Published in: 2021 IEEE International Conference on Communications (ICC)
DOI: 10.1109/ICC42927.2021.9500837

SAMUS: Slice-Aware Machine Learning-based Ultra-Reliable Scheduling

Caner Bektas, Dennis Overbeck, Christian Wietfeld
Communication Networks Institute, TU Dortmund University, Otto-Hahn-Strasse 6, 44227 Dortmund
Email: {caner.bektas, dennis.overbeck, christian.wietfeld}@tu-dortmund.de

***Abstract*—Multiple service types such as Ultra-Reliable Low Latency Communication (uRLLC) and Enhanced Mobile Broadband (eMBB) are envisioned to be incorporated into the next generation mobile communication standard 5G based on a single physical communication network. To unite these services with partly contradicting Quality of Service (QoS) requirements, Network Slicing is considered a key technology. uRLLC slices in particular are highly demanding, requiring extremely high reliability and low latency in the single-digit milliseconds range. Consequentially, the latency impact of radio resource management on the end-to-end latency is optimized in this work by using so-called Configured Grants (CGs), which aim to minimize latency-intensive scheduling requests by pre-allocating radio resources. As predicting future traffic demands and channel conditions are required to use CGs, a data-driven machine learning-based radio resource scheduler prototype is introduced and evaluated in this work based on a specifically developed 5G radio resource simulator. The results show promising latency optimizations and possible trade-offs in uRLLC and eMBB coexistence.**

I. Introduction

The 5th Generation of Mobile Communication Networks (5G) progresses towards incorporating various service types with partly contradicting Quality of Service (QoS) requirements into a single mobile communication network, aiming for heterogeneity regarding supported services and Key Performance Indicators (KPIs). In fig. 1 right, the three envisioned main service types and their requirements are shown, which are defined as specifications of different KPIs [1]:

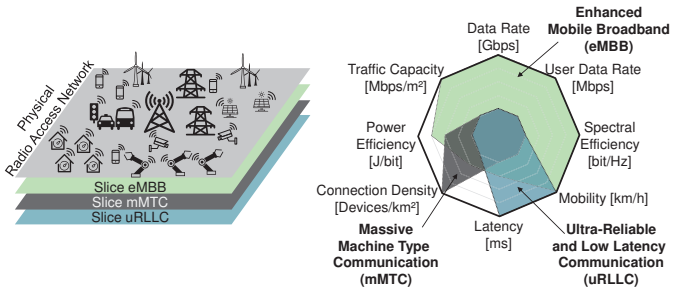


Fig. 1. Network Slicing as key enabler to fulfill all specific service requirements simultaneously [1]

- *Enhanced Mobile Broadband (eMBB)*
- *Ultra-Reliable Low Latency Communication (uRLLC)*
- *Massive Machine Type Communication (mMTC)*

Integrating these different service types into a single physical communication network is a great challenge. One promising key technology is the Network Slicing, where different virtual networks, so-called slices, are utilized on top of a common physical communication network (see fig. 1 left). Several empirical studies on Network Slicing were conducted by the authors based on Long Term Evolution (LTE), where the data rate provisioning was successful based on schedulers developed on top of Software-Defined Radio (SDR) systems [2] [3]. However, end-to-end latency requirements of uRLLC were not possible to implement due to technical limitations of LTE. For this reason, the main focus of this work is to minimize end-to-end latency in 5G networks. End-to-end latency in cellular networks and 5G specifically comprises of different latency-inducing components.

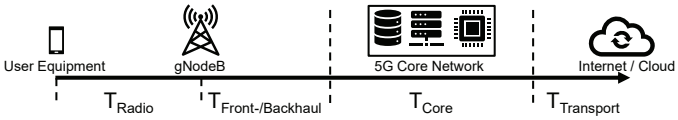


Fig. 2. Latency-inducing components of transmission in cellular networks [4]

Following components can be derived from [4] (c.f. fig. 2 from right to left):

- $T_{Transport}$: Latency induced by the transport network, e.g., when requesting data from or uploading data to the Internet.
- T_{Core}: The core network present in most cellular networks introduces some latency while processing and forwarding data from and between the mobile and wired network.
- $T_{Front-/Backhaul}$: The connection between gNodeB (5G base station) and core network also induces further delay.
- T_{Radio}: This is where the physical channel characteristics, e.g., distance from User Equipment (UE) to gNodeB, as well as Radio Resource Management (RRM) add transmission latency.

