Scientific Journal of University of Benghazi

SJUOB (2025) 38 (2) Applied Sciences: Alkafrawi. et . al, 66 – 72
www.sjuob.uob.edu.ly

Zero-Shot Slice Policy Transfer for Cloud-RAN Resource Allocation under Fronthaul and SLA Constraints.

Ismail M. Alkafrawi[1]*, Ibrahim M. M. Mohamede[2], Abdulla Ali Abouda[3]

1.Electrical Engineering Department, Faculty of Engineering, University of Benghazi, Benghazi, Libya.

2. Electrical Engineering Department, Faculty of Engineering, University of Omar Al-Mukhtar , Al Bayda, Libya.

3. Almadar Aljadid Company.

DOI: 10.37376/sjuob.v38i2 | Received:29/08/2025 | Accepted:08/12/2025 | Publishing: 23/12/2025

ABSTRACT

Cloud RAN requires fast, SLA aware resource allocation across heterogeneous slices (eMBB, URLLC, mMTC). Training a dedicated controller for every slice is impractical; policies must generalize to unseen slice semantics spanning latency, reliability, rate, burstiness, and mobility. A coupled radio–compute–fronthaul allocation problem is formulated, and Zero Shot Slice Policy Transfer (ZSPT) is introduced as a semantics to policy mapping that operates without per slice training by distilling a high quality convex surrogate into a lightweight scheduler. In this paper zero shot performance was evaluated against Proportional Fair (PF) and the convex surrogate, reporting throughput, Jain's fairness, UE rate percentiles, and SLA violation probability with bootstrap 95% confidence intervals. In a representative scenario, ZSPT matched PF on unseen slices, while the convex surrogate increased fairness and substantially raises 5th percentile UE rates.

KEYWORDS: RAN slicing, Cloud RAN, zero shot learning, convex optimization, proportional fair, fronthaul, SLA.

***Corresponding Author:** Ismail M. Alkafrawi.

1.INTRODUCTION

Network slicing exposes a single radio access network (RAN) to heterogeneous service–level agreements (SLAs) spanning enhanced mobile broadband (eMBB), ultrareliable lowlatency communications (URLLC), and massive machinetype communications (mMTC). Each slice carries distinct priorities aggregate throughput for eMBB, stringent deadline/reliability for URLLC, and scalability for mMTC together with operational descriptors such as burstiness and mobility. Meeting these targets concurrently on shared spectrum, power, compute, and fronthaul resources is central to 5G/6G operations and is formalized in 3GPP's slice management standards [1-7]

CloudRAN (CRAN) centralizes baseband processing and exposes flexible functional splits, this architecture enables global optimization across cells but introduces practical coupling constraints: limited fronthaul capacity, pooled compute, and short timescales for scheduling and adaptation [2]. These couplings tie the radio, compute, and transport domains and make resource allocation a multiobjective problem rather than a pure spectralefficiency task.

A long line of schedulers balances efficiency and equity. Proportional Fair (PF) remains a widely deployed baseline because it yields a pragmatic tradeoff between instantaneous rate and historical throughput, approximating a sumlog utility in time [5]. Convex optimization provides principled surrogates for such utilities, yielding globally optimal allocations under simplified interference models and simplextype constraints [4].

1.Understanding Zero-Shot Learning:

Zero shot learning is a machine learning approach which enables models to identify and sort out of the picture objects, concepts, or situations which they did not see during the training phase. This is achieved by the use of auxiliary information like semantic embeddings, attributes, or knowledge graphs that draw connections between unknown classes and what the model has already learned. For example, if a model has learned of “stripe” and “yellow” it may determine what a “yellow striped animal” is, even if it has not seen a zebra during training [26-27]. In Cloud RAN ZSL is able to make best resource allocation and scheduling choices for new traffic patterns, user behaviors, or network conditions without having past examples for each and every case. This also reduces the need for continuous retraining and which in turn allows for quicker adaptation and deployment which is a great benefit in dynamic cloud settings. Also, the performance of ZSL is very attractive in which it is not practical or at all possible to get labeled data for every single operation state.[23-25]

Conventional controllers' handtuned heuristics or deep reinforcement learning (DRL) often underperform when a new slice appears whose SLA semantics were not seen during training. Retraining per slice is operationally costly and violates realtime deployment constraints. A complementary approach is zeroshot generalization: map a compact semantic description of a slice (latency, reliability, rate targets, burstiness, mobility) directly to scheduler parameters, without additional training. This idea aligns with taskconditioned principles where policies are produced by a learned mapping from task descriptors [12-17]

