A Simulation Tool for Dynamic Functional Split Management in 5G and B5G Networks

Valentín Brienza
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
vbrienza@fing.edu.uy

Marcos Rapetti
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
marcos.rapetti@fing.edu.uy

Lucas Inglés
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
IMT Atlantique
Brest, France
lucasi@fing.edu.uy

Alberto Castro
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
acastro@fing.edu.uy

Claudina Rattaro
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay
crattaro@fing.edu.uy

***Abstract*—In virtualized and disaggregated Radio Access Network (RAN) deployments, the choice of functional split involves trade-offs among latency, transport requirements, and computational load. However, existing open-source simulation frameworks provide limited support for evaluating different functional splits and dynamically reconfiguring them during simulation. This paper presents a Simu5G-based tool that models a disaggregated CU–DU gNB and supports three 3GPP functional split options (Splits 2, 4, and 6). Its core component is a Split Manager entity that centralizes packet forwarding across the protocol stack, allowing the active split to be switched at runtime without altering the simulation's module composition. The tool also models per-node processing delays and finite queues, and supports centralized deployments in which multiple DUs share a common CU. Experimental results in representative 5G scenarios illustrate the impact of functional split selection on end-to-end latency and packet loss. The source code is publicly available.**

***Index Terms*—Functional split, simulation, 5G, OMNeT++**

I. Introduction

The heterogeneous service requirements of 5G and beyond 5G (B5G) networks impose diverse constraints on latency, throughput, reliability, transport capacity, and computational resources. Radio Access Network (RAN) architectures are therefore moving away from monolithic, vendor-specific deployments toward flexible, virtualized, and disaggregated designs, a trend reinforced by the Open RAN and O-RAN initiatives [1]. A key design decision in disaggregated RAN architectures is the functional split, which determines how gNB protocol-stack functions are distributed among the Centralized Unit (CU), Distributed Unit (DU) and Radio Unit (RU). Different split points provide different degrees of centralization and impose different requirements on transport capacity, processing resources, and latency [2], [3]. Moreover, the most suitable split may change over time as user demand, radio conditions, transport load, and computational resources vary. Dynamic functional split management has therefore emerged as a promising approach for adapting the degree of centralization to changing network conditions.

Several works have investigated functional split selection and dynamic adaptation. Optimization-based approaches determine the degree of centralization according to transport capacity, processing constraints, or service requirements [4], [5], [6], while prototype-based environments such as OpenAirInterface and open-source O-RAN testbeds have been used to assess disaggregated RAN deployments [7], [8]. However, many existing evaluations rely on mathematical models, specific testbed implementations, or platforms that are not readily extensible, making it difficult to evaluate alternative split configurations and adaptation strategies under common and reproducible conditions. Open-source network simulation provides a complementary approach. Simu5G is an OMNeT++-based system-level framework for end-to-end performance evaluation of 5G networks [9], [10]. However, it does not natively model a disaggregated gNB supporting multiple functional split options or runtime split reconfiguration. Other open-source tools, such as 5G-LENA, provide fronthaul-related functionality for specific split configurations, including O-RAN 7.2x, but do not target the dynamic evaluation of multiple split points [11], [12].

To address this gap, this work extends Simu5G with a disaggregated CU–DU gNB model supporting functional Splits 2, 4, and 6.[1] The main contributions are: (i) a common CU–DU architecture for evaluating different degrees of centralization without requiring a separate node model for each split; (ii) a Split Manager that dynamically routes traffic through the protocol stack according to the active split, enabling runtime split reconfiguration; and (iii) an experimental evaluation illustrating the impact of split selection on end-to-end latency

[1]The source code is publicly available at https://gitlab.fing.edu.uy/mobile-optical-networks/simulator/simusplit.

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

and packet loss in representative 5G scenarios.

II. Simu5G Extension for Functional Split Management

This section describes the proposed extension of Simu5G for modeling and evaluating functional split configurations in 5G and B5G RAN scenarios. The extension reuses the validated Simu5G protocol stack and node models, and introduces additional modules to manage the separation of RAN functions across centralized and distributed entities. Its design follows three principles: preserving compatibility with the Simu5G simulation workflow, so that existing scenarios can be adapted with limited modifications; supporting different functional split configurations through a common internal architecture, avoiding a distinct node model per split option; and enabling runtime modification of the active split point, providing a basis for evaluating dynamic functional split management strategies.

A. Disaggregated gNB Model

To support functional split simulation, a new module named `gNodeB_split` is introduced. This module replaces the original Simu5G `gNodeB` in scenarios that require RAN disaggregation, and it can be configured as either a Centralized Unit (CU) or a Distributed Unit (DU), depending on parameters defined in the simulation configuration files.