The focus of this work is to minimize T_{Radio}, in particular that of RRM as a major part in implementing QoS in Network Slices. Specifically, the process of scheduling requests and granting them can be eliminated using the so-called Configured Grant (CG) scheduling, which is planned to be integrated into 5G [5]. CG scheduling allows the definition of fixed resource allocations in the future, thus eliminating the need for latency-inducing scheduling operations. However, this concept has one major challenge: Future data demands and channel qualities have to be predicted to allocate the exact amount of Resource Blocks (RBs) needed in each slice. This again

imposes a major challenge: If the prediction of required RBs is too low, mission-critical data in the uRLLC slice has to be scheduled again, highly increasing its end-to-end latency. A possible solution would be to over-provision the amount of RBs, which then results in wasting of resources for the other slices, effectively decreasing spectral efficiency. In this work, our Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS) system is introduced, which is an RRM scheduler prototype using CGs. SAMUS utilizes data-driven concepts to predict future traffic demands and thus optimizes the trade-off between latency in uRLLC and data rate in eMBB slices. For this, a specifically implemented 5G-based radio resource simulator is used to evaluate the performance of the scheduler prototype regarding different trade-off strategies.

The remainder of this work is structured as follows: In section II, a general overview of related works regarding data-driven Network Slicing concepts is given. Details of the developed simulation framework as well as a description of the data-driven scheduler prototype is presented in section III, which is rounded up by its evaluation in section IV. Finally, section V presents a conclusion and evaluates possible future work based on our concepts.

II. Related Work

Several related works exist regarding Machine Learning (ML)-based or data-driven Network Slicing, minimizing latencies in 5G, and predicting future traffic demands. In [6], Network Slicing applications, scenarios, and tasks are given, where utilization of automation with data-driven concepts appears to be a promising solution. The work of Calabrese et al. [7] concentrates on handovers between base stations for changing channel qualities and provides an overview of different ML concepts. Both surveys focus on an efficient RRM, which is dependent on reliable scheduling operations by the base station. Bag et al. [8] study the impact of multi-numerology and shortened Transmission Time Intervals (TTIs), which were introduced with the 5G standard, on the scheduling latencies. The utilization of CGs in this context is analyzed in [5], where detection schemes are introduced to provide reliability for CG transmissions by minimizing collisions for resources reserved for multiple UEs at the same time. As for the predictive aspect of the scheduler, time series prediction was evaluated in [9] for Internet of Things (IoT) applications, where the findings for historical data favor the auto-regression based Auto-Regressive Integrated Moving Average (ARIMA) model.

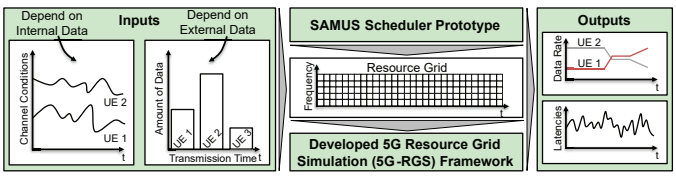


Fig. 3. Overview of the system comprising inputs, outputs, and modules

III. Methods

In the following subsections, the overall developed system comprising the 5G Resource Grid Simulation (5G-RGS) framework and the RRM scheduler prototype SAMUS is described. For this, fig. 3 presents an overview of all inputs, outputs, and modules contained in the system.