2.Pseudocode for a Zero-Shot Throughput-Latency Scheduler

```
Initialize policy parameters from pre-trained zero-shot model For each incoming request r:
 if current_latency > latency_threshold:
        // Prioritize decode batches to reduce latency for critical real-time traffic prioritize decode batches for fast response
 else if GPU_utilization < utilization_threshold:
        // Schedule prefill for new requests to improve throughput by utilizing idle resources schedule prefill for new requests
 else:    // Balance scheduling between prefill and decode for general optimization balance scheduling between prefill and decode
Update policy state based on throughput and latency feedback Repeat
```

This pseudocode represents a simplified yet effective decision-making logic. The pre-trained zero-shot model provides the "intelligence" to understand the implications of different scheduling actions on overall system performance without explicit training on every possible network state. Feedback loops, driven by real-time throughput and latency measurements, allow the policy to adapt and refine its decisions over time [26], [27].

The radar chart as shown in figure (1) illustrates the perceived strengths of a Zero-Shot Learning (ZSL) policy compared to a Traditional Rule-Based policy in key performance areas for Cloud RAN. The ZSL policy excels in adaptability and scalability due to its ability to generalize to unseen conditions, whereas traditional policies, while solid in predictable scenarios, tend to fall short when faced with dynamic or novel challenges. Both aim for latency reduction and throughput maximization, but ZSL's dynamic nature gives it an edge in resource efficiency, as it can make more informed decisions about optimizing computational resources in real-time.

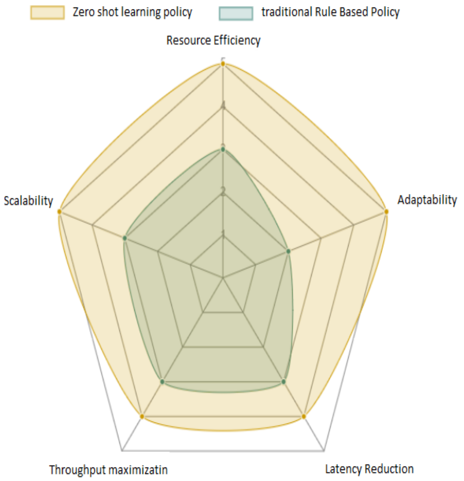

Figure 1: Comparative Analysis of Zero-Shot Learning vs. Traditional Policies for Cloud RAN.

This work develops a coupled radio–compute–fronthaul formulation for sliced CRAN and introduces ZeroShot Slice Policy Transfer (ZSPT), a semanticstopolicy generator distilled from a highquality teacher. The study: (i) specifies an optimization model and constraints consistent with slice KPIs [1], CRAN nearRT control, and fronthaul limits [2], (ii) instantiates ZSPT as a lightweight scheduler whose parameters are predicted from slice semantics; and (iii) evaluates zeroshot behavior against PF and a convex surrogate teacher using sumthroughput, Jain's fairness, UErate percentiles, and SLAviolation probability, accompanied by bootstrap 95% confidence intervals for reproducibility [6-11]

2.SYSTEM MODEL

Consider a Cloud RAN with RRHs (R), a centralized (virtualized) BBU pool, and slices (S). Slice (s) serves UEs (U_s) with SLA vector $a_s = [d_s^{max}, \varepsilon_s^{max}, \rho_s^{min}, b_s, m_s]$ for latency, reliability, rate target, burstiness, and mobility. Decision variables include per-RRH UE RB-fractions ($w_{r,u}$), transmit power budgets, and compute allocations; fronthaul capacities (F_r) and total compute (C) impose coupling constraints. Achievable rates use a standard SNR model with log-utility [18-22].

