O-RAN and Spectrum Sharing

1st Sonali Chaudhari
Department of Computer Science
North Carolina State University
Raleigh, USA
sschaud2@ncsu.edu

2nd Dongming Wu
Department of Computer Science
North Carolina State University
Raleigh, USA
dwu6@ncsu.edu

3rd O'Neal M'Beri
Department of Computer Science
North Carolina State University
Raleigh, USA
ocmberi@ncsu.edu

***Abstract*—The number of wireless devices and their required data rates are increasing with time. Spectrum is an expensive and shared resource that has to be used effectively. Spectrum sharing is a key challenge for service providers that enables the sharing of the same frequency band amongst multiple users of different priorities without impeding one another. Open Radio Access Networks (ORAN) is a new architecture that provides open interfaces and AI/ML capabilities to promote innovation for future networks. In this work, we study existing work done for spectrum sharing in ORAN and replicate one of these solutions.**

***Index Terms*—ORAN, spectrum sharing, Environmental Sensing Capability (ESC)**

I. Introduction

With the popularity of the Internet of Everything, numerous devices have been added to the network. The ability to observe, analyze, and control the system has revolutionized multiple industries. Industry 4.0 is a concept that aims to achieve full digitization and automation in factories for efficient and reduced-cost operations, while Industry 5.0 pushes the need for connectivity forward and provides connectivity in their products for their better customer service, hyper customization, responsive and distributed supply chain, and interactive product [1]. Along with various devices, new applications such as virtual reality, augmented reality, video streaming, and cloud-based services have emerged in recent years, which demand high data rates from the network. This increase in potential users has changed the cellular network and demands spectrum sharing to cater to new devices with different requirements like high capacity, high data rates, and ultra-low latency. Cellular networks have wide coverage and provide wireless connectivity required by many of these devices. Wireless connectivity depends upon the spectrum, an expensive resource that must be utilized efficiently to serve future users.

II. Background

A. Spectrum sharing

Spectrum sharing is a technique that enables multiple users or services to operate on the same frequency band, optimizing the use of limited wireless resources. In the context of 5G and LTE, spectrum sharing allows both technologies to coexist on the same frequency, facilitating a more efficient use of bandwidth[2]. The key benefits of spectrum sharing include reduced costs for service upgrades, faster access to 5G, and improved network performance for both 5G and LTE[3].

B. Open Radio Access Network (O-RAN)

Open Radio Access Network (O-RAN) refers to standards and architecture designed to make radio access networks (RAN) more open, flexible, and software-driven. O-RAN enables the disaggregation of hardware and software, allowing operators to source equipment from different vendors and integrate them seamlessly using open interfaces. This disaggregation removes the traditional vendor lock-in, giving operators more flexibility and lowering costs [4].

Besides, O-RAN integrates advanced AI and machine learning technologies through the RAN Intelligent Controller (RIC) [5], enabling real-time network optimization by automating decision-making processes and adapting network behavior dynamically. This integration allows for the intelligent orchestration of network resources including network slicing and spectrum allocation, ensuring optimal performance and quality of service (QoS), energy efficiency, and operational cost reduction [5, 6, 7].

C. Spectrum sharing with O-RAN

For spectrum sharing, it is essential for commercial cellular service providers to detect interference with incumbent traffic and prioritize them. Commercial networks can allocate the same resources to their users if an incumbent does not use them. Today, networks use dedicated sensors known as Environmental Sensing Capability (ESC) to detect the presence or absence of incumbent transmission. Establishing and maintaining such a dedicated sensor infrastructure is expensive. The emergence of O-RAN has enabled innovation in spectrum sharing and interference detection without depending upon such an additional infrastructure. In this work, we will study some solutions proposed in the literature that use O-RAN to detect interference in spectrum sharing.

III. Problem definition

SenseORAN is a solution that uses O-RAN-compliant base stations for sensing to maximize the accuracy of radar detection in the Citizens Broadband Radio Service (CBRS) band (3.55-3.7 GHz) while maintaining an acceptable level of service for the associated clients. This solution is especially effective for scenarios where pulses fully overlap with interfering LTE signals requiring immediate detection of such occurrence. The authors proposed two stages: sensing slice and network reconfiguration. In the Sensing slice stage, all

TABLE I: System Setup

Core	12
Memory	48 GB
Operating system	Ubuntu 20.04 LTS (x86/64)
Disk	75 GB

the base stations observe the RF spectrum for a finite duration and create a spectrogram. This spectrogram is sent to xApp, which consists of a trained image classification network. The xApp can detect interference under diverse traffic conditions by using an off-the-shelf convolutional neural network (CNN) architecture. The network reconfiguration stage turns off all the network operations and excludes the bands where the interference is detected. This solution was implemented in the Open AI Cellular (OAIC) platform, which uses open cellular software and software-defined radios [8]. Figure 1a shows the system setup for SenseORAN. In this work, we propose to implement and replicate results from SenseORAN.