A. 5G Resource Grid Simulation (5G-RGS) Framework

In the 5G-RGS framework, a resource grid is modeled as a matrix based on the 5G specifications. As presented in fig. 3, the channel conditions of each UE serve as an input to the 5G-RGS, which combined with other parameters such as modulation order, TTI, as well as the allocated RBs, yields the Transport Block Size (TBS). The TBS is the key element in calculating the resulting data rate of each slice, which is conducted as follows [3] [10]:

$$\textit{Data Rate (Mbps)} = 10^{-6} \cdot \sum_{n=1}^{N_{UE}} \left(\frac{TBS^{(n)}}{TTI} \cdot N_{TTI}\right) \quad (1)$$

where N_{UE} is the amount of UEs within the slice, $TBS^{(n)}$ is the TBS allocated for the n-th UE in bit, and N_{TTI} is the amount of TTIs in a second. One TTI has a duration of 1 ms in this work, based on default New Radio (NR) specification.

Moreover, the latency of each transmitted packet is calculated as follows (media access control layer only):

$$\textit{Latency (ms)} = (I_S - I_C) \cdot TTI \quad (2)$$

where I_C is the scheduling interval the packet was created and I_S is the scheduling interval where the last bit of the packet is transmitted in an RB (thus, packets can be split up into multiple RBs).

Retransmissions, e.g., due to packet errors occurred in the radio channel or in the core network, or delays regarding the physical transmission over the air, are neglected. Thus, only RRM scheduler performance is evaluated. The framework was validated successfully recreating scenarios from [3].

B. Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS)

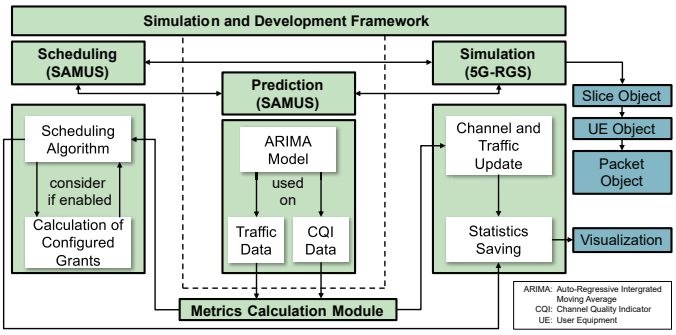


Fig. 4. Interactions and overview of modules within the simulation and development framework

As already mentioned in section I, the Slice-Aware Machine Learning-based Ultra-Reliable Scheduling (SAMUS) system is a scheduler prototype utilizing data-driven methods to allocate Configured Grants (CGs). Based on real systems, channel conditions or Channel Quality Indicators (CQIs), as well as emerging data amounts of each UE, also called Buffer Status Reports (BSRs), serve as input for the SAMUS system (see fig. 3). Apart from its capability to process traditional Scheduling Requests (SRs), which are latency-intensive, historical data is used by SAMUS to generate CGs in order to reduce the scheduling latency to zero, in the best case. This is done via prediction of traffic and CQI data, which is detailed in fig. 4. For this, the ARIMA method was utilized, which is shown to be successfully predicting time-series in [9]. Both the traffic and CQI data are used to allocate CGs in the future, as both essentially determine the amount of RBs needed for each slice or UE. For this, the scheduler first allocates the resources of the mission-critical slices based on the prediction of the ARIMA model, finally granting the remaining RBs to best effort slices (without QoS). This method is based on the Greedy Network Slicing scheduler in [3]. If a packet cannot be transmitted due to wrong prediction of data amounts or channel conditions (or when CGs are disabled), the traditional way of granting resources via SRs is used. After creating a CG for the current TTI, the scheduling module passes a resource grid as a matrix to the 5G-RGS, which in turn calculates the resulting data rate and delays for each slice and saves them in a statistics file for later processing. After that, the CQIs and BSR are updated and the cycle repeats.

