Towards a Semantic-Aware Fine-Grained Link Adaptation for NextG Wireless Networks

†Gaurav Gautam, †Ajay Kumar Gurumadaiah, *Ahan Kak, *Nakjung Choi, †Zhi-Li Zhang
†University of Minnesota - Twin Cities, Minnesota, *Nokia & Bell Labs, USA
{gauta044, gurum021, zhang089}@umn.edu, {ahan.kak, nakjung.choi}@nokia-bell-labs.com

***Abstract*—We present a semantic-aware, fine-grained link adaptation framework for next-generation (NextG) wireless networks to improve end-to-end spectrum efficiency while maintaining assured application quality of experience (QoE). Emerging applications such as AR/VR/XR, edge-assisted autonomous driving and teleoperation of robotic systems produce disparate data streams with varied priority, latency and throughput requirements that have differing impacts on application QoE. Also, with the introduction of technologies like new Radio and mmWave, channels are becoming more diverse; more information is available in the RAN (Radio Access Network) about application data, user mobility, and channel characteristics. This presents an opportunity to use this information for finer-grained adaptation of the radio link. We design an architecture that bases link adaptation on application semantics, channel characteristics, and bandwidth availability. To achieve this objective, we introduce a software-defined, central slow-adapting module that reads semantics, user context, bandwidth availability, etc., for all users, and guides lower-level, fast link adaptation. We present results from our initial experimentation that show significant improvement in useful throughput.**

***Index Terms*—Link adaptation, MCS adaptation, Semantic-aware, NextG RAN, fine-grained adaptation.**

I. Introduction

Emerging applications such as teleoperated or edge-assisted autonomous driving, augmented/virtual/mixed or extended reality (AR/VR/XR) have diverse bandwidth, latency and reliability requirements [9]. Safety-critical applications such as autonomous vehicles (AVs) demand even more radio resources due to their low-latency and reliability requirements [1]. To support these bandwidth-hungry applications, 5G introduced new technologies like New Radio (NR), MIMO, carrier aggregation, and mmWave, along with a QoS framework designed to support services like eMBB and URLLC [4], [12]. However, the 5G flow-based QoS framework remains relatively rigid [16], despite the flexibility and AI-driven potential of O-RAN. Link adaptation is an important aspect that improves reliability for low-latency applications while also helping extract higher throughput. Link adaptation [13] adjusts estimated channel quality (CQI/MCS) values to improve spectral efficiency as well as meet application Quality of Service (QoS) requirements at same time.

First, we note that a higher BLER (Block Error Rate) does not necessarily affect application QoE when data is far from its deadline or has already passed its deadline. The current link adaptation applies to all application data the same way, regardless of whether the data has passed its deadline, is of lower priority, or the system bandwidth availability is low. Supporting data delivery with conservative channel estimates leads to higher radio resource usage, and overestimation can degrade application QoE when data is close to its deadline. Second, not all data is equally useful; for example, in a layered video streaming case, the base layer is more important than the enhancement layers. This different utility of data gives an opportunity to adapt link differently for different types of data. Motivated by these observations, we advocate for fine-grained link adaptation based on data semantics, such as deadlines and data priorities, and system states, such as bandwidth availability.

Current Open RAN (O-RAN) [6] solutions primarily focus on resource allocation and Modulation and Coding scheme (MCS) selection. Link adaptation is a fast process operating at sub-ms levels, whereas RAN Intelligent Controller (RIC) dapps/xapps operate at $\sim$10ms interval. As a result, O-RAN is not well-suited for directly supporting fine-grained, sub-ms link adaptation for multi-priority data. Moreover, sending detailed semantic information, such as deadlines and priorities for each data packet, to the RIC can create excessive signaling overhead. So, even to guide link adaptation from O-RAN, we need architecture support.

To overcome these limitations, we propose a link adaptation architecture that enables fine-grained link adaptation while retaining the potential to work hand-in-hand with O-RAN. Our contributions are as follows:

- We propose a novel architecture for semantic-aware, fine-grained link adaptation for nextG-RANs that incorporates application-level information into the adaptation process.
- We design a hierarchical link adaptation framework that operates at both the per-user level and the centralized level, leveraging data importance, bandwidth availability, and system context to guide adaptation decisions.