IV. Implementation and current status

Multiple platforms are available for O-RAN-based implementations, like OpenGym, ns-o-ran, and OAIC. We identified that the authors have implemented their work on OAIC. The OAIC platform also offers tutorials to help us set the prerequisites. Hence, we chose to work on the OAIC platform for our experiments.

A. *Open Artificial Intelligence Cellular (OAIC)*

Open Artificial Intelligence Cellular (OAIC) is an Open research platform that includes software, libraries, toolsets, and documentation for prototyping and testing AI-based radio access network (RAN) controllers. OAIC uses well-established open-source software for cellular radio communications and implementations for core, eNB, and EPC from Software Radio Systems (SRS). The E2 agent and RIC implementations are taken from the OSC near RT RIC, and xApps are community-driven.

V. Experiment Setup

We used the Google Cloud Platform for our setting up the virtual environment as per specifications mentioned in table I

For this implementation, we installed the following components:

- Near-real time RIC: This component hosts the E2 interface docker and the xApps. It uses Kubernetes, dockers, and Helm to implement the components and services.
- srsRAN: The srsRAN in OAIC is modified to include the E2 interface and its capabilities. The authors have provided implementation for eNB, which collects I/Q samples and sends spectrograms to the xApp.
- xApp: xApp is the application running on the near-real-time RIC that collects the I/Q spectrogram from the base station and modifies the modulation and coding scheme based on the interference detection. Instead of using the E2AP specification, the proposed work uses an E2-like interface for ease of implementation. In our implementation, the xApp has dummy logic for interference detection. The xApp successfully connects with the eNB, collects the spectrogram from the eNB, and sends hardcoded commands to the eNB. The xApp and eNB support two configurations: static Modulation and Coding Scheme (MCS) and dynamic MCS. Due to the set dummy logic for interference detection, xApp, is expected to send the hard-coded command instructing to use dynamic MCS.

Figure 1 compares the proposed system in the paper and our current implementation.

We verified the expected establishment of connection and data transfer between an xApp and srsRAN. The following are the low level details about our setup:

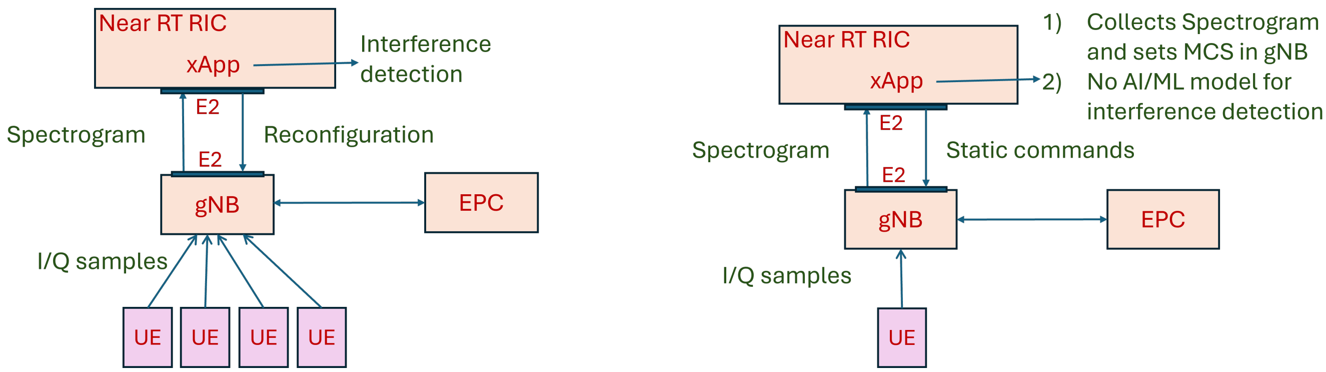
1) **xApp Setup**:
   - Exposed port 5000 using `kubectl expose deployment` to create a NodePort service.
   - Identified the rerouted port using `kubectl get svc -A` and stored it in `XAPP_PORT`.
2) **srsRAN Initialization**:
   - Navigated to the srsRAN build directory and launched the EPC using `sudo srsepc/src/srsepc`.
   - Exported the local IP as `HOST_IP` before starting the base station (`srsenb`), which connected to the xApp.
3) **Base Station and I/Q Data**:
   - Confirmed that `srsenb` established a connection and initiated data exchange.
   - Verified reception of E2-like messages and I/Q data processing.
4) **User Equipment (UE) Setup**:
   - Created a network namespace for UE and started it using `sudo srsue/src/srsue --gw.netns=ue1`.
   - Observed the successful attachment of the UE with an assigned IP (e.g., 172.16.0.3).
5) **Data Transfer and Monitoring**:
   - Launched an `iperf3` server on the nodeB and connected from the UE side, confirming data transfer rates (e.g., $\sim$10 Mbps).
   - Monitored MCS changes and verified adaptive MCS command responses in xApp logs.