This design allows CU and DU nodes to share the same internal structure, while their behavior is determined by their configured role and by the selected functional split. In the current implementation, three split configurations are supported: split 2, between the PDCP and RLC layers; split 4, between the RLC and MAC layers; and split 6, between the MAC and PHY layers. These options represent different degrees of centralization and, therefore, different requirements on the transport network and processing resources. The resulting module structure, common to CU and DU, is shown in Fig. 1.

B. Split Manager

The SplitManager acts as a central forwarding entity within the `gNodeB_split` network interface card (NIC). Instead of requiring each protocol layer to determine whether the next function is local or remote, all protocol submodules forward their traffic to the `SplitManager`, which then decides how each packet must be forwarded according to the active split configuration and the traffic direction, as shown in Fig. 2. For packets that do not cross the selected split boundary, the `SplitManager` forwards the traffic to the next local protocol submodule within the same node. In contrast, when the next protocol function is located at the remote side of the split, the packet is sent through the split interface to the corresponding CU or DU. This behavior is applied in both uplink and downlink directions, allowing the same architecture to support different split configurations.

This centralized forwarding approach provides two advantages. First, it enables splits 2, 4, and 6 to be implemented using the same internal structure, reducing code duplication and simplifying scenario configuration. Moreover, this static structure allows switching between splits without changing module composition, which is not supported by OMNeT++ simulations. Second, it provides a natural control point for runtime split reconfiguration, since the forwarding decision is concentrated in a single module that can be updated during simulation.

C. Support for Multiple DUs

The 3GPP architecture allows one CU to be connected to multiple DUs (3GPP TS 38.801). To support this topology, the CU-side `SplitManager` can manage multiple DU connections simultaneously, with the CU–DU interfaces implemented as variable-length gate vectors. In the uplink direction, this implementation is straightforward because packets arriving from the DUs are forwarded to the CU and then routed to the core network. In the downlink direction, however, the CU must determine the DU associated with each packet received from the upper layers. To support this operation, the CU maintains an address table that maps each UE context to the output gate associated with its serving DU.

D. Processing Delay and Queue Modeling

Comparing functional split points requires modeling not only the transport delay between the CU and the DU but also the processing delay introduced by each node. For this purpose, the extension introduces a configurable parameter, `processingDelay`, that provides an abstract representation of the processing time associated with each node. During this processing interval, the `SplitManager` does not process the next packet in its queue, allowing the model to capture the effects of limited processing capacity. The parameter can be configured as a constant value or as a random variable following any of the probability distributions supported by OMNeT++. To model node overload, the implementation also includes finite-size packet queues for incoming traffic, implemented using the `PacketQueue` module from the INET library. The queue size, packet-dropping policy, and packet-ordering criterion are configurable, enabling modeling of congestion and packet loss due to limited processing resources. In the CU case, where multiple DUs may be connected to the same centralized node, a separate queue is implemented for each DU connection. A common scheduler serves these queues, implemented using the `PriorityScheduler` module from INET, which selects packets from the first queue with pending traffic and can be replaced by other INET scheduling modules or a custom policy. The resulting module is shown in Fig. 3.

E. Runtime Split Configuration

The proposed extension allows the active functional split to be configured at simulation startup and modified during simulation execution. Each `gNodeB_split` module includes a parameter indicating the split point to be used, and consistency across the disaggregated gNB is ensured by making the CU the master entity for split configuration. At the beginning of the simulation, the CU sends a control message to each connected DU indicating which split option to use, and the DU updates

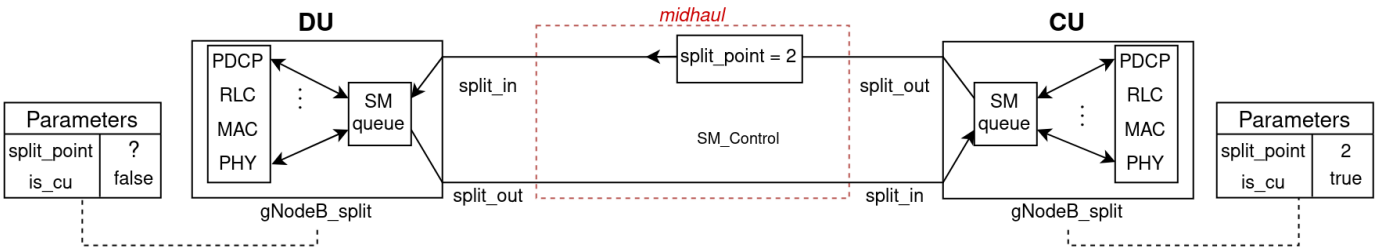


Fig. 1. Structure of the implemented `gNodeB_split` module and its external interfaces. The same module can represent either a CU or a DU, depending on the configured parameters. The figure also illustrates the initial configuration mechanism, in which the CU sets the split option for the connected DU via a control message.

