Accepted for presentation in: 2022 IEEE Global Communications Conference (GLOBECOM)

Proactive Resource Management for Predictive 5G Uplink Slicing

Dennis Overbeck, Niklas A. Wagner, Fabian Kurtz, Christian Wietfeld
Communication Networks Institute, TU Dortmund University, Otto-Hahn-Strasse 6, 44227 Dortmund
Email: {dennis.overbeck, niklas.wagner, fabian.kurtz, christian.wietfeld}@tu-dortmund.de

***Abstract*—The 5th generation of mobile communication networks (5G) introduced the concept of network slicing for enabling multiple, diverging service types by providing virtually independent communications within one physical network. While Ultra-Reliable Low Latency Communication (URLLC) aims to provide latency guarantees below 5 ms for mission-critical applications such as Smart Grid as well as Industry 4.0, Enhanced Mobile Broadband (eMBB) focuses mainly on high data rates. Thus, since different Key Performance Indicators (KPIs) such as latency, data rate or time-criticality need to be considered, the allocation of resources between corresponding slices is challenging. This work, therefore, aims to reduce uplink latency for URLLC transmissions by deploying Proactive Grants to minimize the impact of time-consuming scheduling requests and any negative influence on other slices. Resources are allocated proactively by base stations utilizing Machine Learning (ML) models trained on real-world measurements. An experimental evaluation via an Software-Defined Radio (SDR)-based physical testbed demonstrates delay reductions towards the mission-critical threshold while simultaneously increasing spectral efficiency. Compared to Round Robin (RR)-based slicing, latency decreases by 49 %, while maintaining a high throughput of 98 % in the eMBB slice.**

I. Introduction

For the current generation of mobile communication (5G), three heterogeneous service requirements are specified [1], aiming for disparate Quality of Service (QoS) profiles. These service types comprise according to [2]:

- *Enhanced Mobile Broadband (eMBB)*:
  High data rates within wide areas and hotspot scenarios.
- *Massive Machine Type Communication (mMTC)*:
  Low cost connectivity for massive amounts of devices with low traffic volumes and medium delays.
- *Ultra-Reliable Low Latency Communication (URLLC)*:
  Latencies below 5 ms for mission-critical services.

As each service type requires a different strategy for resource allocation regarding timing and bandwidth, network slicing emerged as key enabler, providing virtually independent and isolated communications on top of a single physical network, such as depicted in Fig. 1 for a modern urban environment. Here, for example, applications requiring high data rates typically consume as many Physical Resource Blocks (PRBs) per timeslot as possible and utilize greater Subcarrier Spacings (SCSs) for higher bandwidths. Conversely, latency-critical applications commonly need less spectrum but readily available time slots. For Long Term Evolution (LTE) (i.e. 4G), several schedulers have been designed to distribute resources fairly or consider device-specific Key Performance Indicators (KPIs). However, scheduling is mostly based on best effort while prioritization is limited to features such as Access Class Barring (ACB). With network slicing, users are processed according to their respective slice priority, with scheduling algorithms, for instance, guaranteeing certain levels of latency.

In previous works, we performed static reactive network slicing using Software-Defined Radios (SDRs) and custom scheduling algorithms [3] as well as a novel approach utilizing Configured Grants (CGs) within a simulation framework [4], which was further examined analytically regarding worst case response times [5]. To achieve the reliability and low latency needed for mission-critical communications, predictive aspects need to be considered. The main contributor to uplink (UL) latency within the Radio Access Network (RAN) is the waiting time for sending requests and receiving grants for each transmission. By utilizing CGs, introduced with Release 15, the time-intensive sending of scheduling requests can be mitigated, by granting radio resources proactively. While CGs mainly focus on periodic traffic, the Proactive Grants (PGs) can be scheduled more dynamically to fit several use cases and traffic shapes. Here, using Downlink Control Information (DCI) packets, resource allocation is proactively announced to User Equipments (UEs) so when data needs to be transmitted they can use the already reserved slots, rendering scheduling requests obsolete. In contrast, CGs are specified using Radio Resource Control (RRC) signaling. Overall, proactive resource management and provisioning offers opportunities but also limitations, as detailed in the following.