II. Background

In a 5G network, data arrives at the RAN with application-specific QoS requirements. These requirements are identified by QoS Flow Identifiers (QFIs) at the Service Data Adaptation Protocol (SDAP) layer. The data is then forwarded to different Data Radio Bearers (DRBs) based on QoS requirements. DRBs are logical channels established between the RAN and the User Equipment (UE). They carry user-plane data from the

ISBN 978-3-903176-82-9

Packet Data Convergence Protocol (PDCP) layer of the gNB to the UE, where QoS treatments are applied.

The Channel Quality Indicator (CQI) is reported by the UE to the RAN at intervals of tens to hundreds of milliseconds, depending on configuration. To transmit user data for specific QoS requirements, radio resources are allocated based on channel conditions and QoS requirements. The required radio resources are estimated using an adapted CQI based MCS selection. The Medium Access Control (MAC) layer uses Hybrid Automatic Repeat Request (HARQ) and retransmissions to ensure reliable delivery. Link adaptation primarily occurs in the gNB's MAC layer. Transport block ACKs/NACKs reported by the UE are used to adapt the CQI. For each cell/channel, link adaptation is performed separately. CQI adaptation is equivalent to MCS adaptation when there is a proportional mapping from CQI to MCS.

The current system uses CQI/MCS adaptation based on Outer Loop Link Adaptation (OLLA) [2]. OLLA is one of the most commonly used techniques. In this technique, an offset coefficient is adjusted based on a target BLER. This coefficient is continuously updated based on ACK/NACK feedback from transport blocks to maintain the target BLER. With every ACK, the coefficient is increased by a step size, say x, and with every NACK, decreased by $((1 - target\_bler) * x)/target\_bler$. A 10% target BLER is commonly used [4]. This coefficient is added to the reported CQI, and the resulting value, called the adapted CQI, is used for MCS selection. The technique helps increase link throughput by pushing MCS to higher values. It also helps maintain lower latency for safety-critical applications by reducing retransmissions, although its primary objective is BLER control rather than latency optimization.

Our measurements of a commercial 5G network in stationary and driving scenarios, shown in Fig. 1, suggest that the network operates around a target BLER of 10%. Compared with the stationary scenario, the driving scenario exhibits a higher overall BLER, occasionally reaching 100%, indicating that adaptation is neither fast nor effective enough to maintain a low error rate under rapid mobility.

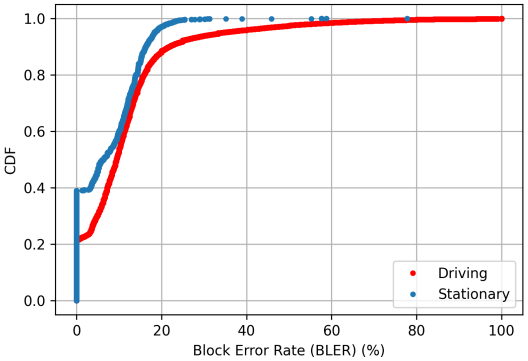


Fig. 1: CDF of measured Block Error Rate (BLER) under stationary and driving scenarios, showing higher error rates under mobility.

III. Motivation & Challenges

We motivate the problem using a hypothetical application with two flows: a higher-priority flow f_h and a lower-priority flow f_l. Examples of such applications include volumetric video streaming with a base layer and enhancement layers. The data to be sent is subject to deadlines d_l and d_h for flows f_l and f_h, respectively. We motivate the problem with the following three scenarios:

Deadline aware adaptation: Consider the cases when data is far from its deadline ($d_h > 100ms$) or when data has already passed its deadline ($d_h < 0ms$). The current system adapts the link in the same way, independent of the data deadline. If the same link adaptation is used, conservative channel estimates can lead to potential throughput loss. We note that high BLER does not necessarily affect the application's QoE when the data is far from its deadline or has already passed its deadline. Hence, overestimation of CQI/MCS provides an opportunity to achieve higher throughput without affecting QoE. By the nature of HARQ, previously transmitted data is not discarded; rather, it is combined with retransmissions to enhance decoding performance [3]. On the other hand, when the data is close to its deadline, a higher BLER can increase latency due to additional retransmissions, thereby increasing the likelihood of missing deadlines. State-of-the-art systems are unable to use these opportunities due to the basic QoS-aware link adaptation.