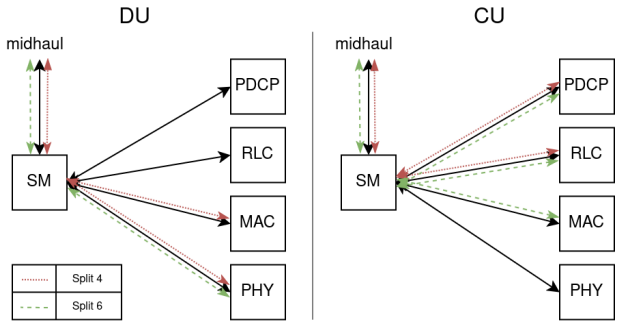


Fig. 2. Implementation of functional splits using the centralized `SplitManager`. The links used in the CU and DU are shown for split 4 and split 6.

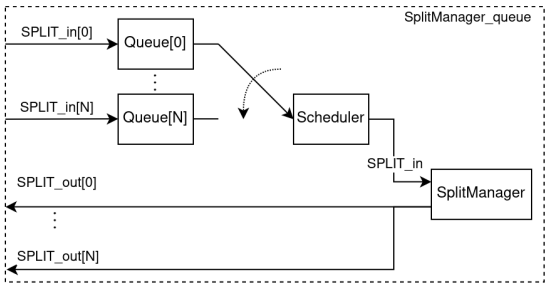


Fig. 3. Internal structure of the `SplitManager_queue` module.

its local configuration accordingly, ensuring that both sides of the split apply the same forwarding logic.

The same control mechanism also provides the basis for dynamic functional split reconfiguration. During simulation, the CU or an external decision entity can trigger a change in the selected split point by sending an updated configuration message to the DU. Once the new configuration is applied, the `SplitManager` updates its forwarding decisions based on the new split boundary. In this way, the extension enables the evaluation of adaptive split management strategies under changing network conditions.

III. Simulation Results and Evaluation

This section evaluates the impact of different functional split configurations on end-to-end latency and packet loss as traffic load increases.

A. Simulation Setup

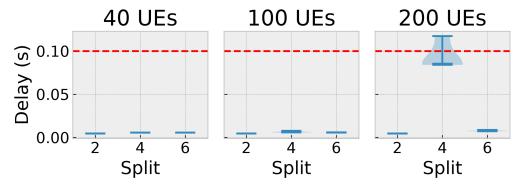
All simulations use the following configuration. The radio access network is configured with 50 available resource blocks, numerology 0, UE and gNB transmission powers of 20 dBm and 40 dBm, respectively, and the `NRChannelModel_3GPP38_901` channel model, following the 3GPP TS 38.901 channel model specification. The Max C/I scheduler is used at the MAC layer, and UEs are placed according to a ring distribution at a distance of 300 m from the DU. The CU and DU queues have a finite capacity of 20 packets, while the default processing delays are set to 50 μs at the DU and 5 μs at the CU. Traffic consists of VoIP flows with 40-byte packets.

B. Impact of Functional Split Selection

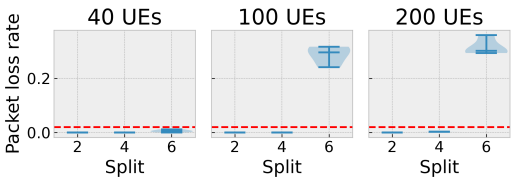
These experiments use a topology with one DU and one CU connected through an ideal link that introduces no propagation delay, transmission errors, or bandwidth limitations, so that the evaluation isolates the processing and queuing effects of the different split configurations. We analyze the system behavior as the number of UEs increases: a higher number of users raises the traffic load and may congest the processing queues of the CU or DU, and configurations that centralize lower-layer functions, such as split 6, are expected to be more sensitive to processing delays and timing constraints.

Fig. 4 shows the end-to-end delay and packet loss ratio for downlink VoIP traffic, while Fig. 5 presents the corresponding results for uplink traffic. In the downlink case, split 6 begins to experience packet losses as the number of users increases, and these losses become significant under high load. Although packets successfully delivered under split 6 exhibit low latency, the increased loss indicates that this configuration cannot meet the timing requirements of the centralized MAC operation under heavy load, making it unsuitable for services requiring high reliability.

