Received 1 September 2023; revised 26 October 2023; accepted 22 November 2023. Date of publication 30 November 2023; date of current version 19 December 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3337854

Deep Learning for Intelligent and Automated Network Slicing in 5G Open RAN (ORAN) Deployment

SHU-PING YEH[1], SONIA BHATTACHARYA[2], RASHIKA SHARMA[2], AND HASSNAA MOUSTAFA[1] (Senior Member, IEEE)

[1]Intel Labs, Intel Corporation, Santa Clara, CA 95054, USA
[2]Network and Edge Group, Intel Corporation, Bengaluru 560103, India

CORRESPONDING AUTHOR: S.-P. YEH (e-mail: shu-ping.yeh@intel.com)

ABSTRACT 5G and beyond networks are considered a catalyst for emerging IoT applications and services by providing ultra-reliable connectivity and massive connections to billions of IoT sensors and devices. However, the scalable deployment of such services requires reduced cost, an open ecosystem for IoT application developers and service providers, and a multi-tenant deployment model enabling the 5G and beyond network infrastructure to host multiple IoT services while preserving the service level agreement (SLA) requirements. AI brings intelligence to the network infrastructure to automate several network functions and predict the service's workload to ensure network function scaling and adaptation. 5G brings AI to the radio access network (RAN) to reduce the operation cost, decrease power consumption and boost service quality. With this evolution towards AI-based features in the network, the Open RAN (ORAN) specification expanded the network functions virtualization to the RAN intelligence by introducing RAN Intelligent Controller (RIC) to enable AI applications for the network functions. This paper focuses on the RAN intelligence ecosystem and presents an intelligent network application (xApp) for network slicing for the RAN using AI and Deep Learning techniques. We evaluated the xApp with a near Real-Time RAN Intelligent Controller (near-RT RIC) and showed the network slicing functionality in an automated and intelligent fashion. We show how intelligent network slicing enables emerging IoT services to co-exist while meeting the required SLAs.

INDEX TERMS 5G, AI, IoT, network slicing, O-RAN, RAN intelligence, SLAs.

I. INTRODUCTION

THE RAN evolution journey started with vRAN [1], [2] to implement the RAN functions entirely in SW that can run on any COTS server and introduced disaggregation of the RAN functions with control plane and data plane separation for ease of scaling and distributed deployment with cost efficiency. Then Cloud-Native RAN [3], [4] introduced vRAN implementation in microservices form with containerization to ease deployment and scaling for the vRAN workload and to enable real-time orchestration with automation and resource-efficiency. Open RAN [5] then introduced standardized open interfaces in the 5G RAN to enable co-existence across the ecosystem players and to open opportunities for new players, and defined the specification for RAN intelligent elements.

With all this evolution, 5G and beyond bring incredible promise in terms of performance improvements, deployment at scale, and opportunities for new services. However, creating new services to leverage the promised capabilities will require service providers to evolve to deliver the demanding use cases that 5G and beyond can unlock. This evolution observes the convergence of the SW-defined network functions and IoT/new services at the edge, applying a microservices-based deployment model adapting the cloud-native approach to the edge. In this ecosystem, AI enables several intelligent services for IoT and Enterprises and expands to the network functions intelligence for automation and reduced OPEX. So, now, AI provides the means for intelligent services and intelligent network functions. A broad and deep set of analytics is needed to derive actionable

insights on the network status in real-time to adapt the service delivery for diverse use cases.

In this context, RAN intelligence introduces intelligent elements in the 5G RAN through the RAN Intelligent Controller (RIC) [6]. RIC enables AI applications for network functions automation saving OPEX and CAPEX to telcos and enhancing QoS.

This paper focuses on RAN intelligence and introduces a Deep Learning-based solution for network slicing in the RAN helping 5G and beyond network to offer not only a means for connectivity but a network infrastructure capable of hosting new services from multiple providers while meeting the required SLAs. The rest of this paper is organized as follows: Section II presents the related work on RAN intelligence and shows the addition we bring through our contribution, Section III introduces our framework for automated and intelligent RAN slicing, Section IV describes the Deep Learning based solution for traffic load prediction and radio resource management for network slices, Section V explains the xApp integration with ORAN RIC, Section VI shows the test results. We conclude the paper in Section VII and present the future work plan.