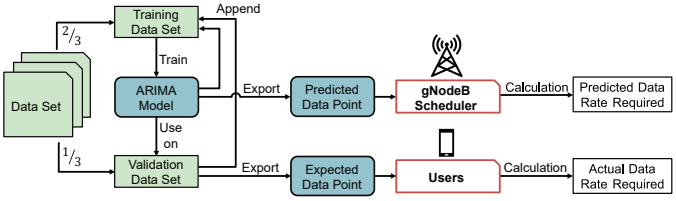


Fig. 5. Training and operation flow chart of SAMUS's prediction module using ARIMA model

As for the training and operation of the ARIMA-based prediction module, fig. 5 gives a detailed overview as a flow chart. First, the underlying data set, e.g., channel qualities and data amounts of an Electric Vehicle (EV) charging slice over a day, is split up into $\frac{2}{3}$ training data and $\frac{1}{3}$ validation data. Then, the training data set is used to train the ARIMA model to the extent that is required to predict future data (offline learning). This predicted future data is then used by the scheduler to allocate CGs, resulting in a **predicted** data rate required. Parallel to this, the **actual** data rate is calculated using the validation data set as a basis for packet generation within the UE objects. Additionally, the prediction module also adjusts its predictions based on the new data acquired during the simulation (online learning).

These methods were evaluated using real data sets for different types of slices simultaneously, while best effort users demand all resources. The results are presented in the following section IV.

TABLE I
OVERVIEW OVER THE DIFFERENT MODES AND THEIR RESPECTIVE SETTINGS USED IN THE EVALUATION OF THE FRAMEWORK

General Settings

Channel Bandwidth	20 MHz
5G Subcarrier Spacing	15 kHz
Channel Quality	Fixed Modulation and Coding Scheme (MCS) of 15
5G MCS Index Table	64QAM
SR Occasion	every 4 ms
Packet TTI	1 ms
Simulated Time	1 h

Slice-Specific Settings

Smart Grid (uRLLC)	**Electric Vehicle Charging (uRLLC)**	**Best Effort (eMBB)**
5 UEs	4 UEs	2 UEs
variable aggregated throughput	variable aggregated throughput	18.96 Mbps aggregated throughput

Mode-Specific Settings

Mode 1	**Mode 2.1**	**Mode 2.2**	**Mode 3.1**	**Mode 3.2**
No Configured Grants (CG)	Fixed CGs (optimistic)	Fixed CGs (pessimistic)	Predicted CGs	Predicted CGs
No Overprovisioning (OP)	No OP	No OP	No OP	10% OP

IV. EVALUATION

The methods described in section III were evaluated using different simulation settings and scenarios. First, underlying evaluation parameters and scenarios are introduced in section IV-A. Finally, the simulation results and their analyses are presented in section IV-B.

A. Simulation Scenarios and Parameters

To evaluate the SAMUS system, the following simulation parameters, called modes, were developed (cf. table I bottom):

- *Mode 1*: With traditional SRs → without CGs.
- *Mode 2*: With fixed CGs
  - *Mode 2.1*: Optimistic approach → fixed grants based on the historical **average** data rate of the slice
  - *Mode 2.2*: Pessimistic approach → fixed grants based on the historical **maximum** data rate of the slice
- *Mode 3*: With ARIMA-based predicted CGs
  - *Mode 3.1*: Without over-provisioning → grant resources as predicted
  - *Mode 3.2*: With over-provisioning → grant 10 % more resources than predicted

These modes represent different trade-off strategies balancing low latency in the uRLLC slices with high data rates in eMBB slices. Moreover, table I top presents all 5G radio and RRM-related settings, which are based on the average of real LTE or 5G macro-cell settings. To focus on data traffic trade-off strategies, modes $1-3.2$ are simulated with a fixed channel quality represented by a Modulation and Coding Scheme (MCS) of 15. Channel quality prediction was also analyzed but is not presented in this work. Finally, slice-specific configurations are listed in the middle area of table

I. There, three slices are defined, which are also presented in fig. 6 and detailed in the following:

- *Smart Grid (SG) slice (uRLLC)*: Data traffic in this slice is modeled after photovoltaic systems transmitting data proportionally to solar activity data obtained from National Renewable Energy Laboratory (NREL)[1]
- *Electric Vehicle (EV) Charging slice (uRLLC)*: EV charging point communication was modeled based on occupancy data gathered from *chargecloud* for the German city of Bonn[2]
- *Best Effort (BE) slice (eMBB)*: UEs and other high data rate devices are modeled in this slice as sending data with a constant rate of $18.96\,\mathrm{Mbps}$, which represents the rest of the available data rate of the cell