5G offers three options to access UL resources and lower one-way delay for slicing:

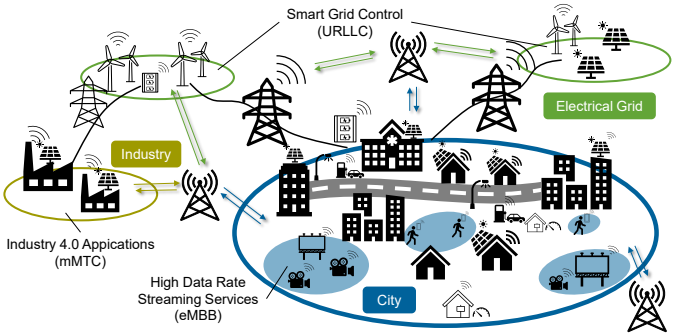

Fig. 1. Network slicing enables 5G to simultaneously adapt to multiple service types. The scenario comprises a latency-critical URLLC and a data rate intensive eMBB slice, competing for resources.

1) *Grant-Free Access* enables base stations to provide reserved resources for a UE in a dynamic or periodical manner, thus eliminating time-intensive scheduling requests. Type 1 CGs are designated for this purpose, as they are particularly suitable for highly periodic transmission patterns. However, if collisions or loss of packets appear, i.e., when reserved resources are shared among devices, mechanisms like Hybrid Automatic Repeat Request (HARQ) or *k-repetitions* procedures need to be implemented, which in turn may decrease spectral efficiency and possible gains from utilizing them [6].
2) The *reduction of Scheduling Request (SR) periodicity* may lead to more frequent opportunities to send data for prioritized UEs, however at the cost of increasing the control overhead within a grant-based network. To cope with this increase for URLLC slices, other less latency-sensitive slices such as mMTC may receive less Scheduling Request Occasions (SROs) in turn. However, as the number of devices increases, the network load increases accordingly with the frequency of SROs.
3) *Proactive Grants* offer the same advantages as grant-based scheduling regarding exclusively assigned resources, while skipping latency-intensive scheduling requests. Therefore, spectral efficiency is increased by mitigating transmission collisions and upholding QoS requirements. Utilizing the End-to-End (E2E) slicing concept, the resources can be assigned exclusively within an Management and Orchestration (MANO) framework, guaranteeing latencies and preventing latency peaks.

Efficient Radio Resource Management (RRM) mitigates wasteful radio resource usage and enables time-critical provisioning utilizing one or multiple of the above mentioned approaches. This work thus focuses on scheduling aspects for RAN resources on the Medium Access Control (MAC) layer. Related work is presented in Sec. II, comprising an overview of current research on 5G network slicing. Sec. III provides a description of our novel approach based on predictive scheduling. Next, the evaluation scenario and results are discussed and put into context in Sec. IV. Finally, Sec. V summarizes key insights and gives an outlook on future work.

II. Related Work

The focus of this section is on different approaches regarding scheduling of resources within wireless networks in combination with Machine Learning (ML)-based algorithms. While the typical approach for scheduling is still the Round Robin (RR) or Proportional Fair (PF) algorithm, the predictive scheduling is on the rise in research and set as key aspect for the 5G and beyond networks. In [3], the RR scheduling approach is extended to provide network slicing features and hard guarantees for the tenants of each network slice. Prioritization takes place in a timely manner, therefore a token mechanism is used: higher priorities allow more tokens or transmissions to be performed, before the next tenant/slice is scheduled to send packets. This leads to the loss of prioritization after the tokens are depleted. In [7] and [8], scheduling algorithms based on QoS priorities are proposed as well as grant-free allocation of resources for URLLC transmissions, also considering different types of traffic. Especially for grant-free accesses, their utilization for highly critical traffic is highlighted in works by [6], [9] and [10]. There are also several works on using ML-based scheduling approaches. Here, [11] gives an overview of combining deep learning algorithms with resource allocation in wireless networks. In [12], a combination of deep and reinforcement learning is proposed to tackle the complexity of network dynamics on the RAN for both large and small time-scale prediction. The authors of [13] discuss a combination of priority-based PF algorithm with Q-Learning to improve reliability and latency of joint URLLC and eMBB traffic. Works of [14] and [15] focus on slice orchestration based on channel quality. [14] concentrates on the reduction of Channel Quality Indicator (CQI) reporting frequency while providing a low-complexity slice orchestration, utilizing time-series prediction, whereas [15] uses real-world measurements to provide an Long Short-term Memory (LSTM)-based model for predicting PRB utilization on a *millisecond* timescale. Previous works emphasized analytical models and simulation-based studies to the best of our knowledge, whereas this work concentrates on a proof-of-concept in an experimental hardware setup using an ML-based approach to predictive uplink scheduling for network slicing.

