$

Then, the discrete-time OCP is solved for the delay-compensated state

$$\min_{u_i, x_i} \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + R u_k^2\right) + x_N^\top P x_N \tag{1a}$$

$$\text{subject to} \quad x_{k+1} = f^{\mathrm{d}}(x_k, u_k), \quad \forall k \in \{0, ..., N-1\} \tag{1b}$$

$$x_0 = \hat{x}(t_k + \tau(t_k)) \tag{1c}$$

$$-7.5 \leq u_k \leq 7.5 \quad \forall k \in \{0, ..., N-1\} \tag{1d}$$

$$-1\pi/2 \leq \theta_k \leq 1\pi/2 \quad \forall k \in \{0, ..., N\}\,. \tag{1e}$$

The OCP hyperparameters are prediction horizon $N = 100$, state cost matrix $Q = \mathrm{diag}(5, 2, 0.1, 0.01)$, input penalty $R = 1$. The terminal penalty P is determined via the solution of the discrete algebraic Riccati equation for the linearized dynamics at the upper equilibrium point. The OCP is implemented in Acados [13]. After the OCP is solved, the controller sends the first element of the input trajectory to the plant, where it is applied when it has been received at time $t_k + \tau(t_k)$, i.e., $u(t_k + \tau(t_k)) = u_0$.

As a metric for comparing control performance over T timesteps, we employ the closed-loop cost

$$J_{\mathrm{CL}} = \sum_{k=0}^{T} x(t_k)^\top Q x(t_k) + R u(t_k)^2. \tag{2}$$

IV. Experimental Setup and Results

In the following section, we discuss the evaluation setup and results for the latency-critical control.

A. Evaluation Scenario and Experimental Setup

The scenario for the following experimental evaluation is constructed to assess the achievable control performance in an agile control task driven by an MPC controller at the mobile edge via our SAMUS-enabled base station. We consider Quanser Servo 2 inverted pendulums [12] as plants to be controlled independently. The regulation of the pendulums to their unstable equilibria is a common benchmark problem facing similar challenges as, e.g., applications in power systems. The computation platform for Evolved Packet Core (EPC), Evolved Node B (eNB) based on srsRAN 21.10 [14], near-RT RIC with xApp and the prediction model is a shared server (AMD Ryzen 5900X, 32 GB RAM, Ubuntu 20.04). On UE side, one compact platform is employed each (AMD Ryzen 7735U, 32 GB RAM, Ubuntu 22.04). The radio side is operated by USRP B210 Software-Defined Radios (SDRs). For reproducible Radio Frequency (RF) conditions, we emulate a realistic channel based on the 3GPP EPA model [15] using a Keysight Propsim F64 radio channel emulator. The eNB is set to a bandwidth of 3 MHz, 15 Physical Resource Blocks (PRBs) and operation is performed at the campus frequency of 3.75 GHz in Frequency Division Duplex (FDD) mode. To enable measurements of one-way delay metrics, all devices are synchronized with a local *chrony* instance using the Network Time Protocol (NTP). Packet transmission times from the pendulum and reception times at the controller are captured by *tshark* traces.