Per-flow adaptation: Our example application consists of two flows f_h and f_l. The traditional approach adapts both flows in the same way. Not all data is of equal utility; thus, treating all flows within an application equally leads to conservative CQI/MCS estimates across all flows and, therefore, a loss of potential throughput. Per-flow adaptation allows lower-priority data to be transmitted with a higher target BLER or using more throughput-oriented configurations, even when the flow is close to its deadline. Such adaptation can be particularly beneficial in unstable channels such as mmWave, where line-of-sight blockages may cause link adaptation to become overly conservative [10].

Central adaptation: The current system adapts the link at the user level; one user's data does not affect another user's link adaptation, and link adaptation is also not affected by the total available bandwidth. However, when system bandwidth availability is low, lower-priority flows (e.g., f_l) can be configured for more throughput-oriented link adaptation. This reduces the bandwidth requirements of the corresponding user, thereby freeing resources for other users and improving their QoE. To achieve this objective, we propose a centralized adaptation design that bases link adaptation on data importance and bandwidth availability.

IV. Overview

In the proposed work, we leverage application semantics, channel characteristics, and bandwidth availability to improve link adaptation, thereby improving overall QoE. Traditional approaches focus on using QoS requirements like latency, data rate, etc for link adaptation, treating all data uniformly

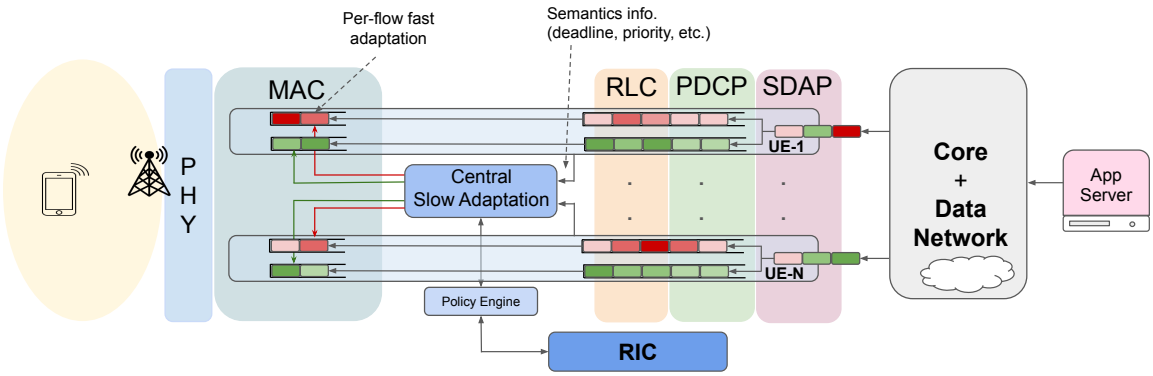


Fig. 2: Semantic-Aware Fine-Grained Link Adaptation Architecture

during link adaptation. First, we identify that application QoE is not always affected by retransmissions and high BLER; knowing the semantics, such as the data deadline, enables more informed MCS selection. Second, not all data is of equal utility; for example, lower-priority flows of an application or data that has passed its deadline are less useful. This gives an opportunity to adapt different data in different ways. For instance, a lower-priority application flow can focus on opportunistic link adaptation with a higher target BLER, thereby improving resource efficiency.

V. Design

A. Architecture

To use semantics for link adaptation, the first step is to identify data semantics, such as data priority, deadlines, QoS requirements, etc. For our case, we assume that there are Service Level Agreements (SLAs) between the network operator and the Application Service Provider (ASP). Data arriving at the RAN with semantic information attached by the ASP is processed by the SDAP layer. Multiple DRBs are created for a user. Different application flows are forwarded to different DRBs based on priority. This allows us to create an architecture capable of per-flow link adaptation. The deadline and QoS information are extracted and passed towards the MAC layer by attaching metadata to the data itself. These deadlines and QoS requirements are read by the Central slow adaptation module.

B. Central Slow Adaptation

We introduce a central slow adaptation module running at a slower speed $\sim 10ms$ sitting between the RLC and the MAC layer. This module checks the quality of service requirements, data priority, buffer occupancy, and deadlines of all application flows for all users from the RLC side. From the MAC side, it receives channel information (CQI, center frequency, etc) and current adaptation information. It has policies that determine the appropriate adaptation for each data flow based on the reported data and system state. RIC can be used to upload these policies or to select the adaptation parameters directly. For all DRBs/application flows, this module passes decisions such as target BLER, step size, and deadline budget to the MAC scheduler for fast adaptation during its execution period.