II. RELATED WORK

Network Slicing brings several benefits to 5G and beyond networks and drives new business models and innovative use cases across all industries. Network slicing enables operators to efficiently manage the network resources and provide differentiated services at scale. Network slicing also provides service flexibility and the ability to provide high security, isolation, and QoS characteristics to meet the SLA. In this paper, we focus on network slicing at the RAN level, and in this section, we review the related work and introduce the motivation for our solution.

For radio resources management and coordination, [7] focuses on inter-slice interference in mobile networks and proposes a solution that uses a non-convex integer programming model to formulate the problem. The solution reduces the interference effect by swapping resource blocks (RBs) between slices in multi-cell multi-slice networks. While [8] presents a framework that provides hierarchical RAN slicing considering the network bandwidth slice for each gNodeB, the RAN slices that each gNodeB provides for the services, and the bandwidth slice at a granular packet level in each RAN slice. Transfer Learning is applied by leveraging the KPI status for services in each slice to provide a feedback loop to the slice decision. Also, [9] studies the RAN slicing and the coordination between RAN slices through a bi-convex problem formulation that accounts for dependencies between resource allocation for each slice and the coordination of slices sharing the same resources. Algorithms are introduced to address the formulated bi-convex problem. The work in [7] and [9] do not consider SLAs through dedicating network slice(s) for each service, but however focus on the radio resources split and management, while [8] looks at network slices from a holistic view for the overall cells' deployment, without focus on the dynamic SLAs needs for services.

For dynamic slicing to satisfy QoS needs SLAs, [10] proposes Reinforcement Learning (RL) algorithms to solve dynamic radio resource management (RRM) problems in 5G networks and beyond, mainly considering a scenario in which SLAs priorities of the available network slices change. To enhance the RL convergence in this scenario, predictive Transfer Learning is used to use policies from previous network slices. While [11] proposes a data-driven resource-sharing algorithm running at the *Slice Orchestrator* (SO) level for sharing RAN resources across heterogenous slices. The RAN slices resources are adjusted periodically using feedback on users' Channel Quality Indicator (CQI). A machine learning approach infers the stability of UE channel conditions with predictive schemes. And [12] developed a deep learning model for dynamic network slicing to optimize the service quality in real time. The proposed model considers large number of users and different service requirements, and it has a reduced training time allowing operators to update the model with reduced development effort. The work in [10], [11], and [12] do not show how the proposed network slicing solutions comply with the ORAN specification with a RIC but are limited to the algorithms side.

Focusing on RAN slicing for vertical solutions, RAN Slicing for factory automation is considered in [13], proposing a solution to guarantee the reliability requirements for industrial applications through introducing a RAN slice descriptor considering reliability and transmission rate requirements for the applications. An analytical model is proposed to define the resources for each slice per application traffic type. Reference [14] considers RAN slicing for two types of services (eMBB and V2X) and presents a RAN slicing scheme using RL and a heuristic algorithm with a target of maximizing resource utilization while fulfilling service requirements for each slice. Reference [15] considers the case of Unmanned Aerial Vehicles (UAVs) expanding the RAN range to provide connectivity for new emerging services and in case of limited RAN resources and introduces a resource allocation algorithm for UAV RAN slicing to meet the services QoS requirements. The work in [13], [14], and [15] mainly consider the network slice association to the service, however, do not consider the algorithms or the deployment perspective to enable the RAN network slicing.