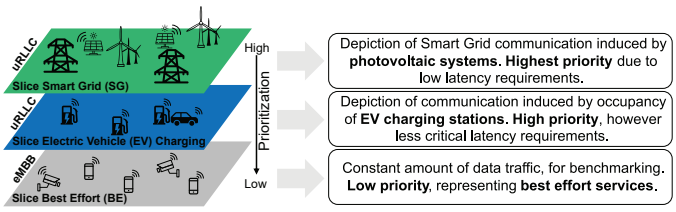


Fig. 6. Network Slicing scenario used in the evaluation of the framework

Based on these simulation parameters, modes, and scenarios, extensive evaluations were performed and are presented in the following section.

B. Evaluation Results

The simulation scenarios presented in section IV-A were performed using the 5G-RGS framework to analyze the different trade-off strategies between channel utilization (eMBB slice) and latency minimization (uRLLC slice). For this, the aforementioned modes of the SAMUS system were evaluated. Note that a channel quality prediction is not included here and a static medium coverage is assumed. This assumption can be realistic for, e.g., smart grid slices, which often consist of stationary devices and thus, do not underlie sudden changes in MCS. In first evaluations however, channel quality prediction was also performed using the SAMUS system based on ARIMA, utilizing measured CQI data of moving vehicles. These evaluations showed equally promising results as the data rate prediction presented here. Further extensive research will be performed and presented in future works.

Based on the data sources described in the previous section, a $60\,\mathrm{min}$ timeframe was analyzed in all slices and modes, according to table I. The $60\,\mathrm{min}$ timeframe represents an interval with high volatility (EV charging: rush hour with many vehicles, SG: sunrise time with rapid change in solar activity). The results for mode 1 are presented in fig. 7. On the y-axis, the average data rate of each slice is depicted in Mbps, while simulated time is represented in min on the x-axis. The lines depict different average data rate progressions. The *continuous* red line serves as an indicator of the channel bandwidth utilization, i.e., the sum of all slice data rates, while the *dotted* red line shows the maximum possible data rate of the specific cell configuration. The black line represents the average data rate of the Best Effort (eMBB) slice, while the green and blue lines show the mission-critical Smart Grid and EV charging uRLLC slices, respectively (cf. fig. 6).

As for the results, the behavior of the system with no CGs is similar to the Greedy Network Slicing algorithm presented in [3]. Based on the traditional scheduling request and grant mechanism, both mission-critical slices SG and EV charging receive all requested resources as soon as they are requested, which depicts the expected behavior. The low channel bandwidth utilization at the beginning is due to the slow increase of SG slice data rate at the start. Similarly, the BE data rate peak at about $54\,\mathrm{min}$ is due to packets being queued, until the channel allows for a higher data rate. As mission-critical slices demand higher data rates, resources of the BE slice are reduced due to being a non-critical slice. Looking at the channel bandwidth utilization (red line), the sum of actual resulting data rate is very close to the theoretical maximum. However, this data rate efficiency has a strong negative impact on overall uRLLC latency, as can be seen in fig. 9. There, the ratio between average received data rate from actual requested data rate in the BE slice is depicted on the left-hand y-axis (gray bar plot). The second y-axis on the right-hand side shows the mean *scheduling* latency resulting in the critical slices (green: SG slice, blue: EV charging slice, black lines: standard deviation). The indicated arrows and values depict the remaining margins for the other end-to-end latency components for both critical slices in their respective colors (cf. fig. 2). On the x-axis, the different modes of the SAMUS system are lined up for a detailed comparison. Focusing on the results of mode 1, $92.74\,\%$ of the requested data rate can be granted for the BE slice, which corresponds with the high bandwidth utilization mentioned regarding fig. 7. This is due to the traditional scheduling request and grant mechanisms used in all slices including the mission-critical ones, exactly requesting the required data rate as needed, which means no resources are wasted. However, this also

