1. Introduction and Context

The activity of this PoC focuses on the evolution of 5G+ technologies towards multiorbital, multiband network architectures to provide a reliable connection for TSN. A core aspect of this activity is the deployment and validation of dual connectivity within a containerized 5G Standalone (SA) architecture. We aim to implement and testing of such a setup using open-source tools — namely **srsRAN** and **Open5GS**. This testbed explores advanced features including dual connectivity, multipath routing, and network slicing, highlighting both the capabilities and constraints of the system.

2. System Overview

The deployed 5G SA architecture comprises multiple key components:

- **Core Network**: Deployed inside a containerized Ubuntu 22.04 system hosted on a DGX A100 server, leveraging 8x A100 GPUs.
- **Centralized Unit (CU)**: Operates within a Docker container and is responsible for managing radio resources.
- **Distributed Units (DU1 & DU2)**: Deployed externally to handle radio communication through separate physical interfaces.
- **User Equipment (UE1 & UE2)**: Simulated in network namespaces, connected via ZMQ (ZeroMQ) based radio frequency emulation.
- **ZMQ RF Interface**: Facilitates interaction between the DUs and the UEs.
- **Routing Mechanism**: Dual routing tables configured using policy-based routing to support distinct IP paths for each UE.

This system allows for the testing of complex 5G SA network configurations, especially focusing on dual connectivity, which is essential for multi-path and high-availability communications.

3. Functional Achievements

We are integrating multiple components and have validated a range of functionalities:

- **End-to-End Connectivity**: Established PDU sessions through the CU and DU split architecture.
- **Dual IP Routing**: Achieved using `ip rule` and `netplan`, enabling multipath support for UEs.
- **Subscriber Management**: Integrated MongoDB with Open5GS, supporting user profile and SIM management via a WebUI.
- **Connectivity Testing**: Employed tools like `ping`, `iperf3`, and Multipath TCP (MPTCP) to validate throughput, latency, and redundancy.

UEs successfully established tunnels (tun_srsue), obtained IPv4 addresses, and demonstrated stable uplink/downlink connectivity, confirming the feasibility of dual connectivity in a simulated 5G SA context.

4. Technical and Operational Challenges

Despite its success, the deployment faced multiple challenges:

- **Routing Conflicts**: The Linux kernel's default behavior favored a single interface, disrupting dual connectivity. This was mitigated using custom routing tables and policy rules.
- **Container Networking**: The CU required privileged network access, which was achieved using Docker's `--network host` and `--privileged` flags.
- **Timing and Synchronization**: ZMQ interface demanded precise tx/rx synchronization to avoid access failures.
- **Resource Bottlenecks**: MongoDB and the WebUI introduced significant memory loads. System performance was optimized by assigning a dedicated bind IP to MongoDB and refining systemd service scripts.

These hurdles revealed the nuanced interplay between software-defined network components and container orchestration within 5G infrastructure.

5. Compatibility and Interoperability Considerations

Integration of disparate software components brought several interoperability issues:

- **CU-AMF Binding Conflicts**: Mismatched configurations in `cu.yml` and `amf.yaml` led to NGAP failures, resolved through meticulous verification of IP and port bindings.
- **ZMQ Dependency Mismatches**: Required consistent versions of `libzmq` and `czmq` across CU, DU, and UE systems.
- **Subnet Planning**: Binding mismatches at the interface level caused SCTP dropouts, necessitating careful subnet and address planning.

These issues underscore the complexity of deploying modular, interoperable systems using open-source stacks.

6. Performance Analysis and Bottlenecks

The system's performance was rigorously tested using tools like `iperf3`:

- **Maximum Throughput**: Reached ~18.5 Mbps with UE2 under optimal conditions.
- **Performance Bottlenecks**: UE1 experienced limitations (~8.86 Mbps), particularly during retransmission scenarios.
- **Retransmission Rates**: Elevated rates were traced to ZMQ's unstable emulation under high bandwidth. These were partially mitigated by tuning transmission and reception gain parameters.

The variability in throughput and reliability reveals the current limits of software-based 5G simulation tools, especially for high-throughput and real-time applications.

7. Root Cause and Mitigation Summary

A detailed root cause analysis table is presented, summarizing problems and their corresponding solutions:

Issue	Root Cause	Mitigation
Routing Table Conflict	Kernel default priority	Custom `ip rule` and `netplan`
SCTP Drop	Incorrect CU/AMF binding	IP and port reconfiguration
Throughput Bottleneck	ZMQ sync and RF gain settings	Tx/Rx synchronization and gain tuning

These solutions have effectively stabilized the platform and allowed for repeatable, consistent test results.

8. Implementation Status (Component-Level)

Each system component was evaluated for operational status:

- **Open5GS Core**: ✅ Fully functional via `run_core.sh`
- **CU**: ✅ Successfully connected to the AMF; F1-C interface stable
- **DU1 & DU2**: ✅ Verified F1-C links
- **UE1 & UE2**: ✅ Active tun interfaces; stable IP assignment
- **ZMQ RF**: ✅ Fully operational and synchronized
- **MongoDB WebUI**: ✅ Accessible, functional for SIM provisioning
- **Multipath TCP (MPTCP)**: ✅ Actively monitored with multi-UE connection

This confirms that the architecture supports concurrent UE sessions with reliable multipath transport.

9. Real-Time Output Validation

Console outputs from CU and UE instances confirm successful system operation:

- **CU Output**: Demonstrates active connections to AMF, readiness for new F1-C connections.
- **UE1 Output**: Shows successful attachment, PRACH transmission, and PDU session establishment (IP: 10.45.0.5).
- **UE2 Output**: Mirrors UE1 with similar success metrics (IP: 10.45.0.3).

The logs show that the UEs are fully functional within the network, confirming protocol compliance and connectivity.

10. Conclusion and Future Directions

The testbed deployment within the TRANTOR framework marks a significant milestone in the evolution of 5G+ multiorbital networks. By leveraging open-source technologies in a containerized environment, the project has demonstrated a functional, dual-connectivity 5G SA architecture.

Key Achievements:

- Fully integrated CU-DU-UE communication model
- Verified end-to-end session management
- Established dual IP routing and multipath TCP

Next Steps:

- Expand to support dynamic slicing and real-time service orchestration
- Integrate satellite link emulation for multiorbital experimentation
- Optimize system performance through kernel-level tuning and GPU offloading

This research lays the groundwork for future explorations into scalable, resilient, and high-performance 5G+ network architectures aligned with TRANTOR's overarching goals.