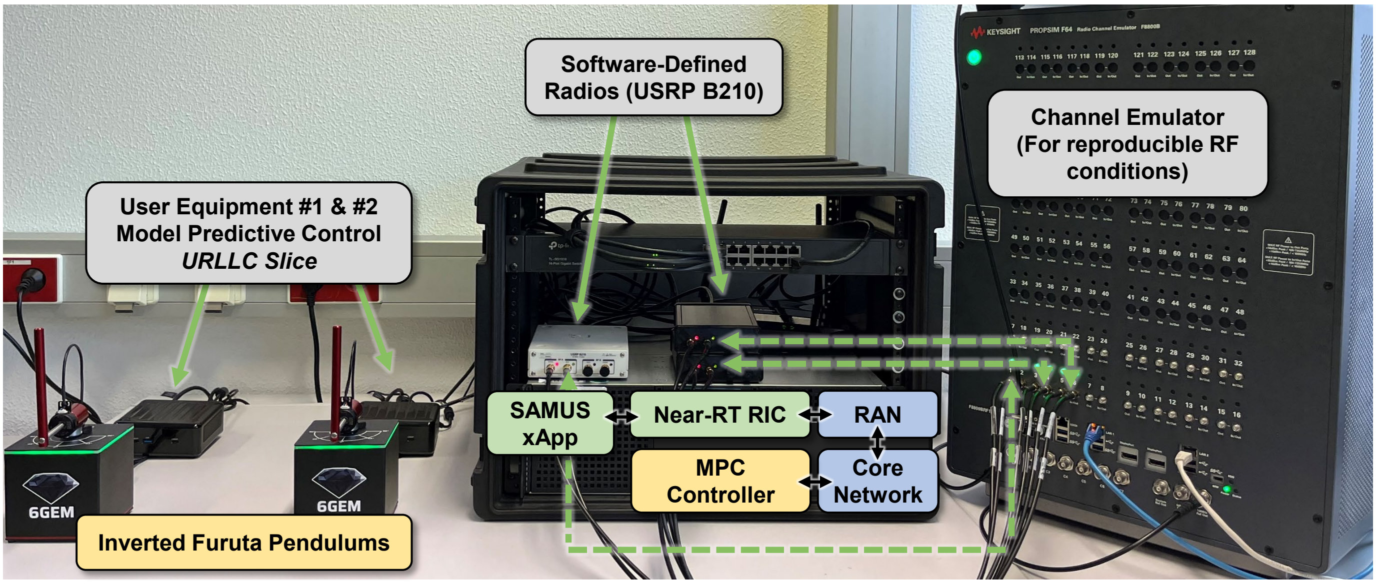

Fig. 4: Real-world experimental setup, comprised of two latency critical UEs serving one inverted pendulum each and the base station including near-RT RIC and SAMUS xApp.

B. Results of Application-based Evaluation

From the communication perspective, we compare the results of our employed scheduling algorithms in terms of experienced one-way delay for the connected devices as well as the spectral efficiency, i.e., the resources used by the UEs to achieve the latency. Furthermore, we discuss the impact of this induced delay in the communication link on the control performance of the MPC (1). We define one-way delay as the time a packet requires to be received on the Packet Data Network Gateway (P-GW) of the EPC after being sent out from the UE application. As a control baseline, we evaluate the closed-loop cost (2) of the MPC based on 25 test runs in a cabled ethernet setup. For the remaining measurements, the experiments were performed for an equal amount of $n = 25$ test runs, which lasted $t = 30\,\text{s}$ each, resulting in an average of approximately $35,000$ data points per pendulum per scheduling algorithm regarding latency and resource allocation. The Fig. 5 is structured to provide emphasis on the relations between delay (top), spectral efficiency (center), and closed-loop cost (bottom). On the left-hand side, the results for the traditional, reactive scheduling are depicted. Following a static allocation leveraging proactive grants to minimize latency in the center, our proposed modification using predictions is highlighted on the right-hand side of the figure. Starting at the top of Fig. 5, the measured one-way delay is depicted, split for the two deployed UEs, where both act as a communication gateway for the state transmission towards the controller residing at the core network. With the reactive scheduling approach, depicted on the left-hand side, we achieve a mean latency of 14.16 ms for both UEs with outliers reaching up to 69.59 ms for UE2. However, as shown in the center of the figure, the allocated resources remain low since they are only given to the UE when needed, resulting in virtually no waste of resources. Nevertheless, the closed-loop costs increase due to the high latency, as depicted at the bottom of the figure, indicating deteriorated control performance. Several retries in swinging up the pendulum induce high closed-loop cost according to (2). The baseline regarding the experienced delay via the demo testbed is given by statically allocating resources every TTI. By doing so, we achieve a mean delay of as low as 4.37 ms as depicted in the center row of Fig. 5. This behavior is mirrored by the closed-loop costs, as the state of the plant is transmitted without significant delays, enabling precise real-time control. However, since the resources are blocked every *ms*, the spectral efficiency decreases noticeably as they are uniquely assigned to the UEs. By employing our proposed SAMUS xApp (Fig. 5, right-hand side), the best of both worlds can be achieved with only minor outliers in latency due to prediction errors outside the broadening of the time step. The mean delay decreases to 4.9 ms, while the spectral efficiency increases by 60,70% compared to the proactive static allocation of resources. The performance achieved by the MPC for the SAMUS xApp is close to the level of static allocation.