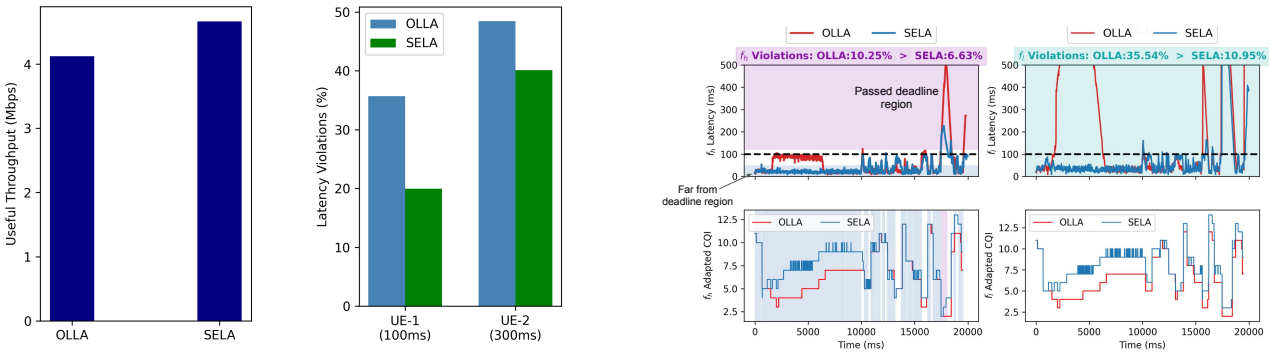
C. Fast dynamic link adaptation per-flow

In the MAC layer, actual fast (sub-ms) adaptation occurs, guided by the slow adaptation module. In our design, each data flow queue/DRB is associated with its own link adaptation. For example, when all app flows use separate OLLA, offset coefficients for all application data flows are tracked and updated at every ACK/NACK. During scheduling, the adapted CQI is calculated based on the data transmitted in the current TTI, which directly determines the MCS used. If data from multiple priorities is transmitted in the current TTI, the least MCS among priorities is used. This enables fine-grained link adaptation. These link adaptation parameters, such as target BLER, step size, and other parameters, dynamically change based on the deadline, channel characteristics, flow priority, and bandwidth availability. For example, when data is far from its deadline or has already passed its deadline, we can use a higher target BLER, say 20%, and when data approaches its deadline, the target BLER can be reduced to a lower value, say 5%. In our example, we used OLLA, but other adaptation schemes, such as Deep Reinforcement Learning based, can be used for different flows at different times.

O-RAN Integration. The slow adaptation module can interface with the near-real-time RIC via the E2 interface. If RIC operation is enabled, the module forwards the minimum required semantic and channel information to the RIC. RIC can provide either policies or per-flow link adaptation decisions.

VI. Implementation and Experimental Setup

We implemented the proposed solution in srsRAN [11] only for the downlink. Data flow priorities are identified using packet destination port numbers, and deadlines attached to packets are read at the SDAP layer. This information is passed to the MAC layer via metadata. The central slow adaptation module is implemented as a separate component that checks the deadline of the first packet in each user's RLC buffer every 10 ms. It calculates the step size and target BLER for each application flow using predefined policies and reports them to the MAC layer, where fast adaptation occurs. At the MAC



(a) Average Total Useful Throughput (End-to-End).

(b) End to End latency Violations across UEs.

(c) Latency and adapted CQI for the high-priority flow f_h and low-priority flow f_l of the application with target latency of $100ms$.

Fig. 3: Evaluation.

layer, we track a coefficient for each DRB/flow, which is updated with every ACK/NACK. Each HARQ process stores the priority of its transmitted data, enabling incoming ACK/NACK feedback to be mapped to the correct DRB/flow. When the MCS is calculated, the corresponding flow coefficient is used to calculate the adapted CQI.

Experiment Setup. We used a setup comprising two UEs, a RAN, and a core network. We used B210 software-defined radios (SDRs) as radio frontends for Over-The-Air (OTA) experiments. For some experiments, we used emulated ZMQ channels with the GNU Radio broker. The channel was configured in FDD mode with a 5 MHz bandwidth. To compare the baseline side-by-side with our proposed solution, we divided the downlink bandwidth into two equal parts. This was done by running the baseline on odd TTIs and our solution on even TTIs. The CQI reporting interval was set to 100ms. We used a multi-flow deadline-aware resource block allocation algorithm. To synchronize the nodes, we used a local NTP server.

