Research on Real-Time and Robust MU-MIMO Scheduling under O-RAN Architecture

Jaemin Kim, Dongwook Won, Donghyun Lee, Junsuk Oh, Chihyun Song,
Seungchan Lee, Juyoung Kim, and Sungrae Cho
School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea
Email: {jmkim, dwwon, dhlee, jsoh, chsong, sclee, jykim}@uclab.re.kr, srcho@cau.ac.kr

***Abstract*—The O-RAN architecture's disaggregation enables flexible PHY/MAC control via standardized non-RT and near-RT loops, but imposes a fundamental tension between the sub-millisecond latency requirements of the real-time (RT) control and the data-driven intelligence (e.g., for robust scheduling) envisioned for the near-real-time (near-RT) loop. This paper synthesizes two critical, complementary design philosophies—a compute-aware GPU-centric design and an uncertainty-aware chance-constrained design—contrasting their architectural choices and problem decompositions. We unify notation, state wall-clock constraints explicitly, and connect these systems to xApp-based near-RT control and robust MU-MIMO literature to extract actionable design lessons for 5G/6G O-RAN deployments.**

***Index Terms*—O-RAN, near-RT control, RT scheduling, MU-MIMO, RB allocation, beamforming/precoding, GPU acceleration, robust scheduling, imperfect CSI, QoS guarantees, cell-free massive MIMO**

I. Introduction

Next-generation wireless networks (5G and beyond) face unprecedented demands for high data throughput, ultra-low latency, and massive connectivity. Multi-user multiple-input multiple-output (MU-MIMO) is a cornerstone technology to meet these spectral efficiency demands by enabling spatial multiplexing. Concurrently, the Open RAN (O-RAN) architecture has emerged as a dominant paradigm, promising to disaggregate RAN components and introduce AI-driven intelligence [1]. However, this very disaggregation, while enabling flexibility and multi-vendor interoperability, introduces significant challenges for the real-time (RT) physical (PHY) and MAC layer control loops. Specifically, O-RAN's architecture enables complex multi-cell MU-MIMO scheduling but forces a hard wall-clock constraint within the RT pipeline [2], [3]. Let T_{TTI} denote the transmission time interval (ms-scale in NR numerology) and let T_{RT} denote the end-to-end latency of the RT scheduling pipeline at the DU (including device–host transfers, kernel launches, and synchronizations). Time to interactive (TTI) is on the order of sub-ms to 1 ms, depending on numerology; we use 1ms as a representative budget. We use standard terms: resource block (RB), modulation and coding scheme (MCS), channel state information (CSI), and joint transmission (JT) where a UE is served by multiple O-RUs. Under a common single-grant-per-UE abstraction, MU-MIMO scheduling enforces per-UE uniformity across the RBs assigned in a TTI, using one MCS and one stream count per UE.

To compare representative systems in a common template, let $\mathcal{U}$ be the UE set, $\mathcal{B}$ the RB set, and $\mathcal{R}$ the O-RU set. In each TTI, the scheduler selects serving O-RUs $\mathcal{R}_u \subseteq \mathcal{R}$ for each $u \in \mathcal{U}$ (single-cell or JT), an RB assignment $x_{u,b} \in \{0,1\}$, a per-UE stream count $s_u \in \{0,1,\dots,s_{\max}\}$, a per-UE MCS m_u, and a per-RB precoder $\mathbf{W}_b$ (block-diagonal across the O-RUs active on RB b). With weights $w_u \geq 0$, often derived from higher-layer QoS/QoE requirements (e.g., video streaming metrics [4], [5]) or set by a near-RT xApp to manage SLAs, a generic objective is

$$\max_{\mathcal{R}_u, x_{u,b}, s_u, m_u, \mathbf{W}_b} \sum_{u\in\mathcal{U}} w_u \sum_{b\in\mathcal{B}} x_{u,b}\, R(\mathrm{SINR}_{u,b}(\mathbf{W}_b; \mathcal{R}_u), m_u)\,. \tag{1}$$