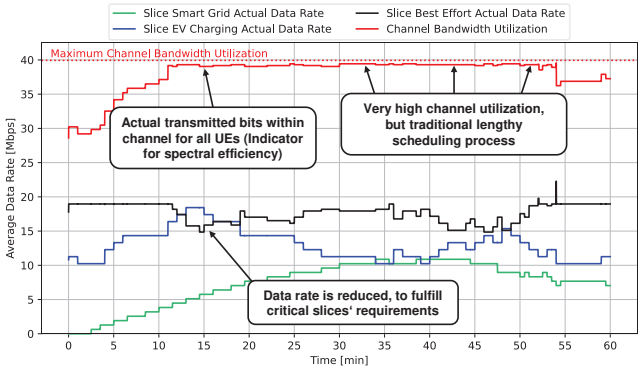
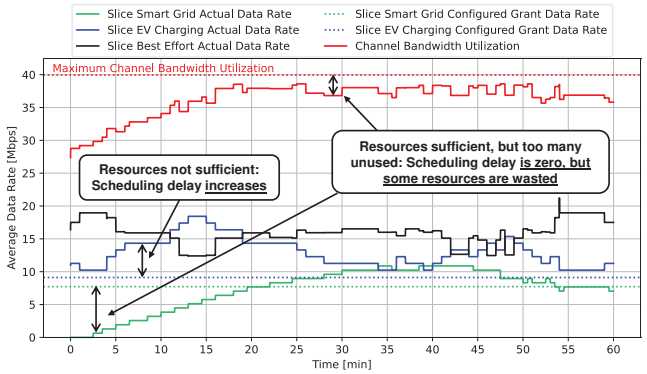
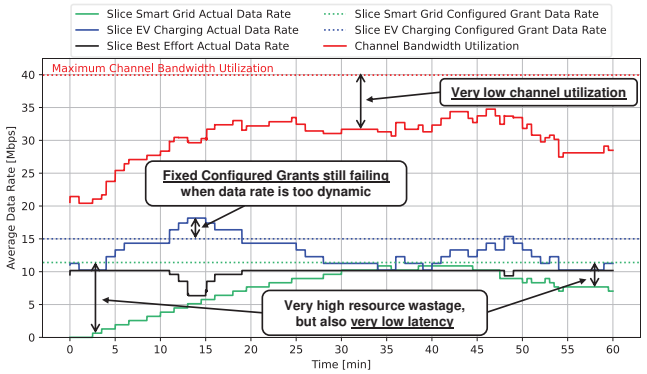


Fig. 7. The progression of data rates for the different network slices in mode 1 (5G parameters only)

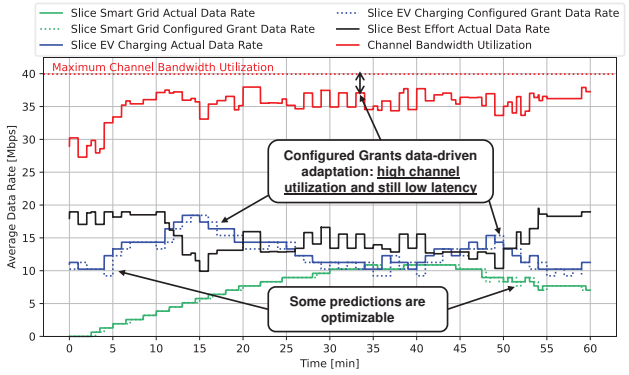
[1] https://www.nrel.gov/grid/solar-power-data.html
[2] https://new-poi.chargecloud.de/bonn (January 2020)



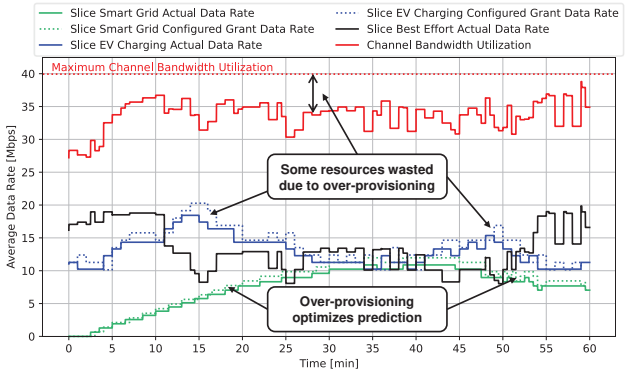
(a) Mode 2.1: Fixed Configured Grants with optimistic approach