Application: We used an application with two flows f_h and f_l, both with the same latency requirement. The first flow is more important than the second one.

VII. Evaluation.

In this section, we first evaluate end-to-end performance and then deep dive to see why the proposed solution performs better. For a fair comparison, we run the baseline on odd TTIs and our solution on even TTIs, each with its own resource allocation, ensuring equal access to resource blocks under identical channel conditions.

Baseline: We use OLLA with 0.1 (10%) target BLER and 0.001 step size as the baseline. All flows are adapted with the same adaptation in this case.

SELA (Semantic Enabled Link Adaptation) Policy: We use this example policy where when $deadline > 50$ ms and $deadline < -20$ ms then $target\_bler = 0.2$ and $stepsize = 0.01$; For lower priority flow f_l, $target\_bler = 0.2$ and $stepsize = 0.01$ are used at all times; otherwise $target\_bler = 0.1$ and $stepsize = 0.001$.

The policy uses a heuristic that a higher BLER does not affect application QoE when data is far from its deadline or has already passed its deadline. This is a heuristic-based example policy for the given applications and may not generalize to all possible applications. Complex policies need to be learned for generalization.

A. End to End

In the end-to-end evaluation, two UEs are connected to the RAN. The first UE runs an application with a 100ms target latency, while the second UE runs an application with a 300ms latency requirement. Several experiments were conducted using the OTA testbed and trace-driven emulated channels. The OTA experiments cover interesting scenarios in which a resource bottleneck occurs under stationary, motion, and fast-motion conditions. We measure overall system useful throughput, defined as the throughput that does not violate deadlines, as shown in Fig 3a. The metric is a good indicator of improvement in overall system performance. For clarity, we highlight representative cases where performance differs from the baseline; runs with similar performance are omitted because they provide limited additional insight. As shown, there is a significant increase in useful throughput compared to OLLA. The actual throughput gains are lower than the useful throughput gains. This shows that, even under similar overall throughput, performance can be improved with our technique. The primary reason for these gains is that our system can leverage opportunities enabled by knowledge of data semantics, allowing the target BLER and step size to be set to higher values to extract more throughput.

For the same experiments, we measure the overall latency violations for both UEs, as shown in Fig 3b. The results show a reduction in latency violations for both UEs. The lower latency violations for both UEs also indicate that they can extract more useful throughput from the wireless link compared to OLLA. The primary gains are observed in moving and fast-moving scenarios; in stationary scenarios, adaptation is rapid at the start of the experiment, but after initial adaptation, performance is similar. In other cases, greater gains are observed in regions where the system pushes the target BLER to higher values, thereby enabling higher MCS.

B. DeepDive

To demonstrate why our system performs well, we conduct an experiment with a single UE connected to the RAN. For this experiment, we use a trace-based emulated channel. The application has two flows, each with a target latency of 100 ms. We plot latency and adapted CQI to compare the baseline with the proposed solution for both flows; results are shown in Fig 3c. MCS selection can be inferred from the adapted CQI, as the adapted CQI is proportionally mapped to the MCS.

For the higher-priority flow f_h. First, we focus on the region where the data is far from its deadline. We note that in this region, adaptation occurs quickly with a high target BLER, resulting in a higher adapted CQI whenever possible. This higher adapted CQI leads to higher MCS, which extracts more throughput from the link. The impact is a reduction in latency for the lower-priority flow f_l as more bandwidth becomes available for it. Second, the regions where the data has already passed its deadline. Higher latency spikes show these regions. In these regions, the CQI adapts more quickly than OLLA. As a result, higher throughput is achieved, and latency is reduced.

For the lower-priority flow f_l, the target BLER is always 20%, and the adaptation is faster with a step size of 0.01. As a result, the adapted CQI is mostly higher than in OLLA. The resulting effect can be seen in the reduction in latency spikes. Moreover, there are some regions where data is far from the deadline; in these regions, resource blocks are minimized, since resource blocks are not shown here, those gains are not explicitly visible. We note that in some instances, our solution latency exceeds the baseline; however, these are rare, and latency is mostly below the deadline.