In contrast, split 2 and split 4 exhibit more stable packet loss behavior. However, split 4 shows an increase in end-to-end delay in the highest-load downlink scenario. This behavior is associated with the additional interactions required in the downlink path, where the RLC layer must notify of available traffic, wait for resource allocation, and then transmit the corresponding data. In the uplink direction, the impact on delay is less pronounced because the RLC layer primarily

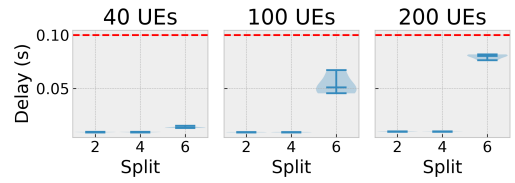


(a) DL end-to-end delay

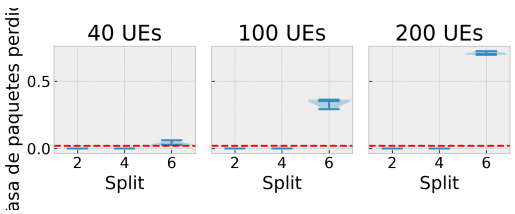


(b) DL packet loss

Fig. 4. End-to-end delay and packet loss ratio for downlink VoIP traffic, averaged over all users, for the three implemented split options.



(a) UL end-to-end delay



(b) UL packet loss

Fig. 5. End-to-end delay and packet loss ratio for uplink VoIP traffic, averaged over all users, for the three implemented split options.

stores packets received from the MAC layer in its buffer before forwarding them to upper layers.

These results illustrate that split selection introduces different processing and queuing trade-offs. Under the considered configuration, higher centralization levels become more sensitive to processing delays as traffic load increases. The most appropriate configuration therefore depends on the traffic load, the service requirements, and the processing capabilities available at the CU and DU.

IV. Conclusion

This paper presented an extension of Simu5G for modeling and evaluating functional split management in 5G and B5G RAN scenarios, based on a disaggregated gNB model that can be configured as a CU or a DU. Its central component is the `SplitManager`, which controls packet forwarding according to the selected split point and provides the basis for runtime split reconfiguration during simulation. The extension also models processing delays and finite queues, enabling the simulation of congestion at CU and DU nodes, and supports multiple DUs connected to a common CU. Results show that the selected split significantly affects end-to-end latency and packet loss, with configurations that centralize lower-layer functions becoming more sensitive to processing delays under high traffic load. Future work will focus on extending the set of supported split options and on dynamic decision policies that select the split configuration from network conditions such as traffic load, queue occupancy, latency, and available processing capacity, as well as on more detailed CU–DU transport models including bandwidth limitations, delay, jitter, and packet losses.

Acknowledgment

This work was funded and supported by the “Intelligent Management of Optical-Mobile Networks through Disaggregated Architectures and Digital Twins” research project (ANII FMV_1_2025_1_186855), the “L’Oréal–UNESCO” WIS program, the national postgraduate scholarship program (ANII POS_NAC_2024_1_183216), and the CSIC project 22420250200292UD (Iniciación a la Investigación).

References

[1] J. Navarro, A. Castro, C. A. Astudillo, and C. Rattaro, “Demystifying o-ran implementations under resource-constrained hardware: an experimental evaluation,” *Annals of Telecommunications*, May 2026.

[2] L. M. P. Larsen, A. Checko, and H. L. Christiansen, “A survey of the functional splits proposed for 5g mobile crosshaul networks,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 2019.

[3] J. Pérez-Romero, O. Sallent, A. Gelonch, X. Gelabert, B. Klaiqi, M. Kahn, and D. Campoy, “A tutorial on the characterisation and modelling of low layer functional splits for flexible radio access networks in 5g and beyond,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2791–2833, 2023.

[4] A. Martinez Alba and W. Kellerer, “A dynamic functional split in 5g radio access networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[5] D. Harutyunyan and R. Riggio, “Flex5g: Flexible functional split in 5g networks,” *IEEE Transactions on Network and Service Management*, vol. PP, pp. 1–1, 07 2018.

[6] Y. Wu, L. Liang, Y. Jia, W. Wen, and Z. Chen, “Slicing enabled flexible functional split and multi-dimensional resource provisioning in 5g-and-beyond ran,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 2, pp. 1213–1227, 2024.

[7] S. Matoussi, I. Fajjari, S. Costanzo, N. Aitsaadi, and R. Langar, “5g ran: Functional split orchestration optimization,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1448–1463, 2020.

[8] J. Luis Herrera, S. Montebugnoli, D. Scotece, L. Foschini, and P. Bellavista, “A tutorial on o-ran deployment solutions for 5g: From simulation to emulated and real testbeds,” *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 1709–1748, 2026.

[9] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, “Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks,” *IEEE Access*, vol. 8, pp. 181 176–181 191, 2020.

[10] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, ser. Simutools ’08. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[11] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, “An e2e simulator for 5g nr networks,” *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019.

[12] K. Koutlia and S. Lagén, “On the impact of open ran fronthaul control in scenarios with xr traffic,” *Computer Networks*, vol. 253, p. 110722, 2024.