(b) Mode 2.2: Fixed Configured Grants with pessimistic approach



(c) Mode 3.1: Predicted Configured Grants without Over-Provisioning



(d) Mode 3.2: Predicted Configured Grants with 10 % Over-Provisioning

Fig. 8. The progression of data rates for the different slices and modes 2.1-3.2 in comparison. The 60 min timeframe represents an interval with high volatility (EV charging: rush hour with many vehicles, SG: sunrise time with rapid change in solar activity)

means that this lengthy request and grant sequence leads to comparatively high (scheduling) latency for the uRLLC slices. Although the hardest 3rd Generation Partnership Project (3GPP) requirement of 5 ms for 5G [11] can be met, little to no margin for the other latency-inducing components (cf. fig. 2) is left (EV: 1.53 ms, SG: 2.47 ms).

In modes 2.1 and 2.2, fixed CGs with optimistic and pessimistic approach are utilized and are depicted in fig. 8a and 8b, respectively. In the optimistic variant, most of the packets transmitted in the mission-critical slices are still processed with traditional request and grant scheduling, as the fixed CGs are based on the historical average data rate observed in each mission-critical slice and thus, mostly lower than the required resources. As can be seen in fig. 9, this results in a comparatively high data rate utilization of 83.93 %. Also latency in both mission-critical slices is approximately halved in comparison to mode 1. Additionally, the EV charging and SG slice mean scheduling latency is now lower than 1 ms, leaving 4.04 ms and 4.58 ms margin for other end-to-end latency components, respectively. This is even further improved in mode 2.2, where the SG slice scheduling latency even drops down to zero, which enables the max. possible margin of 5 ms for the other latency-inducing components of the communication network. However, this comes with a cost: BE slice resource grant ratio drops down to 52.22 %, which means that almost half of the cell's resources are wasted.

In fig. 8c and 8d, the data-driven ARIMA-based approach is depicted, without and with 10 % over-provisioning, respectively. The dotted lines represent the predicted data rate for each slice by the SAMUS system in the respective colors. Considering that resource allocation is now more dynamic, higher scheduling latency occurs within the critical slices (cf. fig. 9) compared to mode 2.2. However, far better trade-offs can be made with this data-driven approach. Until now, mode 2.2 led to the best latency result, but with high resource wastage of approx. 50 %. In mode 3.1, nearly the same and even better latency margins of ∼5 ms can be achieved for both slices, while still providing 79.22 % of requested resources for the BE slice, which is comparable to mode 2.1. Finally, Over-provisioning of predicted resources leads to lower *mean* scheduling latency within the EV slice, trading 9 % of resources with 4.95 ms margin.

V. Conclusion and Outlook

In this work, we presented the scheduling prototype of the SAMUS system, evaluated on a specifically developed 5G-RGS framework. For this, a realistic Network Slicing scenario based on real-world data was utilized, where different