V. Conclusion and Outlook

This paper demonstrated the control of a wirelessly networked pendulum with an MPC controller located at the network edge. Employing our SAMUS xApp, we were able to achieve 37% decreased closed-loop control cost compared to traditional scheduling by improving the resource allocation leveraging an LSTM model, predicting the precise timing of incoming packets. With focus on the achieved latency reduction, we were able to achieve a 65% decrease in mean uplink delay. While showing persistent control performance compared to the static proactive allocation of resources, our proposed improvements reach higher spectral efficiency, however, increase resource usage by 88% compared to traditional

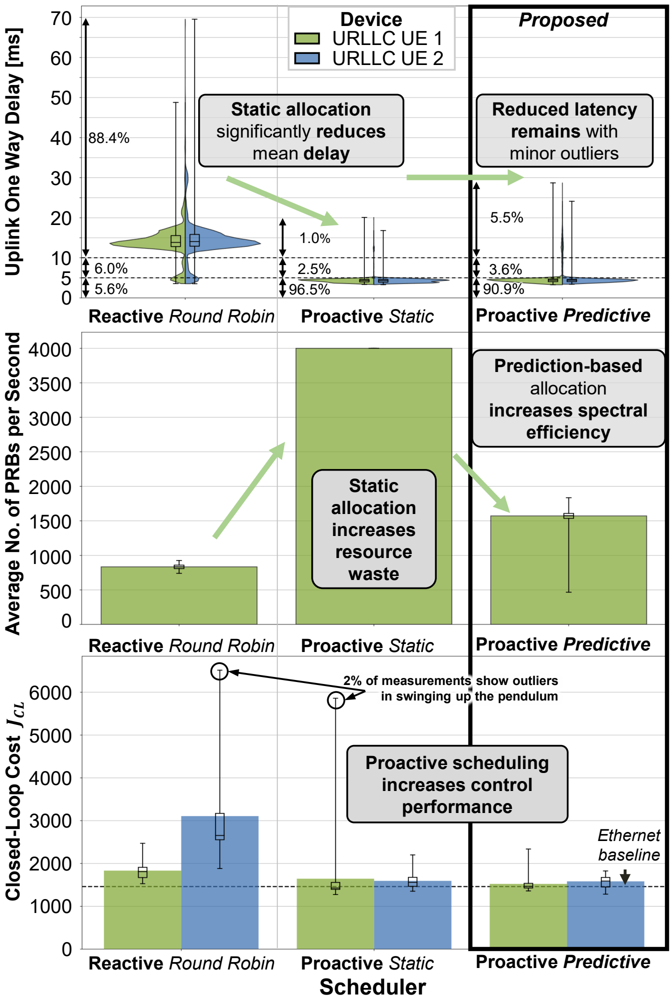

Fig. 5: Measured uplink one-way delay in relation to the resulting PRB allocation and the experienced closed-loop costs based on (2) for the discussed scheduling approaches.

reactive scheduling, emphasizing the trade-off needed to provide stable low latency. Moreover, first investigations show that reduced Signal-to-Noise Ratio (SNR) heavily impacts the performance of the pendulum control. Future work, therefore, concentrates on considering channel conditions when allocating resources, e.g., in cell edge and mobility scenarios. Furthermore, spectral efficiency is aimed to be improved by optimizing scheduling decisions regarding latency limits for specific use cases in scaled scenarios. In parallel, event-triggered model predictive control can be utilized to reduce communication effort and thus increase energy as well as spectral efficiency.