III. Proactive Resource Management for Reliable Network Slices

Proactively assigned radio resources are enabled by efficient RRM, which can be improved by utilizing ML-based approaches as described in the previous Sec. II. Standardized scheduling schemes for allocating resources on the RAN are shown in Fig. 2. Conventional approaches rely on reactive scheduling, i.e. SROs are communicated by the base station to the UE such that transmissions can be announced to the base station via SRs. This procedure is the most time-consuming approach, since waiting times result from awaiting the occasion to send but also the request itself, and waiting for the resource allocation by the base station, as depicted on the left-hand side of the figure.

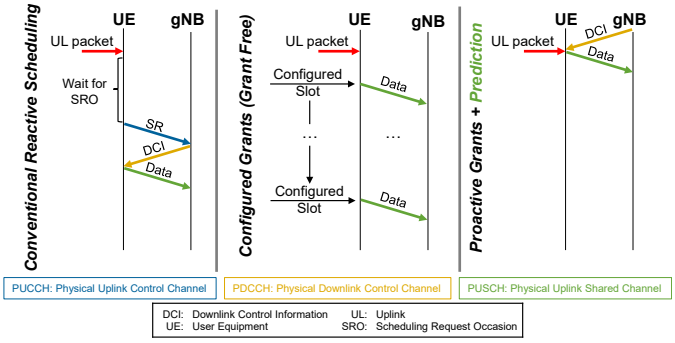

Fig. 2. 5G provides various resource allocation mechanisms for use in RAN slicing. These can be categorized into conventional reactive, grant-free and PG scheduling. The proposed approach combines proactive resource allocation with data traffic prediction, as depicted on the right.

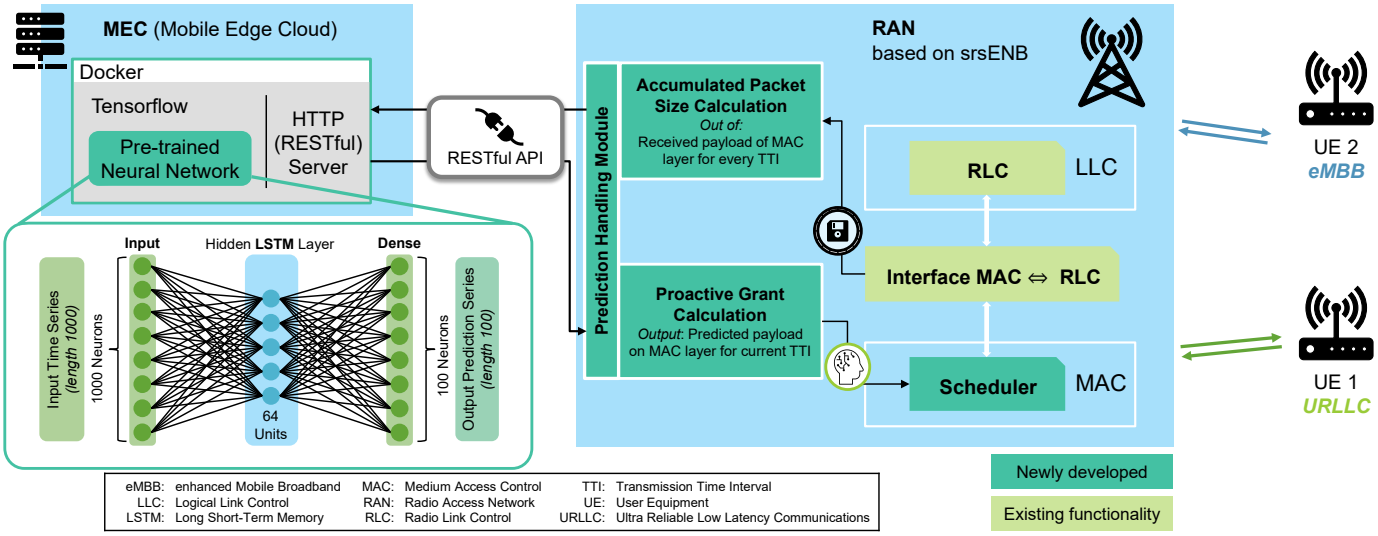

Fig. 3. The evaluation setup is designed to include a RESTful API for splitting the computationally intensive prediction from the base station towards, e.g., a MEC. The scheduler of the base station monitors packets received on the MAC/RLC interface and forwards the accumulated packet sizes to the prediction module. Based on output data, the base station then proactively schedules the grants.