While the research contributions in RAN slicing is rich, we noticed that most of the existing research contribution focus on either RAN resources management to offer network slicing, intelligent algorithms to guarantee QoS/KPIs for services, and offering resources sharing with RAN slicing to help new emerging solutions as V2X and UAV. All these contributions present big value to help the industry leverage the proposed algorithms to build RAN slicing solutions in an ORAN deployment. Also, most of the existing contributions use simulation for validation of the proposed algorithms,

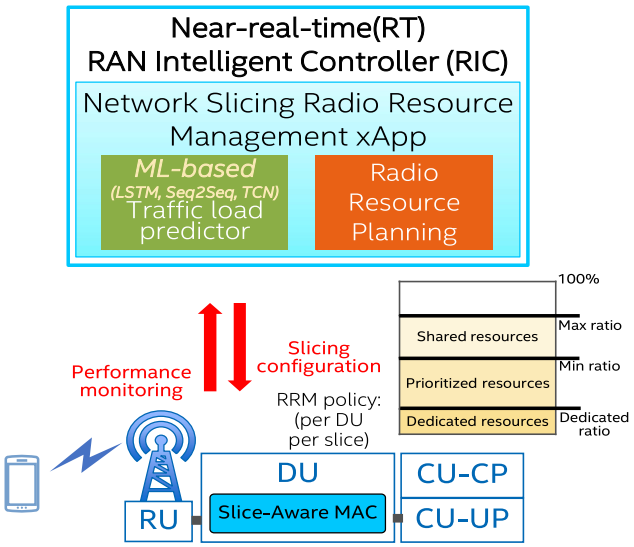


FIGURE 1. Intelligent and automated RAN Slicing Framework.

which will require new validation with real data sets from the network to help deployment in an ORAN architecture.

Our contribution adds to the existing research contributions by introducing AI-based algorithms for dynamic RAN slicing meeting multi-services SLAs, implementing the algorithms in an xApp form following the ORAN specification, and integrating with an ORAN compliant near-RT RIC and doing detailed performance evaluation. Our contribution offers a reference implementation available for download and that the industry and academia can use and train the AI algorithms with diverse data matching diverse deployment scenarios, where we offer training capabilities.

III. INTELLIGENT AND AUTOMATED SLICING

To fulfill the end-to-end service SLA negotiated between network slice service provider and network slice customer, network operators need to carefully monitor and automatically update configurations for every part of the network, including core, transport, and RAN. In this paper, the focus is SLA assurance in RAN for diverse co-existing services handled by the network operators.

We introduce an intelligent network slicing resource management framework based on O-RAN [5] architecture as illustrated in Fig. 1. O-RAN introduced RAN intelligent Controller (RIC) to host RAN control functions as micro-services, called xApp, for cloud-native RAN management. Our framework includes an intelligent network slicing radio resource manager implemented as an xApp for semi-static radio resource planning and dynamic slice-aware scheduling performed at MAC (medium access control) layer.

The intelligent network slice radio resource management (NSRRM) xApp interacts with O-RAN compliant radio access nodes via O-RAN E2 signaling received and sent through the near-RT RIC. Our NSRRM xApp continuously monitors RAN conditions and network slice SLA from RAN measurements reported by E2 Service Model (SM) for Key performance measurements (KPM) [16] and then calculates proper RAN configurations to assure slice SLA. One type of RAN configurations that can help assure slice SLA is to allocate dedicated or prioritized radio resource to a network slice. 3GPP defined RRM policy in [17] that specifies three types of radio resource ratios: dedicated ratio for the portion of radio spectrum that is reserved exclusively for a network slice, minimum ratio for the portion of radio spectrum that a network slice has the highest priority to access, and maximum ratio for the maximum portion of radio spectrum that can be used to serve a network slice. The NSRRM xApp calculates RRM policies for each slice and sends the control to O-RAN compliant RAN nodes via O-RAN E2 signaling. There are two O-RAN E2 service models support the configuration of RRM policy: E2SM RAN Control (RC) [18] and E2SM Cell Configuration Control [19]. In our work, E2SM-RC is used to configure RRM policies for each cell.

The NSRRM xApp determines the quantity of radio resource to be reserved or prioritized for network slice traffic usage whereas the slice-aware MAC scheduler enforces the radio resource reservation or prioritization rules provided by the NSRRM xApp. The MAC scheduling functionality is a layer-2 RAN function located in gNB or the distributed units (DU) in CU (centralized unit)-DU split deployment. MAC scheduling determines which data flow(s) to be transmitted over the air for each radio resource block (RB). This procedure dynamically adjusts radio resource allocation to each user to achieve the best performance based on channel condition and the QoS target for the flow. With network slicing, MAC scheduler should be aware of slicing configuration and be able to enforce the slicing related configuration to assure SLA.