Here $\mathrm{SINR}_{u,b}(\cdot)$ is computed from (possibly imperfect) CSI on RB b given the user group with u and the precoder $\mathbf{W}_b$. The per-RB spatial multiplexing budget imposes

$$\sum_{u\in\mathcal{U}} x_{u,b}\, s_u \leq S_{\mathrm{tx}}(b), \quad \forall b \in \mathcal{B}, \tag{2}$$

and per-UE uniformity is expressed by

$$m_u \text{ is constant over } \{b : x_{u,b} = 1\}, \tag{3}$$

$$s_u \text{ is constant over } \{b : x_{u,b} = 1\}. \tag{4}$$

With imperfect or fronthaul-limited CSI, a per-UE probabilistic QoS can be written as

$$\mathbb{P}[\mathrm{SINR}_{u,b} \geq \gamma_u,\ \forall b :\ x_{u,b} = 1] \geq 1 - \varepsilon_u, \tag{5}$$

where γ_u is the SINR target and $\varepsilon_u \in (0,1)$ is the violation budget. Here, ϵ_u controls the risk level, and the exact probabilistic model can vary depending on the CSI uncertainty representation.

To navigate this design space, we structure this survey as follows. We first review two representative systems: **(i) a compute-aware, GPU-first design** that prioritizes meeting the $T_{\mathrm{RT}} \leq T_{\mathrm{TTI}}$ constraint even for complex multi-cell JT [6], and **(ii) an uncertainty-aware, chance-constrained design** that prioritizes robust QoS guarantees via a novel near-RT/RT split [7]. We then synthesize these findings, summarized in Table I, and provide context from related O-RAN xApp [8]–[10] and cell-free studies [3] to extract actionable design lessons for future 5G/6G O-RAN schedulers.

II. Literature Review

A. Real-Time Performance and Compute-Aware Design

This line of research focuses on ensuring the sub-millisecond latency required by the O-RAN RT pipeline, often by leveraging parallel computing resources. A prime example is the compute-aware, GPU-first multi-cell MU-MIMO scheduler [6]. This approach aggressively exploits GPU parallelism to keep T_{RT} within 1 ms, even when implementing complex Joint Transmission (JT) for edge users. It achieves this by expressing core functions—per-O-RU/RB scoring, candidate grouping, and small-shape linear algebra (ZF/MMSE)—as batched kernels with minimal host intervention. These hardware-centric designs underscore the necessity of scheduler–PHY co-design at scale to maintain T_{TTI} feasibility [2], [11].

TABLE I: Two representative O-RAN MU-MIMO schedulers aligned to a unified template

Axis	GPU-first multi-cell (JT-capable)	Uncertainty-first (near-RT/RT split)
Primary aim	Sub-ms RT latency with JT gains [6]	Probabilistic QoS with few-CSI samples [7]
Decision split	Single RT pipeline (two UE classes)	RB/MCS in near-RT; beams/groups in RT
CSI treatment	Measured CQI/correlation; batched LA	SAA + ambiguity sets (distributional robustness)
RT feasibility	Kernel independence; minimal sync	Reduced/prioritized RT search width [7]
Time-scale support	Implicit (all-in-RT pipeline)	xApp-driven warm starts [8], [9]

B. Robustness and Near-RT Intelligence via xApps

To address the uncertainty inherent in RAN operation (e.g., delayed or imperfect CSI), a second major research line focuses on robustness and policy control , primarily leveraging the near-RT RIC and its xApps.

1) Uncertainty-Aware, Chance-Constrained MU-MIMO: This approach treats scheduling as a decision-making process under explicit CSI uncertainty with probabilistic QoS guarantees. The uncertainty-aware, chance-constrained scheduler [7] splits the problem: coarse decisions like RB/MCS assignment are solved in the near-RT loop, and the highly time-critical beamforming and instantaneous grouping are finalized in the RT pipeline within a bounded search space. This leverages Sample-Average Approximations (SAA) and ambiguity sets to tune the trade-off between violation risk (ε_u) and RT complexity. Robust MU-MIMO literature with limited feedback or PMI selection in O-RAN-like scenarios highlights adjacent mechanisms for reducing search space and handling uncertainty [12], [13].

2) Near-RT Control and xApp Implementation: Independent of the RT kernel design, the near-RT RIC provides the platform for intelligent policy control. Near-RT xApps can tailor scheduler parameters at runtime to manage trade-offs like throughput, latency, and reliability [8]. AI/ML techniques, such as Deep Reinforcement Learning (DRL), are commonly explored for intelligent control and resource optimization in these contexts. Furthermore, the ability of xApps to coordinate resources, for instance via learning-based cooperative control [14], highlights their crucial role in achieving system-wide efficiency and robustness. The O-RAN RIC itself has been subjected to intensive research regarding its open-source implementation and real-world validation [15], demonstrating the practical feasibility of xApp-based intelligent control. Studies on SLA/QoS-driven packet schedulers in O-RAN validate this policy-level control consistent with timing and interfaces [9]. Practical evaluation frameworks using NS-3/OAI offer vehicles for pre-ranking candidates, warm-starting the RT pipeline, and validating scheduler policies against KPIs [10]. The concept of distributed adaptive communication is also relevant, providing a framework for robustly managing resources in the decentralized O-RAN structure.