For highly periodic traffic, the 5G standard enables base stations to provide fixed resource blocks for each UE in a periodic manner through CGs. Here, certain recurring resource blocks are reserved for transmission of a certain UE, which is typically communicated for larger time periods. Therefore, the need for SRs vanishes, however resulting in PRB wasting if the resources remain unused, since they are exclusively assigned to a device. Furthermore, this approach may still lead to longer waiting times whenever recurring CGs are far apart. Yet another approach for scheduling resources is given with PGs, where the base station can send DCI messages to the UE, thus providing resources, before a SR is sent. Since a UE typically expects this information before the transmission, PGs are generally compatible with previous generations of mobile radio networks, although not standardized. In this work, these grants are combined with ML to optimize resource allocation and match the UEs' future requests. Thereby, latencies incurred by waiting are reduced massively, while maintaining high spectral efficiency as depicted by Fig. 2.

In Fig. 3, the overall concept of the used framework is visualized. The base station constantly monitors packets received on the interface from MAC layer to RLC layer and accumulates the calculated packet sizes for the last amount of $n = 1,000$ Transmission Time Intervals (TTIs), which turned out to be a sufficient basis for the prediction.

TABLE I
TRAINING PARAMETER SETTINGS

Learning Rate	10^{-4}
Layer Structure	LSTM (vanilla, 64 Units) Dense Layer (Activation: linear)
Batch Size	4
Epochs	16
Loss	MSE
Optimizer	ADAM

Via REST API, the calculation is given to the MEC, which comprises the RESTful server for interaction and a Tensorflow-implemented [16] ML model for application. Our prediction model is based on LSTM, which is suitable for time series and traffic prediction as shown in [15] and other works. It is trained with data from real-world measurements[1] containing control transmissions in form of International Electrotechnical Commission (IEC) 60870 messages. For training the model, Keras with Tensorflow [16] backend is used.

The utilized data set is split into ⅔ training and ⅙ data for validation respectively evaluation. Cross-validation is conducted to perform hyper-parameter tuning using the *Hyperband Tuner* algorithm. Training, using the final hyperparameter configuration given in Tab. I, results in an accuracy of $92.45\,\%$. Test data is then transmitted via the radio interface as input to the base station by a UE. After the trained model anticipates the demands, these payload predictions are used by the base station to allocate resources accordingly.

Our framework consists of an SDR-based LTE software stack based on the srsRAN project [17], in which we integrated our proposed scheduling algorithm as well as our adaptations for prediction extension. The setup is deployed in virtualized containers to enable flexible and rapid use in Edge-Cloud scenarios, partly based on the previous works of [18]. As core network the nextEPC project [19] is used. Building on the concepts of [4], a Proactive rather than Configured Grant approach is used. The model is pre-trained on the training subset, since prediction would otherwise take too long to be available in time. Payloads are predicted in an asynchronous model execution loop and constantly fed back to the base station. Prediction needs to be performed on a *millisecond* basis. Our CPU- as well as GPU-based evaluation showed that LSTM achieves this goal while offering high precision.

[1]Anonymized data set available under the following link: https://github.com/overbeckd/TransmissionDataSet

The execution duration of predictions including transmission from the MEC to the base station lies within an interval of $t_{Prediction} = 14 - 34\,\mathrm{ms}$. Although the design goal of the prediction model is a fast execution time, continuous in-time calculation of the next prediction step is challenging since the scheduling is performed once every *millisecond*. Thus, multiple steps are predicted simultaneously and are processed by the prediction handling module in sets of multiple time slots. Predictions are initiated consecutively. To determine the optimal amount of predicted time steps, the elapsed time during a prediction process k and the duration of the subsequent prediction process $k+1$, while the predicted values of k are used, need to be considered. This results in the algebraic relation depicted in (1). In this specific case we set the number of predicted time steps to a value of 100 including a safety margin for irregularities.

$$\#predictedSteps \geq 2 \cdot \frac{t_{Prediction}[\mathrm{ms}]}{\mathrm{ms}} \tag{1}$$