IV. DEEP LEARNING BASED SOLUTION

The intelligent RAN slicing manager consists of two key modules: traffic load prediction and radio resource planning.

The radio resource planning module would calculate the amount of radio resource to be reserved or prioritized for a network slice based on the forecast of network slice loading provided by the traffic load prediction module and the estimated spectrum efficiency for users belong to the target network slice. User spectrum efficiency can be estimated based on UE throughput and radio resource utilization measurements collected through E2SM-KPM.

We applied deep learning for traffic load predictions. Traffic load data, such as PDCP data volume measurements, can be collected via E2 messages for E2SM-KPM. Depending on SLA requirements, the time scale for traffic load prediction and the loss function design may be different. We utilized Intel AI Tool for time-series prediction to develop a flexible training pipeline that can process data with different sampling granularities and apply different loss functions for model training. Fig. 2 illustrates the detailed ML training pipeline with BigDL-Chronos [20] for:

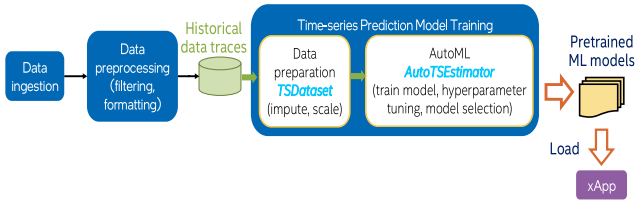


FIGURE 2. ML training data pipeline for time series prediction model.

TABLE 1. Prediction performance for traffic data sampled at different rate.

Data sampling rate	Optimal Model	RMSE (Mbps)
5 min	Seq2seq	76.13
10 min	TCN	63.57
15 min	Seq2seq	61.74
20min	LSTM	57.66
30min	Seq2seq	56.48

- Data preparation: training and testing data can be prepared as ***TSDataset***, where BigDL-Chronos supports API to handle missing data, normalization and feature generation, e.g., datatime and rolling feature.
- Built-in ML models: BigDL-Chronos supports multiple widely used time-series prediction models, including LSTM, Seq-to-seq, TCN, Autoformer, ARIMA and more.
- AutoML: BigDL-Chronos offers ***AutoTSEstimator*** for automatic hyperparameter tuning. Trained ML model is saved as a ***TSPipeline*** object.

A ML model trained from historical traces with good prediction performance can be integrated into a rApp at non-RT RIC or a xApp at near-RT RIC.

Table 1 summarizes the best root mean square error (RMSE) performance for the ML models trained via the AutoML features in BigDL Chronos with 3 candidate ML models: long short-term memory (LSTM), Sequence-to-sequence (Seq2seq) and temporal convolutional network (TCN), when sampling data at different time granularity for the public dataset from MAWI WIDE project [21].

Via hyperparameter tuning from BigDL-Chronos, we observe an average improvement in RMSE by 5Mbps. Most publicly available traffic datasets are sampled at time granularity at minute level. We expect that traffic load prediction module with prediction time scale at minute level to be implemented as rApp in non-RT RIC and the output from traffic load predictor can be used for longer term radio resource coordination between multiple base stations. In addition, the cost of SLA violation can be incorporated into loss function design for the cross-cell slice resource coordination problem when performing traffic load prediction.

When traffic load variation exhibits short-term pattern, such as periodic wireless data exchange in factory operation, traffic prediction at shorter time scale can be applied in xApp for better radio resource adaptation with load variation. Fig. 3 shows the throughput and delay performance with

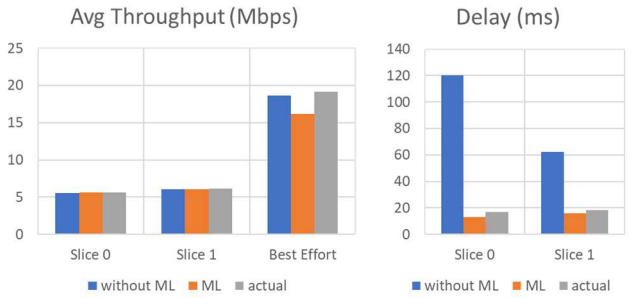


FIGURE 3. Throughput and delay performance with and without ML.