C. Architectural Context and Cross-Layer Interplay

Finally, the architectural and service contexts provide boundaries for O-RAN scheduling design. Cell-free Massive MIMO is highly relevant as O-RAN's disaggregated architecture is an ideal fit for its cooperative, multi-AP operation [3]. Studies in this area inform how multi-AP scheduler designs must align with O-RAN's standardized interfaces and timing constraints. Furthermore, the physical layer infrastructure connecting RUs and DUs is continuously evolving, with Optical Wireless Communications (OWC) emerging as a key technology for next-generation RANs, offering high-throughput connectivity essential for centralized, low-latency control [16]. Separately, cross-layer QoS/QoE context provides canonical formulations where resource allocation is tied directly to end-user metrics [4], [5]. These metrics drive how objective weights (w_u) in the scheduling problem (2) are determined. For instance, dynamic resource allocation and scheduling are essential for services like dynamic streaming [17] or managing service differentiation in prioritized environments. The importance of the near-RT loop setting these w_u values, driven by service-layer demands, bridges the gap between application requirements and the PHY/MAC decisions in the RT pipeline.

D. Distilled Design Lessons

(i) **Exploit time-scale separation**: Use near-RT xApps not only for pre-ranking candidates [8], [9] but also for dynamically setting RT-loop policies, such as the objective weights (w_u) or risk budgets (ε_u) based on end-to-end QoS/QoE metrics [4], [5]. Advanced AI techniques like DRL and Multi-Agent RL are crucial for this policy setting [14]. (ii) **Formulate for GPUs**: Prefer batched, fixed-shape primitives and shallow iterations to ensure $T_{\mathrm{RT}} \leq 1$ ms, even if sacrificing theoretical optimality [6]. (iii) **Treat CSI as a budget**: Expose explicit violation parameters (ε_u) and bound RT search complexity, while leveraging limited-feedback or PMI mechanisms when helpful [7], [12], [13]. (iv) **Align with**

O-RAN interfaces: When expanding to cooperative/multi-AP regimes (like Cell-Free MIMO), align designs with O-RAN interface/timing constraints demonstrated in practice [3], [10]. Leveraging distributed control mechanisms can enhance system robustness in this decentralized structure.

III. Conclusion

O-RAN's layered control creates both an opportunity and a constraint for MU-MIMO scheduling: near-RT loops can precompute structure and policies, while RT pipelines must finalize decisions within a millisecond budget under imperfect CSI. Two representative lines illustrate complementary answers. A compute-aware, GPU-first pipeline achieves sub-ms scheduling and unlocks multi-cell JT gains by saturating parallelism and minimizing synchronization [6]. An uncertainty-aware, chance-constrained design enforces probabilistic QoS with few CSI samples by splitting RB/MCS and beamforming across near-RT/RT and by capping RT search complexity [7]. Surrounding these cores, xApps provide runtime handles to reconcile throughput, latency, and reliability, and evaluation studies ground the design choices in measurable KPIs [8]–[10].

Looking forward, several recommendations emerge for deployments and future research:

1) Make information constraints first-class. Expose per-UE violation budgets and CSI sampling windows as tunable inputs at near-RT; reflect them in RT candidate widths so that $T_{\mathrm{RT}} \leq T_{\mathrm{TTI}}$ holds by construction [7].
2) Treat the scheduler as a compute graph. Favor batched, fixed-shape kernels and data residency; measure end-to-end wall-clock including transfers and synchronization, not just kernel times [6].
3) Use xApps to prepare low-entropy RT surfaces. Pre-rank UE groups, suggest MCS/stream priors, and adjust risk/weight parameters online to track traffic, mobility, and fronthaul variability [8], [9].
4) Leverage adjacent mechanisms to shrink RT search. Integrate limited-feedback user/beam selection and interference-aware PMI strategies where appropriate, while staying within O-RAN interface and timing constraints [3], [12], [13].
5) Report reproducible metrics. Publish per-TTI wall-clock distributions under stated hardware and loads, plus QoS violation rates under the chosen risk budgets; align KPIs with evaluation frameworks used in the O-RAN community [10].