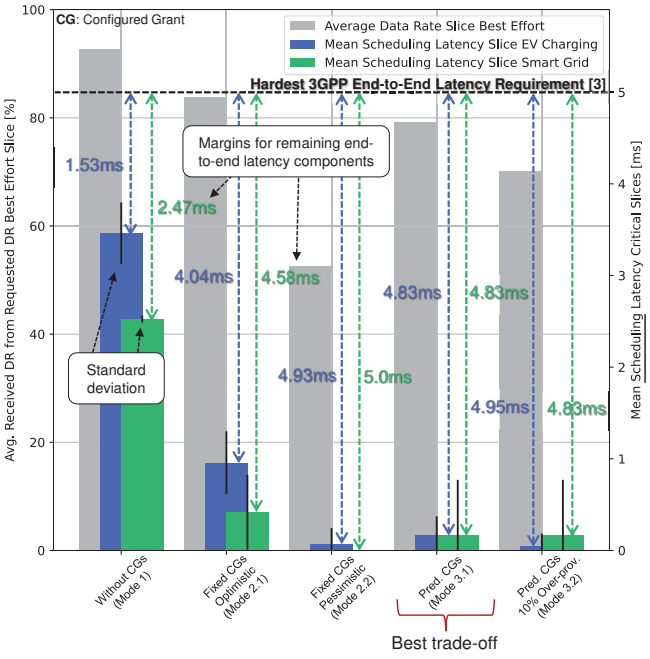


Fig. 9. Average best effort data rates versus mean and standard deviation of mission-critical slice latencies (averaging window: 2 s according to [11]) in comparison for different conducted modes. The arrows indicate margins for remaining end-to-end latency components (cf. fig 2).

modes of operation were analyzed. In conclusion, the data-driven ARIMA-based approach of SAMUS enables the best trade-off between high data rates in eMBB slices and low latency in mission-critical uRLLC slices, while still maintaining comparatively high channel bandwidth utilization. However, if the mission-critical slices are required to have the absolutely minimal latency, fixed Configured Grants (CGs) have to be utilized, accepting the high waste of resources coming along with it. This work gives an insight into different models of Network Slicing operation, where in public networks, the interests of end users and critical infrastructure safety have to be staked out against each other (so-called mixed-critical operation). In conclusion, Service Level Agreements (SLAs), which include traditional Key Performance Indicators (KPIs) such as packet rates and latency, need to incorporate the predictability of these KPIs as a new factor in 5G networks.

Extensive future work can be conducted based on the SAMUS system. For example, the SG slice latency shown in this work can be improved in future work by not solely relying on ARIMA. One approach could be to introduce fixed CGs at the start of the transmission, which is absolutely predictable as solar activity begins with pre-calculable sunrise times. As previously mentioned, the prediction of channel quality has to be included and evaluated. In this context, network planning methods incorporating the network slice planning could be developed, thus avoiding constraints imposed by low MCSs.

Acknowledgment

This work has been supported by the Ministry of Economic Affairs, Innovation, Digitalisation and Energy of the State of North Rhine-Westphalia (MWIDE NRW) along with the *Competence Center 5G.NRW* under grant number 005-01903-0047 and by the Federal Ministry for Economic Affairs and Energy (BMWi) in the course of the project *5Gain* under the funding reference 03EI6018C, as well as by the German Research Foundation (DFG) within the Collaborative Research Center SFB 876 "Providing Information by Resource-Constrained Analysis", project B4.

References

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

[2] C. Bektas, S. Monhof, F. Kurtz, and C. Wietfeld, "Towards 5G: An Empirical Evaluation of Software-Defined End-to-End Network Slicing," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.

[3] C. Bektas, S. Bocker, F. Kurtz, and C. Wietfeld, "Reliable Software-Defined RAN Network Slicing for Mission-Critical 5G Communication Networks," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[4] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3098–3130, May 2018.

[5] F. Laue, P. Karunakaran, and R. Schober, "Detection Schemes and Model Mismatch Analysis for 5G Configured-Grant Access for URLLC," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019.

[6] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, "Consideration On Automation of 5G Network Slicing with Machine Learning," in *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*, Nov. 2018, pp. 1–8.

[7] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning radio resource management in RANs: Framework, opportunities, and challenges," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 138–145, Sep. 2018.

[8] T. Bag, S. Garg, Z. Shaik, and A. Mitschele-Thiel, "Multi-Numerology Based Resource Allocation for Reducing Average Scheduling Latencies for 5G NR Wireless Networks," in *2019 European Conference on Networks and Communications (EuCNC)*, Jun. 2019, pp. 597–602.

[9] C. Arendt, S. Böcker, and C. Wietfeld, "Data-Driven Model-Predictive Communication for Resource-Efficient IoT Networks," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, Apr. 2020, pp. 1–6.

[10] 3GPP, "User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), TS 38.306, Release 15.11.0, Tech. Rep., Sep. 2020.

[11] ——, "System Architecture for the 5G System," 3rd Generation Partnership Project (3GPP), TS 23.501, Release 15.11.0, Tech. Rep., Sep. 2020.