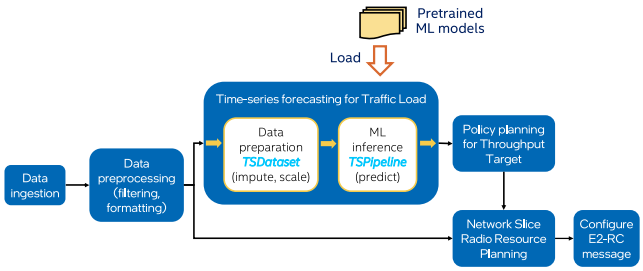


FIGURE 4. Intelligent RAN slicing xApp data pipeline and ML inference data pipeline for time series prediction model.

and without ML prediction for 2 delay sensitive slices with synthetic generated traffic pattern. The grey bars indicate the optimal performance when ground truth is provided for the radio resource planner. We observe that ML prediction can greatly improve delay performance, but throughput performance is less sensitive to prediction accuracy. The BigDL Chronos offers the flexibility for customized loss function which can be used to design different loss functions when training the traffic predictor for different types of per flow SLA targets.

In the next section, we describe the proof of concept we developed with the traffic predictor module implemented as part of the Near-RT NSRRM xApp. Figure 4 shows the data pipeline for traffic prediction inference and RAN-slicing resource management xApp.

V. REFERENCE IMPLEMENTATION WITH NEAR RT-RIC

We developed a reference implementation for the Network Slicing xApp to be able to test it with a near-RT RIC and evaluate its performance. We used an ORAN compliant near-RT RIC from the ONF SD-RAN project and an E2 simulator associated with the RIC. The reference implementation release [22] is available to help the research and industrial community. It can serve testing of multiple services and use cases (e.g., mission-critical applications, real-time applications, AR/VR, and immersive media) prior to deployment in an ORAN 5G and beyond network.

We also tested the reference implementation in an Edge Native Kubernetes-based node leveraging Intel Smart Edge Open Developer Experience Kit (DEK) [23] for optimized deployment for the xApp compute-intensive workload. Fig. 5 provides an overview on the xApp deployment architecture.

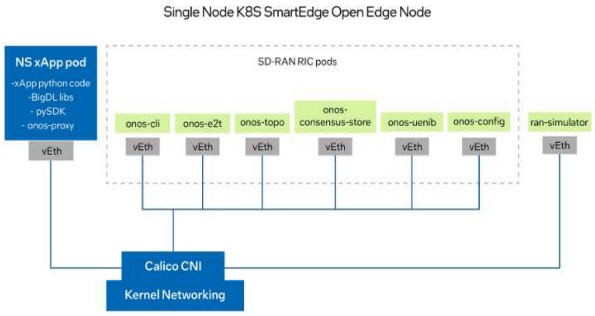


FIGURE 5. Deployment Architecture on Intel Edge Native Platform.

TABLE 2. Input dataset in the RAN simulator.

Slice	Timestamp	Pdcp_rate	Utilization	Volume
1	9/22/2023 9:09:41 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:41 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:42 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:42 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:43 AM	115683.75	0.06	116225.25
2	9/22/2023 9:09:43 AM	265160.25	0.15	269996.85
1	9/22/2023 9:09:44 AM	149343	0.08	149993.25
2	9/22/2023 9:09:44 AM	450535.5	0.25	459510.3

- NS xApp, BigDL, pySDK are all containerized using Kubernetes framework for cloud-native applications provided by Intel Edge Native Platform
- SD-RAN v1.4.129 is deployed on Intel Smart Edge Open Developer Experience Kit. It supports the E2SM KPM service model v2.
- Calico as data plane: Calico CNI is default CNI used for communication between xApp pod, RAN simulator pod and SD-RAN pods. Calico is optimized for high performance networking.
- Network Slicing (NS) xApp interacts with the SD-RAN RIC using the Python SDK.