Ultimately, unifying compute- and uncertainty-awareness—xApp-prepared candidates and risk budgets feeding GPU-optimized RT kernels—offers a pragmatic path to robust MU-MIMO gains in O-RAN deployments, with room to extend toward multi-AP/cell-free operation as interfaces and fronthaul mature [3].

Acknowledgment

This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation) - ITRC (Information Technology Research Center) (IITP-2026-RS-2022-00156353, 50%) grant and YKCS Open RAN Global Collaboration Center (IITP-2026-RS-2024-00434743) grant funded by the Korea government (Ministry of Science and ICT).

References

[1] A. Elyasi, A. Ashdown, K. Rumman, and F. Restuccia, "O-ran xapps: Survey and research challenges," *Available at SSRN 5237491*, 2025.
[2] S. Hassouna, J. Kaur, B. Kizilkaya, J. u. R. Kazim, S. Ansari, A. A. Kherani, B. Lall, Q. H. Abbasi, and M. Imran, "Development of open radio access networks (o-ran) for real-time robotic teleoperation," *Communications Engineering*, vol. 4, no. 1, p. 176, 2025.
[3] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen, and E. Vinogradov, "Cell-free mmimo support in the o-ran architecture: A phy layer perspective for 5g and beyond networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 28–34, 2022.
[4] J. Kim, G. Caire, and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2319–2331, 2015.
[5] M. Choi, J. Kim, and J. Moon, "Wireless video caching and dynamic streaming under differentiated quality requirements," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1245–1257, 2018.
[6] Y. Chen, Y. T. Hou, W. Lou, J. H. Reed, and S. Kompella, "Om 3: Real-time multi-cell mimo scheduling in 5g o-ran," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 339–355, 2023.
[7] Y. Wu, Y. Shi, Y. T. Hou, W. Lou, J. H. Reed, and L. A. DaSilva, "R 3: A real-time robust mu-mimo scheduler for o-ran," *IEEE Transactions on Wireless Communications*, 2024.
[8] N. Longhi, S. D'Oro, L. Bonati, M. Polese, R. Verdone, and T. Melodia, "Tailo-ran: O-ran control on scheduler parameters to tailor ran performance," *arXiv preprint arXiv:2508.12112*, 2025.
[9] W. Zhang, B. Vucetic, and W. Hardjawana, "5g real-time qos-driven packet scheduler for o-ran," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*. IEEE, 2024, pp. 1–6.
[10] A. Subudhi, A. Piccioni, V. Gudepu, A. Marotta, F. Graziosi, R. Hegde, and K. Kondepu, "Performance evaluation of scheduling scheme in o-ran 5g network using ns-3," in *2024 IEEE Future Networks World Forum (FNWF)*. IEEE, 2024, pp. 590–595.
[11] S. S. R. Jonnavithula, I. K. Jain, and D. Bharadia, "Mimo-ric: Ran intelligent controller for mimo xapps," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 2315–2322.
[12] M. Bakulin, T. Ben Rejeb, V. Kreyndelin, D. Pankratov, and A. Smirnov, "Multi-user mimo downlink precoding with dynamic user selection for limited feedback," *Sensors*, vol. 25, no. 3, p. 866, 2025.
[13] R. Ntassah, G. M. Dell'Aera, and F. Granelli, "Interference-aware pmi selection for mimo systems in an o-ran scenario," *arXiv preprint arXiv:2504.14745*, 2025.
[14] S. Park, C. Park, and J. Kim, "Learning-based cooperative mobility control for autonomous drone-delivery," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 4870–4885, 2023.
[15] M. V. Ngo, H.-M. Yoo, Y.-H. Pua, T.-L. Le, X.-L. Liang, B. Chen, E.-K. Hong, T. Q. Quek *et al.*, "Ran intelligent controller (ric): From open-source implementation to real-world validation," *ICT Express*, vol. 10, no. 3, pp. 680–691, 2024.
[16] A. Wadud and A. Basalamah, "Optical wireless communications for next-generation radio access networks," *ICT Express*, 2025.
[17] S. Park, H. Baek, and J. Kim, "Spatio-temporal multi-metaverse dynamic streaming for hybrid quantum-classical systems," *IEEE/ACM Transactions on Networking*, 2024.