We also conducted the same experiment with OLLA using a step size of 0.01 and 10% target BLER. In that case, our solution also performed better, although we do not show the results here. OLLA performed similarly to our solution in the first half of the experiment but worse in the second half, where the channel is unstable.

VIII. Related Work

Link adaptation has been widely studied in the literature. OLLA [2] is a widely used technique in commercial systems and implemented in srsRAN [11] as well. SALAD [13] infers inaccuracies in SINR estimation based on past MCS selections and HARQ feedback (ACK/NACK), and adjusts the MCS to compensate for overestimation or underestimation of channel conditions. More recently, deep reinforcement learning (DRL)-based approaches have been proposed. DRLLA [14] employs deep reinforcement learning to maximize throughput while maintaining the BLER below a predefined target in LTE/NR systems. QDRLLA [7] is a QoS-aware link adaptation framework based on deep reinforcement learning that jointly considers transmission power and OFDM numerologies, enabling multi-domain adaptation. Several other works also explore DRL-based MCS selection under various system models and assumptions [5], [8], [15]. While prior work primarily focuses on channel conditions and QoS parameters, QoS-aware link adaptation remains limited to coarse-grained cross-layer optimization. In contrast, our work considers data semantics, such as deadlines, priorities, and bandwidth availability, to enable fine-grained adaptation at the level of individual data flows.

IX. Discussion and Future Work

Our evaluation focused on comparing our solution against OLLA with 10% target BLER. However, for safety-critical applications, the operator may choose a lower target BLER (e.g., 2%). In such cases, our proposed approach is more useful, as it dynamically targets high-priority data with a lower BLER while allowing higher BLER for lower-priority data. During our experiments, we observed that setting a very high target BLER (>30%) can lead to inefficient bandwidth utilization. This is because higher BLER increases the number of retransmissions, and once the maximum number of HARQ retransmissions is reached, the MAC layer discards the data, resulting in wasted transmission resources. One way to mitigate this issue is to increase the maximum number of HARQ retransmission attempts; however, this may not always be feasible due to implementation constraints and memory limitations. Our experiments were conducted on a small two-UE testbed; however, we expect the qualitative benefits to generalize to other applications and larger deployments. When a user conserves radio resources by operating at a higher MCS (i.e., a higher target BLER), those saved resources can be reallocated to other users, potentially improving overall system performance. The magnitude of this gain, however, depends on the scheduling and adaptation policies employed.

The policy adopted in this paper is based on the heuristic that when data is farther from its deadline, or has already passed its deadline, QoE is not significantly impacted by operating at a higher target BLER, as long as doing so improves link throughput. A deadline margin of at least 20 ms provides sufficient time for multiple retransmissions and link adaptation adjustments. Similarly, for lower-priority data, a slightly higher BLER is unlikely to significantly affect user QoE. Even simple policies can be effective in this framework, while more sophisticated policies can be learned based on operator objectives, application requirements, and user QoE feedback. For example, when the objective is to conserve radio resources and improve energy efficiency, lower-priority flows may be transmitted using more aggressive MCS settings, even when bandwidth is available, to avoid over-provisioning reliability. In contrast, when the objective is strict deadline compliance, particularly for high-priority traffic, the system can adopt more conservative MCS settings as data approaches its deadline to improve delivery reliability. Further research is needed to learn optimal policies.

Deployment. We acknowledge that deployment of such a solution is challenging, as semantic information and fine-grained application information are not available in the current 5G RAN deployments. However, next-generation RAN is moving towards a fine-grained approach, as evidenced by advances in the 5G QoS framework. As discussed earlier, our solution can be deployed under SLAs or within a private 5G network. Data semantics can be conveyed to the RAN via packet headers.

An application service provider can use AI to classify data and annotate packets with semantic information. The current 5G QoS framework supports the establishment of multiple DRBs, allowing distinct application flows to be mapped onto separate bearers for per-flow adaptation. Centralized adaptation is also feasible, as the central slow adaptation module can monitor bandwidth availability and access flow-level QoS requirements. However, application-level deadlines are not explicitly exposed to the RAN; although packet delay can be measured, it does not directly reflect application deadline constraints. Incremental deployment is possible, as the system can handle some users with basic adaptation and others with semantic-aware adaptation.