The implementation is based on a TTI of $1\,\mathrm{ms}$, according to the 3rd Generation Partnership Project (3GPP) LTE standard Release 10. Therefore, scheduling requests are expected to be sent within this time frame. As a result, the subframe length and the comprised slots for resources are equal to a SCS of $15\,\mathrm{kHz}$ in the 5G standard. UEs of the URLLC slice are allocated resources according to their Cell Radio Network Temporary Identifier (C-RNTI), which is exclusively provided once a UE connects with the base station and is thus in the RRC state *connected*. Resource allocation is performed as follows: First, the amount of data a UE may send and volume predicted by our model is fed into the calculation module for PG dimensioning. This is done before the regular, reactive scheduling process is performed, i.e., resources are not scheduled yet but calculated afterwards. Then, necessary UL resources are computed, estimating channel quality by measures of Signal plus Interference to Noise Ratio (SINR) or CQI and setting Modulation and Coding Scheme (MCS) accordingly. The prediction itself does not predict the channel quality but focuses on the amount of data to be transmitted, returning the predicted packet size. Regarding channel quality, the scheduling relies on the measurements by the base station since this work focuses on mostly stationary devices, also reflecting the chosen scenario described in Sec. IV, and thus, volatile channel conditions are not expected. In this step, the PG payload size is given to the calculation module, which calculates the exact PRBs to allocate. These PRBs are next added to the pending UL grants including the calculated MCS, which is then transmitted as DCI via Physical Downlink Control Channel (PDCCH) towards the UE. The scheduling procedure is presented in Algorithm 1. UEs need to have the ability to deal with the PGs, to skip sending a SR and to transmit buffered packets directly, which are supported features of the srsRAN project.

Algorithm 1 Joint URLLC and eMBB predictive scheduling

Input: $UEList$ consisting of UE_i including scheduling- and slice-specific properties

Output: Assigned PRBs in form of Grant Vector Θ

$SortedUEList \leftarrow$ Sort($UEList$,$Priority(UE_i)$)
$\Theta \leftarrow \emptyset$
for UE in $SortedUEList$ **do**
  if $Slice(UE)$ = 'URLLC' **then**
    Fetch predicted packet sizes $\hat{X}$ of $UE.rnti$
    $\hat{X}_{lim} \leftarrow \min(\hat{X}, UE.maxBudget)$
    $\gamma \leftarrow$ ProactiveGrantCalc($\hat{X}_{lim}$)
    $assignedPRBs \leftarrow$ ContiguousPRBCalc(γ, Θ)
    $\Theta \leftarrow \Theta \cup (assignedPRBs, UE)$
  end if
  Process pending SRs(UE) on RR basis and add to Θ
end for

IV. Evaluation

In the following section, the hardware-based test setup utilizing SDRs is introduced. Moreover, the evaluation scenario and the related results are presented and discussed.

A. Setting the Scene: Setup and Scenario Description

An overview of the setup used for this work is given by Fig. 4. The experimental platform consists of a server with an AMD Ryzen 5900X CPU and $32\,\mathrm{GB}$ of RAM used as eNodeB. Two Intel NUCs with a Core i7-6770HQ, $16\,\mathrm{GB}$ RAM and Ubuntu 20.04.2 LTS (kernel: 5.11.0-34-lowlatency) connect to Ettus Research USRP B210 SDRs and serve as UEs. A wired setup is chosen to minimize interference, utilizing two Wilkinson splitters for connecting UEs to the base station. All devices are synchronized by the Precision Time Protocol (PTP) to facilitate precise measurement of end-to-end one-way delays. As depicted in Fig. 4, the scenario includes two slices, each with one SDR-based UEs. The first one acts as critical URLLC slice, which is based on control messages derived from real-world measurements of IEC 60870 traffic, whereas the other acts as less delay-sensitive eMBB slice, where loads are generated via iPerf v3.7.

The channel in uplink direction is capable of transmitting a Transport Block Size (TBS) of $12.576\,\mathrm{Bit}$ per TTI, according to [20], with a MCS of 23 and 25 PRBs available for scheduling on a bandwidth of $5\,\mathrm{MHz}$. This evaluation not only demonstrates latency reductions by harnessing LSTM for proactive allocation, but also increases spectral efficiency.