VI. TESTED USE CASES

The testbed consists of NSxApp 1.0, sdran 1.4.129 and SD-RAN RAN Simulator v1.4.15 all deployed as containerized pods on edge native Kubernetes-based node leveraging Intel Smart Edge Open Developer Experience Kit (DEK). Please refer Fig. 5.

The tests are done using SD-RAN RAN Simulator in which the following data is generated. Table 2 shows the input data set for the RAN simulator.

- *pdcp_rate:* the average transmitted data throughput.
- *Utilization:* the average number of PRB (physical resource blocks) used to transmit traffic belong to a network slice.
- *Volume:* the incoming data volume of a slice

After fetching the E2 nodes, the NS xApp subscribes to an E2 node using the E2SM-KPM service model. In this test bed, RAN simulator is simulating one E2 node, one cell, one UE. Once the subscription is successful, the RAN simulator sends the above measurements in the form of E2SM-KPM indications to the RIC which in turn sends these indications to the xApp. Once the xApp starts receiving these indications, it monitors the measurements in the indications and via the traffic load prediction and radio resource planning modules, it calculates the PRBs for each network slice. Here the number of network slices used are two as seen in Table 2.

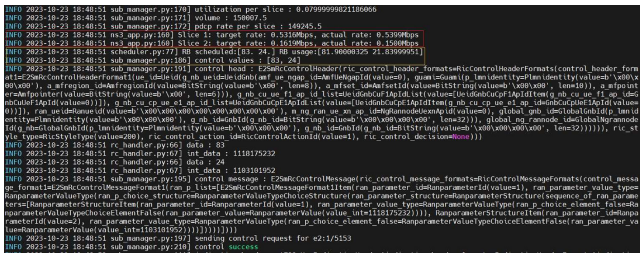


FIGURE 6. Predicted PRBs for Slice Allocation.

After calculating the dedicated number of PRBs, a Control request for both the slices is sent to the SD-RAN, using the E2SM-RC service model. Fig. 6 shows a sample of the predicted PRBs for the slice allocation.

In the Figure, the red box shows that the target rate, based on traffic load prediction is close to the actual traffic data rate. Similarly, the yellow box shows that the calculated PRB allocation is very close to the actual PRB usage. The calculated PRBs are sent as control request to the SD-RAN for the two slices.

VII. CONCLUSION

Network Slicing is a key technology in 5G and beyond networks opening opportunity for services co-existence while guaranteeing the required SLAs for each service. The contribution in this paper focuses on RAN Network Slicing and presents intelligent automated network slicing algorithms applying deep learning techniques to build an ORAN compliant xApp for these algorithms. A full reference implementation was created to show-case network slicing xApp integrated and tested with ORAN compliant near-RT RIC from open source ONF SD-RAN project. The results obtained so far are very encouraging and sets up pedestal for future work on Intelligent and Automated Network Slicing to be expanded to advanced use cases of intelligent traffic flows predication and data aggregation management. In addition to applying ML for traffic prediction to realize automatic radio resource adaptation to slice traffic, other ML techniques, including applying reinforcement learning to enhance radio resource allocation for slices with diverse SLA targets, such as mix of slice requiring low latency and slice requiring guarantee throughput. It is also envisioned that ML can be used to provide capacity and QoS predictions that would be useful for feasibility check for network slice resource provisioning. The solution can also be further expanded to a hierarchical slice management framework with slice

SLA assurance rApp at non-RT RIC providing longer-term regional SLA guidance to each Near-RT RIC and NSRRM xApp at Near-RT RIC to derive radio allocation based on SLA guidance and RAN measurements.