Future Work. There are multiple future directions. First, a more rigorous evaluation of the proposed link adaptation solution is needed. Second, this work can be integrated with RIC using the O-RAN framework, where the E2 interface can serve as the communication channel between the central adaptation module and the near-RT RIC. That further enables the development of AI/ML-based policies that maximize overall QoE. Third, this work focuses on downlink adaptation. Extending the proposed framework to the uplink presents additional challenges, particularly in conveying semantic information from the UE to the RAN, and is therefore left for future work. Another potential direction is to optimally propagate ACK/NACK feedback across flow offset coefficients. We observed that adaptation was slower for flows transmitted less frequently, since they receive fewer ACK/NACK updates. In our implementation, we addressed this by initializing the offset coefficient for such flows using the coefficient of another frequently transmitted flow to improve the estimate. For example, when a lower-priority flow with a 20% target BLER is transmitted for the first time, we initialize its offset coefficient using that of the frequently transmitted higher-priority flow with a 10% target BLER, rather than starting from zero.

X. Conclusions

We presented a semantic-aware, fine-grained link-adaptation architecture that uses data deadlines, priority, and bandwidth availability to adapt the link. Through OTA and trace-based evaluation, we showed that a simple heuristic policy can significantly outperform the baseline in terms of useful throughput and latency violations.

Acknowledgment

We thank anonymous reviewers for their feedback. The research was supported in part by NSF awards CNS-2220286, CNS-2220292, CNS-2321531, CNS-2323174, DMS-2436333, and ITE-2453815.

References

[1] Mehdi Bennis, Mérouane Debbah, and H. Vincent Poor. Ultrareliable and low-latency wireless communication: Tail, risk, and scale. *Proceedings of the IEEE*, 106(10):1834–1853, 2018.

[2] Francisco Blanquez-Casado, Gerardo Gomez, Maria del Carmen Aguayo-Torres, and Jose Tomas Entrambasaguas. eolla: an enhanced outer loop link adaptation for cellular networks. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):20, 2016.

[3] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[4] Erik Dahlman, Stefan Parkvall, and Johan Skold. *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[5] Yan Huang, Y. Thomas Hou, and Wenjing Lou. Deluxe: A dl-based link adaptation for urllc/embb multiplexing in 5g nr. *IEEE Journal on Selected Areas in Communications*, 40(1):143–162, 2022.

[6] O-RAN Alliance. O-ran: Towards an open and smart ran. https://www.o-ran.org, 2018. Accessed: 2026-02-28.

[7] Ali Parsa, Neda Moghim, and Sachin Shetty. Qos-aware link adaptation for beyond 5g networks: A deep reinforcement learning approach. *IEEE Open Journal of the Communications Society*, 6:6368–6382, 2025.

[8] Vidit Saxena, Hugo Tullberg, and Joakim Jaldén. Reinforcement learning for efficient and tuning-free link adaptation. *IEEE Transactions on Wireless Communications*, 21(2):768–780, 2022.

[9] M Series. Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond. *Recommendation ITU*, 2083(0):1–21, 2015.

[10] Christopher Slezak, Vasilii Semkin, Sergey Andreev, Yevgeni Koucheryavy, and Sundeep Rangan. Empirical effects of dynamic human-body blockage in 60 ghz communications. *IEEE Communications Magazine*, 56(12):60–66, 2018.

[11] srsRAN Project. srsRAN, 2025.

[12] Jeanette Wannström and 3GPP. Carrier aggregation explained. Web page, December 2022. Last updated December 12, 2022.

[13] Reinhard Wiesmayr, Lorenzo Maggi, Sebastian Cammerer, Jakob Hoydis, Fayçal Aït Aoudia, and Alexander Keller. Salad: Self-adaptive link adaptation. *arXiv preprint arXiv:2510.05784*, 2025.

[14] Xiaowen Ye, Yiding Yu, and Liqun Fu. Deep reinforcement learning based link adaptation technique for lte/nr systems. *IEEE Transactions on Vehicular Technology*, 72(6):7364–7379, 2023.

[15] Lin Zhang, Junjie Tan, Ying-Chang Liang, Gang Feng, and Dusit Niyato. Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks. *IEEE Transactions on Wireless Communications*, 18(6):3281–3294, 2019.

[16] Zhi-Li Zhang, Udhaya Kumar Dayalan, Eman Ramadan, and Timothy J. Salo. Towards a software-defined, fine-grained qos framework for 5g and beyond networks. In *Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration*, page 7–13, New York, NY, USA, 2021. Association for Computing Machinery.