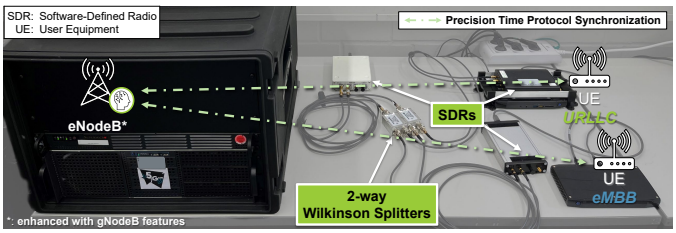

Fig. 4. The experimental platform includes a base station (left-hand side) and two compact computers acting as UEs (on the right). To mitigate interference, a wired setup with a total of three SDRs is employed. Prediction is executed on an off-site Edge-Cloud but may also be integrated into the base station.

B. Evaluation Results

The evaluation is performed using the proposed LSTM-based PG algorithm on the base station. Measurements compare classic RR and PF scheduling with our approach concentrating on latency and throughput, since both in combination imply a certain trade-off. Within the evaluation, the srsRAN-based open-source LTE stack is used in Frequency Division Duplex (FDD) mode. Since the focus lies on the UL scheduling, the end-to-end one-way delay is measured, i.e., the time a packet needs from being sent out by the application on UE side until reception at the Packet Data Network Gateway (P-GW). From left to right, the different scheduling algorithms are depicted in Fig. 5 with their respective impact on packet delays. The reactive static slicing is taken as reference from previous works [3], whereas RR and PF algorithms are based on the standard schedulers provided by the srsRAN project. Starting with the legacy scheduling algorithms, the left-hand side of the figure shows the results achieved with the PF and RR schedulers with a mean latency of $30.3\,\mathrm{ms}$ and $27.4\,\mathrm{ms}$. In general, it is noticeable that the average latency accumulates over a time frame of approximately $20\,\mathrm{ms}$, consistent with the frequency of scheduling request occasions. Both algorithms show outliers ranging between $41\,\mathrm{ms}$ and $13\,\mathrm{ms}$, while the RR algorithm performs slightly better regarding the mean latency. In comparison, the RR and PF scheduling algorithms achieve higher mean latency results than the slicing-based approaches, and additionally do not provide hard service guarantees. Round-robin based static slicing of [3], which enables the mentioned hard service guarantees but without strongly focusing on latency, is shown by the middle plot. Here, a mean one-way delay of $24.0\,\mathrm{ms}$ is achieved. The reactive static slicing achieves a slightly lower delay than the conventional schedulers since the token-based approach ensures priority, but still reacts to scheduling requests and delays them if the available tokens are depleted. However, the resources are constantly given to the highest priority slice first, thus enabling the network slicing concept on the RAN.

All of these scheduling approaches do not reach the goal of limiting the latency for mission-critical transmissions to a maximum of $5\,\mathrm{ms}$, thus are unsuitable for control messaging in Smart Grid scenarios. In contrast, our approach enables low latencies in combination with guaranteed prioritization of packets leveraging network slicing using proactive allocation of resources. On the right-hand side of Fig. 5, the proposed ML-driven proactive scheduling algorithms are depicted. The second to right plot shows the best achievable results in one-way latency by reserving resources for the anticipated packet sizes for every time slot. Thus, UEs are capable of sending their data directly when a packet is generated, eliminating scheduling requests completely. As a consequence, the one-way delay can be reduced to a mean of $4.5\,\mathrm{ms}$ with our approach, when reserving a constant amount of resources, also mitigating larger deviations in latency. However, this leads to less available data rate and spectral efficiency for the overall network and other network slices. To achieve a better trade off between critical slices latency and overall network throughput, we enhanced this approach by predicting the amount of data and the time slots the packets appear. The results can be seen in the violin plot on the right-hand side. Here, the mean latency goes up, in comparison to the static proactive slicing, to $12.1\,\mathrm{ms}$, which is due to falsely predicted time slots and thus the necessity of sending scheduling requests. This leads to outliers ranging up to $41\,\mathrm{ms}$, however an accumulation point can be recognized around the $5\,\mathrm{ms}$ threshold. Nevertheless, this approach decreases the mean delay in comparison to the RR-based slicing scheduler by $11.9\,\mathrm{ms}$ (approx. $49.4\,\%$).