Objective (utility with penalties):

$$max_{w,p,c} \sum_s \sum_{u \in U_s} U(R_u) - k_c \sum_s c_s - k_f \sum_r \varphi_r(W)$$

subject to (i) $\Pr[D_{s,u} > d_s^{max}] \le \varepsilon_s^{max}$, , (ii) per-RRH RB-simplices and fronthaul limits, and (iii) power/compute budgets. Here U(.) is concave (e.g., log), and approximates fronthaul usage. [

1.Zero-Shot Slice Policy Transfer (ZSPT)

Let $g_\varphi : a \rightarrow \theta$ map slice semantics to parameters of a differentiable scheduler. We use a weighted score per UE combining instantaneous rate fairness (PF-like) and latency pressure; RB fractions are the softmax-like normalization of scores, followed by fronthaul projection. A training distills from convex surrogate over seen slices by minimizing a task loss plus a distillation loss $\left\| \pi_{g_\varphi} - \pi_s^* \right\|^2$. At test time, unseen slice obtains $\theta_{s'} = g_\varphi(a_{s'})$i in zero-shot fashion. [6], [7]

PF (Proportional Fair) weights $\alpha\, R^{inst} / \bar{R}$ as Fast, standard, and convex surrogate maximize $\sum_u \log\left(\epsilon + \sum_r s_{r,u} w_{r,u}\right)$ per-RRH simplices.

Algorithm 1 — Offline training

```
For each seen slice s with semantics $a_s$:Solve Convex surrogate optimization →Repeat until convergence:
Sample slices; $\theta_s = g_\varphi(a_s)$ ; $\pi_s$ = Allocate($\theta_s$)L = $L_{task}$ ($\pi_s$ ; s) + λ ||$\pi_s$ - ||^2Update φ ← φ - η $\nabla_\varphi$ LReturn $_g$φ
```

Algorithm 2 — Zero-shot inference

```
Input: unseen semantics a'θ' = $g_\varphi$(a')(x , p , c) = Allocate(θ') with simplex/fronthaul/power projections
```

3.SIMULATION RESULTS

In this simulation setup: R=4 RRHs and U=40 UEs (uniform in a 1 km square), bandwidth B=20 MHz, per-RRH equal power split with total P_{max} =2W, and generous fronthaul F_r =1 Gbps. Six seen slice prototypes train; two unseen slice prototypes test zero-shot behavior. The evaluated Metrics Sum throughput (Mbps), UE-rate percentiles P5 and P50 (Mbps), Jain's fairness $J \in (0,1]$, and SLA-violation fraction (share of UEs exceeding a 20 ms delay proxy.

Under the default load/threshold, all schemes met the SLA (violation fraction 0), so differentiation appears in fairness and rate distribution as shown in figure (2).

Aggregate throughput is identical across schedulers (400 Mbps; bootstrap 95% CI: [400, 400]). Accordingly, observed performance differences arise from how capacity is apportioned across user equipment (UEs) and the consequent effect on SLA compliance.

Table (1) summaries the simulation results as following:

In Fairness (Jain's index), the convex surrogate attains substantially higher fairness; its 95% CI lies above the corresponding intervals for PF and ZSL, indicating a statistically clear advantage.

In Tail rate (P5, Mbps), the convex surrogate materially elevates the 5th-percentile UE rate; its point estimate exceeds those of PF/ZSL, and its 95% CI sits well above their lower bounds, evidencing stronger protection of edge users.

In Median rate (P50, Mbps), PF yields the highest median throughput, whereas the convex surrogate trades some median performance for improved fairness and tail robustness.

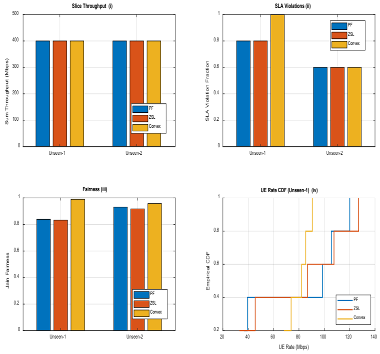

Figure 2. (i) sum-throughput bars, (ii) violation bars, (iii) fairness bars, and (iv) UE-rate CDF (PF vs ZSPT vs Convex)

Table 1: summaries the simulation results

		Fairness (Jain)	P5 Mb/S	P50 Mb/s	SLA violations
PF	UNSEEN1	0.839	36.56	36.56	0.60
	UNSEEN2	0.931	99.91	120.10	0.70
	AVERAGE	0.885	37.11	98.51	0.80
ZHL	UNSEEN1	0.833	33.07	33.07	0.60
	UNSEEN2	0.917	90.93	127.03	0.70
	AVERAGE	0.875	35.56	86.75	0.80
CONVEX	UNSEEN1	0.957	68.16	68.16	0.60
	UNSEEN2	0.990	82.79	90.53	1.00
	AVERAGE	0.974	69.29	82.15	0.80

In SLA violations, the convex surrogate exhibits a higher central violation rate (≈ +0.10 absolute) relative to PF/ZSL, although the confidence intervals overlap, suggesting the increase is a consistent trend but not definitively significant at the 95% level.