Results

- The xApp successfully connected via SCTP, processed incoming I/Q data, and responded to control signals.
- Logs did not indicate any interference during the experiment and hence xApp commands eNB to use static MCS.

VI. Roadblocks

We followed the tutorials on the OAIC platform but still encountered many issues during this exercise. We could resolve



(a) Overview of the system proposed in the paper

(b) Implemented system with proposed interactions but dummy xApp

Fig. 1: Comparison of proposed system in the paper and our implementation

issues with srsRAN deployment by asking for help from the groups and communities working on the same platforms. We documented the steps that worked for us and wrote scripts for quick setup. We still have some issues with the logic in xApp that cannot detect the interference, but we are working to resolve them.

Conclusion and Future Work

No interference was detected during the experiment. Potential reasons include a controlled environment, low network load, configuration thresholds limitations. Future work will involve testing with varied conditions and enhanced interference simulation.

VII. Future work

Until now, we have created an O-RAN-Based 5G network consisting of one EPC, one eNB, one UE, near-real-time RIC, and one xApp. We established the expected interaction between these components, such as UE attaching to the network, BS sending the spectrogram to xApp, and xApp sending E2 commands to the eNB. For future work, we will focus on the xApp logic so that the xApp can dynamically control the MCS for the eNB based on interference detection.

VIII. Contributions of the Team Members

- Sonali Chaudhari:
  - Initial research for OAIC platform, available resources, and tutorials
  - Hands-on the tutorial and initial setup for the experiment
  - Resolved issues and documented the steps for the setup that worked for us
- Dongming Wu:
  - Repeated and verified the documented steps for the setup
  - Identified a few more issues and scripted these steps to make the network deployment easy and repeatable
  - Initial groundwork on xApp for interference detection
- O'Neal M'Beri:
  - comprehensive exploration of the dummy xApp's implementation and its difference from proposed xApp in paper
  - deep dive into I/Q sample collection logic and identify any difference between the proposed logic in the paper and the implementation
  - Proposed future work and continued work

References

[1] W. Xiang et al. "Advanced Manufacturing in Industry 5.0: A Survey of Key Enabling Technologies and Future Trends". In: *IEEE Transactions on Industrial Informatics* (Feb. 2024).

[2] Wan Siti Halimatul Munirah Wan et al. Ahmad. "5G technology: Towards dynamic spectrum sharing using cognitive radio networks". In: *IEEE Access* (2020).

[3] B. G. Barb, F. Alexa, and M. Otesteanu. "Dynamic Spectrum Sharing for Future LTE-NR Networks". In: *Sensors (Basel)* (2021).

[4] M. Sheikh et al. *O-RAN Evolution Towards Open and Intelligent RAN Architecture*. arXiv preprint arXiv:2202.01032. 2022.

[5] B. Balasubramanian et al. "RIC: A RAN Intelligent Controller Platform for AI-Enabled Cellular Networks". In: *IEEE Internet Computing* (Mar. 2021).

[6] Parallel Wireless. *OpenRAN Network Software Suite*. [Online]. Available: https://www.parallelwireless.com/wp-content/uploads/OpenRAN-Network-SW-Suite.pdf. 2024.

[7] S. K. Singh, R. Singh, and B. Kumbhani. "The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities". In: *Computer Networks* (2024).

[8] G. Reus-Muns et al. "SenseORAN: O-RAN-Based Radar Detection in the CBRS Band". In: *IEEE Journal on Selected Areas in Communications* (Feb. 2024).