Beyond latencies, throughput in the eMBB slice is examined in Fig. 6. From left to right, the achieved data rates of the corresponding scheduling algorithms are shown in the same order as in the previous Fig. 5. Here, the static proactive slicing approach shows the impact of reserving resources for the prioritized slice and the associated trade off on the throughput of other slices. The amount of configured data rate results from the reservation of resources in each time slot, although the actual transmission is only present at certain points in time. Therefore, a significant reduction of impact on

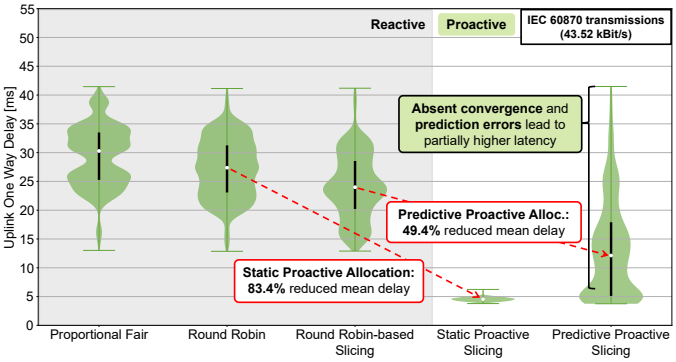

Fig. 5. Latency comparison of various reactive legacy and the proposed novel predictive proactive network slice uplink scheduling algorithms. Static proactive slicing (second from the right) enables a mean of $4.5\,\mathrm{ms}$ one-way latency at the cost of overall network throughput due to overly generous resource reservation. This loss of spectral efficiency is mitigated by utilizing predictive methods, enabling precise resource allocation.

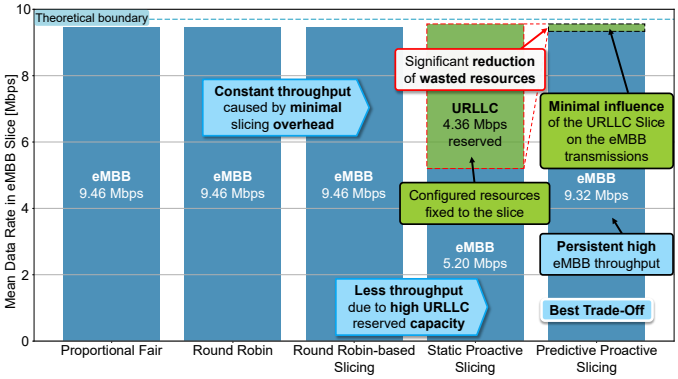

Fig. 6. Comparison of data rates between different scheduling algorithms, highlighting the impact of prioritization on eMBB slices. A massive gain of spectral efficiency is achieved relative to fixed resource reservations for the URLLC slice and predicting those time-slots when resources are actually needed. Combined with lowered latencies, the novel predicted proactive slicing approach thus constitutes the best trade-off.

the overall data rate in the network can be achieved by utilizing prediction modules to provide resources only in the time slots, where there is an expected transmission, as can be seen on the right-hand side of the figure. Here, the data rate for the eMBB slice achieves 98.5% in comparison to the PF scheduler without such latency guarantees. Therefore, the results show the capability of our proactive approach to enable substantially increased QoS demands for low latency communications. Furthermore, the considered trade-off between URLLC and eMBB slices does not result in systematic waste of resources but rather increased jitter in the less critical transmissions of the eMBB slice, which can be mitigated by buffering packets (as stated in [2] up to $300\,\mathrm{ms}$).

Limitations in our approach to predicting resources result from the offline learning method, which prevents further learning based on misallocated resources. Therefore, results can be enhanced by implementing additional online learning capabilities to mitigate convergence issues. Moreover, the prediction algorithm in the current state does not know when the packet is generated, and thus how long it has been saved for the Buffer Status Report (BSR) waiting for a SRO. As a consequence, the foundation for the prediction might be biased. That being said, if 100% accuracy can be achieved, no resources would be wasted, and thus SRs became obsolete resulting in zero latency induced by scheduling operations. However, unforeseen conditions such as irregularities in the transmissions but also external influences on the physical layer caused by, e.g., heavy rain storms prevent conclusive prediction for every conceivable use case.

V. Conclusion and Outlook

In this work, the usage of Proactive Grants for reliable, low-latency RRM is evaluated in terms of latency and data rate trade-offs based on an experimental SDR platform. Scheduling is enhanced by an ML-based prediction module, which demonstrates significant steps towards the $5\,\mathrm{ms}$ threshold for mission-critical communication. Our proposed proactive slicing approach reduces one-way latency by $49\,\%$ compared to RR-based slicing, while simultaneously achieving $98\,\%$ of the eMBB slice data rate. In future work, this approach will be ported to comply with O-RAN specifications and thus be deployed as xApp. Also, further work on more sophisticated ML-based prediction models will be performed to enhance accuracy on *millisecond* timescales.