APPENDIX
ABBREVIATIONS AND ACRONYMS

AI: Artificial Intelligence
ARIMA: Auto Regressive Integrated Moving Average
E2SM: E2 Service Model
eMBB: Enhanced Mobile Broadband
CAPEX: Capital Expenditure
CNI: Container Network Interface
COTS: Commercial off-the-Shelf
CU-CP: Centralized Unit - Control Plane
CU-UP: Centralized Unit - User Plane
DU: Distributed Unit
gNB: Next Generation NodeB
LSTM: Long Short Term Memory Networks
MAC: Medium Access Control
ML: Machine Learning
Near-RT RIC: Near Real-Time RAN Intelligent Controller
NSRRM: Network Slice Radio Resource Management
ONF: Open Networking Foundation
OPEX: Operational Expenditure
ORAN: Open RAN
PDCP: Packet Data Convergence Protocol
PRB: Physical Resource Block
QoS: Quality of Service
RAN: Radio Access Network
RIC: RAN Intelligent Controller
RL: Reinforcement Learning
RRM: Radio Resource Management
RU: Radio Unit
rApp: Non-RT Intelligent Application
SD-RAN: Software-Defined Radio Access Network
SLA: Service Level Agreement
TCN: Temporal Convolutional Neural Network
UAV: Unmanned Aerial Vehicle
UE: User Equipment
V2X: Vehicle to Everything
vRAN: virtualized RAN
xApp: Extended Application - Intelligent app for near-RT

ACKNOWLEDGMENT

The authors are grateful to the ONF SD-RAN project for all the support during the xApp integration with the RIC. They are also very thankful to Phani Bhushan Athlur from Intel for all his inputs with his team for the reference implementation development, integration, and validation, and Shilpa Talwar from Intel Labs for sponsoring the network AI research effort. Finally, they would like to thank Menglei Zhang from Intel Labs for the simulation support in the early stages of the project.

REFERENCES

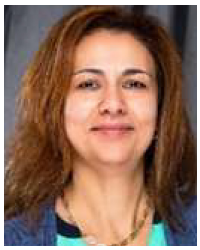
[1] (3GPP, Sophia Antipolis, France). *NG-RAN Architecture*. Accessed: Feb. 19, 2021. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/ng-ran-architecture
[2] (VMware Cloud Comput. Co., Palo Alto, CA, USA). *Telco Cloud Platform—RAN Reference Architecture Guide 2.2, vRAN Architecture Overview*. (2022). [Online]. Available: https://docs.vmware.com/en/VMware-Telco-Cloud-Platform-RAN/2.2/telco-cloud-platform-ran-reference-architecture-guide-22/GUID-840C5E5C-134B-4E36-8F7D-FA212ABB363B.html
[3] (Ericsson, Stockholm, Sweden). *The Four Key Components of Cloud RAN*. (2020). [Online]. Available: https://www.ericsson.com/en/blog/2020/8/the-four-components-of-cloud-ran
[4] "Study on management of cloud-native virtualized network functions (VNF)," 3GPP, Sophia Antipolis, France, Rep. TR 28.834, 2023.
[5] "ORAN Alliance." 2023. o-ran. [Online]. Available: https://www.o-ran.org/
[6] (O-RAN Alliance e.V., Alfter, Germany). *RAN Intelligent Controller Architecture*. (2019). [Online]. Available: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html
[7] C. Yang, T. Wang, Z. Xiao, and S. Wang, "A suggest-and-improve RAN slicing strategy for mobile networks," in *Proc. IEEE 14th Int. Conf. Wireless Commun. Signal Process.*, 2022, pp. 748–752.
[8] J. Mei, X. Wang, and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," *Intell. Converg. Netw.*, vol. 1, no. 3, pp. 281–294, Dec. 2020.
[9] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for RAN network slicing," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 970–973, Dec. 2018.
[10] A. Nagib, H. Abou-Zeid, and H. 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
[11] S. Bakri, P. A. Frangoudis, A. Ksentini, and M. Bouaziz, "Data-driven RAN slicing mechanisms for 5G and beyond," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4654–4668, Dec. 2021.
[12] S. Kumar et al., "Dynamic network slicing using deep reinforcement learning," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2021, pp. 384–389.
[13] M. M. Rashid, M. Carmen Lucas-Esta, M. Sepulcre, and J. Gozalvez, "5G RAN slicing to support reliability in industrial applications," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, 2022, pp. 1–4.
[14] H. Albonda and J. Perez-Romero, "An efficient RAN slicing strategy for a heterogeneous network with eMBB and V2X services," *IEEE Access*, vol. 7, pp. 44771–44782, 2019.
[15] J.-W. Cho, P. Yang, T. Q. S. Quek, and J.-H. Kim, "Service-aware resource allocation design of UAV RAN slicing," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2020, pp. 801–805.
[16] *Near-RT RIC, E2 Service Model (E2SM) KPM v02*. O-RAN Alliance e.V., Alfter, Germany, 2021.
[17] "5G network resource model," 3GPP, Sophia Antipolis, France, Rep. TS 28.541, 2018.
[18] *Near-RT RIC, E2 Service Model (E2SM) RAN Control v01*. O-RAN Alliance e.V., Alfter, Germany, 2021.
[19] *Near-RT RIC, E2 Service Model (E2SM) Cell Configuration Control v01*. O-RAN Alliance e.V., Alfter, Germany, 2022.
[20] "BigDL-chronos online documentation," Intel. Santa Clara, CA, USA, document, 2020. [Online]. Available: https://bigdl.readthedocs.io/en/latest/doc/Chronos/index.html
[21] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *Proc. Annu. Conf. USENIX Annu. Tech. Conf.*, San Diego, CA, USA, 2000, p. 51.
[22] "The network slicing reference implementation release." intel.com. 2023. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/reference-implementation/network-slicing-radio-resource-management.html
[23] "Intel® smart edge developer experience kit (DEK)." intel.com. 2022. [Online]. Available: https://www.intel.com/content/www/us/en/developer/articles/reference-implementation/smart-edge-open-developer-experience-kit.html