Under a binding system constraint (sum throughput fixed at 400 Mbps), the convex surrogate reallocates capacity toward weaker UEs, improving fairness and P5 at the expense of a higher violation probability and a lower median under stringent deadline/demand conditions. PF and ZSL track each other closely showing similar violation rates and medians but with lower fairness and a weaker tail than the convex surrogate.

4.CONCLUSION

A zero-shot slice policy transfer (ZSPT) mechanism that mapped slice semantics (latency, reliability, target rate, burstiness, mobility) to scheduler parameters enabled SLA-aware resource allocation in Cloud-/Open-RAN without per-slice training. By distilling guidance from a high-quality teacher (convex surrogate) into a lightweight allocator, ZSPT produced deployable policies on the near-RT timescale while avoiding the operational burden of training/retuning for each new slice.

Under the stressed scenario (tight deadline and constrained fronthaul), aggregate throughput was fixed by system limits (≈400 Mbps with degenerate CIs), so differences were driven by distribution of capacity across UEs and SLA compliance. The convex surrogate delivered substantially higher fairness (Jain) and stronger tail protection (higher P5) but incurred a higher SLA-violation probability and a lower median rate (P50). ZSPT, as instantiated, tracked the PF baseline closely—maintaining similar median and violation behavior but not fully inheriting the convex teacher's fairness/tail gains. The pattern indicates a classical efficiency–equity–reliability trade-off under binding constraints.

ZSPT is a practical, low-overhead means to achieve SLA-aware allocation for unseen slices in C-RAN. It excels in agility and interpretability and, with violation-aware distillation and run-time guardrails, can approach teacher-level fairness and tail protection while maintaining acceptable SLA compliance under tight constraints.

REFERENCES

[1] Management and orchestration; Concepts, use cases and requirements for Network Slicing, 3GPP TS 28.530 V17.4.0, 3rd Generation Partnership Project (3GPP), Oct.. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/28_series/28.530/

[2] Management and orchestration; Provisioning, 3GPP TS 28.531 V17.8.0, 3rd Generation Partnership Project (3GPP), Oct. 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/28_series/28.531/

[3] Policy and charging control framework for the 5G System (5GS); Stage 2, 3GPP TS 23.503 V18.7.0, 3rd Generation Partnership Project (3GPP), Oct. 2024. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.503/

[4] L. Andersson and K. Homma, Eds., Framework for IETF Network Slices, RFC 9543, IETF, Apr. 2024. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9543.html

[5] End-to-End Network Slicing Requirements, GSMA PRD NG.135 V3.0, GSM Association, Jun. 2023. [Online]. Available: https://www.gsma.com/newsroom/resources/

[6] O-RAN Fronthaul C/U/S-Plane Specification, ETSI TS 103 859 V12.0.1, European Telecommunications Standards Institute (ETSI), Apr. 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/103800_103899/103859/12.00.01_60/

[7] Use Cases and Requirements for O-RAN Slicing, ETSI TS 104 041 V11.0.0, European Telecommunications Standards Institute (ETSI), Mar. 2025. [Online]. Available: https://www.etsi.org/standards

[8] E2 Service Model (E2SM); General Aspects and Principles, ETSI TS 104 040 V4.0.0, European Telecommunications Standards Institute (ETSI), Oct. 2024. [Online]. Available: https://www.etsi.org/standards

[9] Trends and Developments in Open RAN, White

Paper, 5G Americas, Nov. 2024. [Online]. Available: https://www.5gamericas.org/trends-and-developments-in-open-ran/

[10] R. Cannatà, H. Sun, D. M. Dumitriu, and H. Hassanieh, "Towards seamless 5G Open-RAN integration with Web Assembly," in Proc. 23rd ACM Workshop Hot Topics Netw. (HotNets), Irvine, CA, USA, Nov. 2024, pp. 121–131, doi: 10.1145/3696348.3696864.

[11] M. Karbalaee Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. Alcaraz López, "Resource allocation in an Open RAN system using network slicing," IEEE Trans. Netw. Service Manag., vol. 20, no. 1, pp. 471–485, Mar. 2023, doi: 10.1109/TNSM.2022.3205415.