Acknowledgment

This work has been supported by the Federal Ministry of Education and Research (BMBF) in the course of the project *6GEM* under the funding reference 16KISK038 and by the Federal Ministry for Economic Affairs and Climate Action (BMWK) in the course of the project *5Gain* under the funding reference 03EI6018C.

References

[1] ITU-R, "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond," International Telecommunication Union - Radio Communication Sector, Tech. Rep. Recommendation ITU-R M.2083-0, 2015.

[2] 3GPP, "Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2; Release 17," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, Sep. 2022, Version 17.4.0.

[3] C. Bektas, S. Böcker, F. Kurtz, and C. Wietfeld, "Reliable Software-Defined RAN Network Slicing for Mission-Critical 5G Communication Networks," in *IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, Hawaii, USA, Dec. 2019.

[4] C. Bektas, D. Overbeck, and C. Wietfeld, "SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling," in *IEEE International Conference on Communications*, Montreal, Canada, Jun. 2021.

[5] A. Nota, S. Saidi, D. Overbeck, F. Kurtz, and C. Wietfeld, "Providing Response Times Guarantees for Mixed-Criticality Network Slicing in 5G," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 552–555.

[6] Y. Liu, Y. Deng, M. Elkashlan, A. Nallanathan, and G. K. Karagiannidis, "Analyzing Grant-Free Access for URLLC Service," *IEEE Journal on Selected Areas in Comm.*, vol. 39, no. 3, pp. 741–755, 2021.

[7] A. Karimi, K. I. Pedersen, N. H. Mahmood, G. Pocovi, and P. Mogensen, "Efficient Low Complexity Packet Scheduling Algorithm for Mixed URLLC and eMBB Traffic in 5G," in *IEEE Vehicular Technology Conference (VTC-Spring)*, 2019, pp. 1–6.

[8] T.-K. Le, U. Salim, and F. Kaltenberger, "Improving Ultra-Reliable Low-Latency Communication in multiplexing with Enhanced Mobile Broadband in grant-free resources," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–6.

[9] T.-K. Le, U. Salim, and F. Kaltenberger, "Enhancing URLLC Uplink Configured-grant Transmissions," in *IEEE Vehicular Technology Conference (VTC-Spring)*, 2021, pp. 1–5.

[10] C. Liang, S. Xia, X. Han, and P. Hao, "Configured Grant Based URLLC Enhancement for Uplink Transmissions," in *International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 1053–1058.

[11] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[12] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent Resource Scheduling for 5G Radio Access Network Slicing," *IEEE Transactions on Vehicular Tech.*, vol. 68, no. 8, pp. 7691–7703, 2019.

[13] M. Elsayed and M. Erol-Kantarci, "AI-Enabled Radio Resource Allocation in 5G for URLLC and eMBB Users," in *IEEE 5G World Forum (5GWF)*, 2019, pp. 590–595.

[14] S. Bakri, P. A. Frangoudis, A. Ksentini, and M. Bouaziz, "Data-Driven RAN Slicing Mechanisms for 5G and Beyond," *IEEE Transactions on Net. and Service Management*, vol. 18, no. 4, pp. 4654–4668, 2021.

[15] A. M. Nagib, H. Abou-Zeid, H. S. Hassanein, A. Bin Sediq, and G. Boudreau, "Deep Learning-Based Forecasting of Cellular Network Utilization at Millisecond Resolutions," in *IEEE International Conference on Communications (ICC)*, 2021, pp. 1–6.

[16] Martín Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015. [Online]. Available: https://www.tensorflow.org/.

[17] I. Gomez-Miguelez, A. Garcia-Saavedra, P. Sutton, P. Serrano, C. Cano, and D. Leith, "srsLTE: an open-source platform for LTE evolution and experimentation," Oct. 2016, pp. 25–32.

[18] F. Eckermann, P. Gorczak, and C. Wietfeld, "tinyLTE: Lightweight, Ad Hoc Deployable Cellular Network for Vehicular Communication," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[19] *NextEPC: Open Source Evolved Packet Core*, 2019. [Online]. Available: https://nextepc.org.

[20] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures (Rel. 16)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.213, Oct. 2021, v16.7.1.