SHU-PING YEH received the M.S. and Ph.D. degrees in electrical engineering from Stanford University in 2005 and 2010, respectively. She is a Senior AI-Applied Research Scientist with the Wireless System Research Lab, Intel, where she conducts research on wireless broadband technologies. She has over ten years of research and development experience in wireless industry and holds more than 40 U.S. patents. Her current research focus includes open RAN architecture, AI/ML for RAN control and management, network slicing, and interworking of multiple radio access technologies within a network.

SONIA BHATTACHARYA was born in New Delhi, India. She received the M.Sc. degree (Hons.) in computer science from Delhi University, India, in 2007, where she was awarded the Gold Medal for the highest marks.

She is a Software Engineering Manager with the Network and Edge Division, Intel Corporation and is working on Edge Computing and AI xApps on network edge infrastructure. She has over 16 years of experience in wireless protocols and has published whitepapers on connection management xApp in Intel Network Builders Group. Her current focus is on building edge native node software which will enable the extensions and compute needed to run AI-based solutions on the edge seamlessly. Prior to this, her focus was on integrating 5G solutions in the edge infrastructure to enable the private 5G use cases for mobile-edge computing.

RASHIKA SHARMA received the bachelor's degree (batch topper) from the Computer Science Department, Indian Institute of Technology Guwahati, India, and the M.Tech. degree in computer science and engineering from the Indian Institute of Technology Guwahati in 2021. She is a Cloud Software Development Engineer with the Edge Services Division, Intel Corporation, where she is working on Edge Computing and AI-based xApps. She has over two years of work experience in various domains that include AI-based xApps in which she has contributed to whitepapers on connection management xApp, edge computing platform, and security-related tasks like fuzzing, for which she also won the NPS Security Assurance Bug Bounty Award. Her current work focuses on integrating 5G solutions in edge infrastructure, along with UI/UX for realizing E2E solutions deployment.



HASSNAA MOUSTAFA (Senior Member, IEEE) received the master's degree in distributed systems from the University of Paris XI, and the Ph.D. degree in wireless and mobile networks from Telecom Paris Tech. She also holds a Research Director Qualification from the University of Paris XI. She is a Principal Engineer with Intel Corporation, working on edge computing and AI solutions across IoT segments and network edge infrastructure. Previously at Intel, she led car-to-cloud solutions for connected/autonomous vehicles and connectivity technologies across IoT segments. Before joining Intel, She was a Senior Research and Development Engineer with Orange, France. She contributed to wireless network solutions for the EMEA region and led engineering efforts on optimizing video and multimedia services over wireless networks. She has over 80 publications in international conferences, journals, and books and over 300 filed patents (over 100 granted).