Acknowledgment

The authors would like to thank Julian Golembiewski (Institute of Energy Systems, Energy Efficiency and Energy Economics, TU Dortmund University) and Pascal Jörke (Communication Networks Institute, TU Dortmund University) as well as Jonas Stehr for their contributions that laid the groundwork on which this paper is built. This work has been partly funded by the Federal Ministry of Education and Research (BMBF) via the project *6GEM* under funding reference 16KISK038 and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project number 508759126.

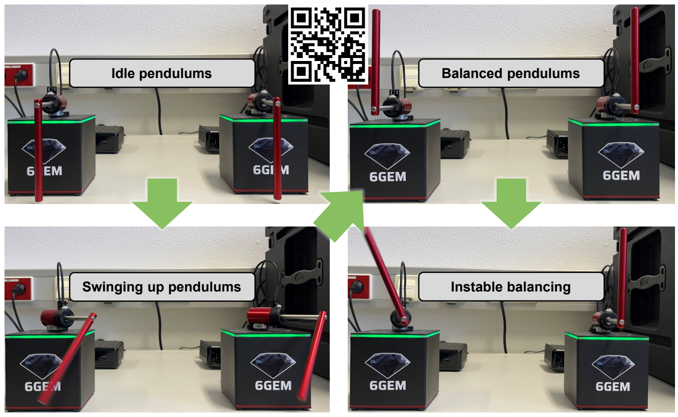

Fig. 6: To watch the pendulum testbed showcase video, scan the QR-Code or use the following link: http://tiny.cc/6GEMPendulumDemo

References

[1] R. Wiebusch, N. A. Wagner, D. Overbeck, F. Kurtz, and C. Wietfeld, "Towards Open 6G: Experimental O-RAN Framework for Predictive Uplink Slicing," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 4834–4839.

[2] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Applications of Machine Learning in Resource Management for RAN-Slicing in 5G and Beyond Networks: A Survey," *IEEE Access*, vol. 10, pp. 106 581–106 612, 2022.

[3] S. Ali, N. Rajatheva, and W. Saad, "Fast Uplink Grant for Machine Type Communications: Challenges and Opportunities," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 97–103, 2019.

[4] P. Yang, X. Xi, *et al.*, "How Should I Orchestrate Resources of My Slices for Bursty URLLC Service Provision?" *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1134–1146, 2021.

[5] M. Alsenwi, N. H. Tran, *et al.*, "Intelligent Resource Slicing for eMBB and URLLC Coexistence in 5G and Beyond: A Deep Reinforcement Learning Based Approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4585–4600, 2021.

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

[7] A. Casparsen, B. Soret, J. J. Nielsen, and P. Popovski, "Near Real-Time Data-Driven Control of Virtual Reality Traffic in Open Radio Access Network," in *Globecom 2023 - IEEE International Conference on Communications*, 2024.

[8] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Orchestrating Network Intelligence in the Open RAN," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2023.

[9] C. Bektas, D. Overbeck, and C. Wietfeld, "SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[10] D. Johnson, D. Maas, and J. Van Der Merwe, "NexRAN: Closed-Loop RAN Slicing in POWDER -A Top-to-Bottom Open-Source Open-RAN Use Case," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*, New Orleans, LA, USA: ACM, 2021, pp. 17–23.

[11] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[12] J. Apkarian and M. Levis. "Qube-Servo 2 Experiment for Matlab/Simulink Users: Instructor Workbook." (2016), [Online]. Available: https://www.quanser.com/products/qube-servo-2/.

[13] R. Verschueren, G. Frison, *et al.*, "Towards a modular software package for embedded optimization," in *Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, 2018.

[14] I. Gomez-Miguelez, A. Garcia-Saavedra, *et al.*, "srsLTE: an open-source platform for LTE evolution and experimentation," Oct. 2016, pp. 25–32.

[15] 3GPP, "E-UTRA; Relay radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.116, Apr. 2022, Version 17.0.0.