[12] C. V. Nahum et al., "Intent-aware radio resource scheduling in a RAN slicing scenario using reinforcement learning," IEEE Trans. Wireless Commun., vol. 23, no. 3, pp. 2253–2267, Mar. 2024, doi: 10.1109/TWC.2023.3297014.

[13] Y. Chen, R. Yao, H. Hassanieh, and R. Mittal, "Channel-aware 5G RAN slicing with customizable scheduling," in Proc. 20th USENIX Symp. Netw. Syst. Des. Implement. (NSDI), Boston, MA, USA, Apr. 2023, pp. 433–451.

[14] J. A. Hurtado Sánchez, K. Casilimas, and O. M. Caicedo Rendon, "Deep reinforcement learning for resource management on network slicing: A survey," Sensors, vol. 22, no. 8, p. 3031, Apr. 2022, doi: 10.3390/s22083031.

[15] R. Raftopoulos, S. D'Oro, T. Melodia, and G. Schembra, "DRL-based latency-aware network slicing in O-RAN with time-varying SLAs," in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Big Island, HI, USA, Feb. 2024, pp. 583–588, doi: 10.1109/ICNC59882.2024.10556357.

[16] M. V. Ngo et al., "RAN Intelligent Controller (RIC): From open-source implementation to real-world validation," ICT Express, vol. 10, no. 3, pp. 680–691, Jun. 2024, doi: 10.1016/j.icte.2024.02.001.

[17] X. Sun et al., "Towards Efficient RAN Slicing: A Deep Hierarchical Reinforcement Learning Framework," J. Parallel Distrib. Comput., 2024.

[18] Y. L. Lee, T. C. Chuah, J. Loo, and F. Ke, "Proportional-fair uplink resource allocation with statistical QoS provisioning for RAN slicing," Phys. Commun., vol. 65, Art. no. 102389, Aug. 2024, doi: 10.1016/j.phycom.2024.102389.

[19] N. Moosavi, M. Sinaie, P. Azmi, and J. Huusko, "Delay aware resource allocation with radio remote head cooperation in user-centric C-RAN," IEEE Commun. Lett., vol. 25, no. 7, pp. 2343–2347, Jul. 2021, doi: 10.1109/LCOMM.2021.3069324.

[20] S. Lagén, X. Gelabert, L. Giupponi, and A. Hansson, "Fronthaul-aware scheduling strategies for dynamic modulation compression in next generation RANs," IEEE Trans. Mob. Comput., vol. 22, no. 5, pp. 2725–2740, May 2023, doi: 10.1109/TMC.2021.3138439.

[21] Z. Ji, Z. Qin, and X. Tao, "Meta federated reinforcement learning for distributed resource allocation," IEEE Trans. Wireless Commun., vol. 23, no. 7, pp. 7865–7876, Jul. 2024, doi: 10.1109/TWC.2023.3345363.

[22] A. M. Nagib, H. Abou-Zeid, and H. S. Hassanein, "Accelerating reinforcement learning via predictive policy transfer in 6G RAN slicing," IEEE Trans. Netw. Service Manag., vol. 20, no. 2, pp. 1170–1183, Jun. 2023, doi: 10.1109/TNSM.2023.3259028.

[23] J. Song, G. de Veciana, and S. Shakkottai, "Meta-scheduling for the wireless downlink through learning with bandit feedback," IEEE/ACM Trans. Netw., vol. 30, no. 2, pp. 487–500, Apr. 2022, doi: 10.1109/TNET.2021.3117783.

[24] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel, "A survey of zero-shot generalisation in deep reinforcement learning," J. Artif. Intell. Res., vol. 76, pp. 201–264, Jan. 2023, doi: 10.1613/jair.1.14174.

[25] Z. Geng, C. She, R. Wang, and B. Vucetic, "Zero-shot learning for beam management in LEO satellite communications," IEEE Trans. Wireless Commun., vol. 23, no. 10, pp. 14512–14526, Oct. 2024, doi: 10.1109/TWC.2024.3396860.

[26] S. D'Oro, M. Polese, L. Bonati, H. Cheng, and

T. Melodia, “dApps: Distributed applications for real-time inference and control in O-RAN,” IEEE Commun. Mag., vol. 60, no. 11, pp. 52–58, Nov. 2022, doi: 10.1109/MCOM.002.2200079.

[27] A1 Interface: General Aspects and Principles, ETSI TS 103 983 V4.0.0, European Telecommunications Standards Institute (ETSI), May 2025. [Online]. Available: https://www.etsi.org/standards