Poster: Towards Accelerating the 5G Centralized Unit with Programmable Switches

Xin Zhe Khooi*, Archit Bhatnagar*†, Satis Kumar Permal*, Nishant Budhdev‡, Cha Hwan Song*, Mun Choon Chan*
National University of Singapore* BITS Pilani† Nokia Bell Labs‡

CCS CONCEPTS

• **Networks** → **Mobile networks**; **Programmable networks**.

KEYWORDS

5G, cellular networks, centralized unit, radio access network, open RAN, hardware acceleration, programmable switches, offloading

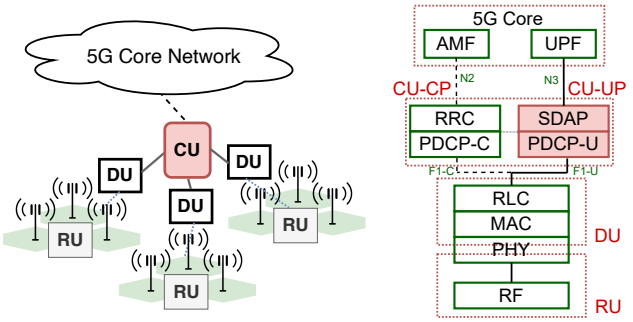
ACM Reference Format:
Xin Zhe Khooi, Archit Bhatnagar, Satis Kumar Permal, Nishant Budhdev, Cha Hwan Song, and Mun Choon Chan. 2023. Poster: Towards Accelerating the 5G Centralized Unit with Programmable Switches. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3603269.3610839

1 INTRODUCTION

5G networks are envisioned to support various emerging use cases, such as telemedicine, remote construction, autonomous driving, industrial automation, drone control, and immersive entertainment. These applications demand low latency, high reliability, and in some cases require ultra-high-bandwidths. Specifically, these applications require 5G networks to provide 1ms end-to-end latency with 99.99% reliability [12] for ultra-reliable low-latency communications (URLLC). Various studies [11, 19, 26] have shown that the radio access network (RAN) remains the bottleneck in realizing low-latency communications.

The 5G RAN consists of three main components [4], namely the Radio Unit (RU) [21], the Distributed Unit (DU), and the Centralized Unit (DU). Fig. 1a shows a CU connected to multiple DUs as it acts as their primary logic unit in a disaggregated RAN deployment. Additionally, CU acts as the "gateway" between the Core Network and the RAN/User Equipment (UE) as it sees all user data as well as control data in the network.

Network operators have also adopted virtualization in the 5G RAN to support a large number of use cases with varying requirements. While virtualization offers operational flexibility and potential for continuous evolution using commodity hardware like general-purpose processors (GPPs), it also introduces communication and processing overheads that compound the challenges in a disaggregated RAN deployment. Despite efforts to optimize



(a) Disaggregated RAN. **(b) 5G radio stack.**

Figure 1: Disaggregated RAN and 5G New Radio (NR) protocol stack. In this work, we focus on the CU-UP.

performance, GPPs still face challenges in satisfying the performance SLAs, i.e., "taming the tail" [5, 18]. Additionally, GPPs have poor scalability and hence cannot support high network bandwidth demand without significant cost [20]. To address these limitations, hardware accelerators such as FPGAs [16], and GPUs [17] are introduced to complement GPPs, especially for the computationally intensive PHY layer processing in the DU [25]. However, the scalability of the CU, which serves as the aggregation point of the RAN, has not received sufficient attention despite the massive data rates expected in 5G networks. Therefore, the implementation of highly scalable hardware-accelerated CUs, becomes crucial for supporting 5G applications, especially URLLC applications with strict latency and reliability constraints.

The CU [4] consists of two parts, i.e., the control plane (CU-CP) and the user plane (CU-UP). The CU-CP is tasked with Radio Resource Control (RRC) [3] in managing UE connections to the RAN, whereas the CU-UP is responsible for the higher layers of the 5G New Radio (NR) stack (Fig. 1b) – Service Data Adaptation Protocol (SDAP), and the Packet Data Convergence Protocol (PDCP) [2]. The SDAP mainly deals with QoS flow mapping, while the PDCP is more complex with functionalities like transmission buffering, packet re-ordering, packet de-/duplication, header compression, ciphering, and integrity verification. SDAP and PDCP layers aside, the CU-UP also handles GPRS Tunneling Protocol (GTP) [1] tunnel encap/decap for the DU ⇔ CU and CU ⇔ UPF traffic. In this work, we focus on functions in the CU-UP.

Our key insight here is that key CU functionalities are similar to traditional packet processing functions like table lookup, header modification, tunnel encap/decap, sequencing, and multicast that can be implemented using programmable switches. However, not all functionalities can be (efficiently) realized on programmable switches due to the limited hardware capabilities of these switches

ACM SIGCOMM '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0236-5/23/09…$15.00
https://doi.org/10.1145/3603269.3610839

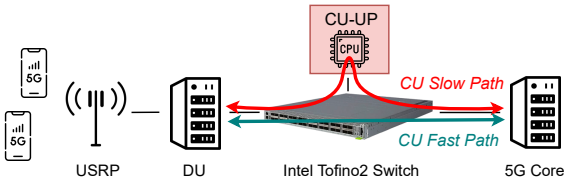


Figure 2: 5G testbed and CUP$_4$ prototype. We show the GPP-based slow path and the Tofino2-based fast path.

to ensure high-speed processing with strict latency guarantees. For instance, ciphering and integrity verification. That said, the standards [3] allow room for concessions on selectively enabling or disabling certain functionalities at the PDCP layer while maintaining a fully functional RAN. This leaves the opportunity open for a multi-Tbps CU-UP running on programmable switches for use cases/ users that do not require more complex functionalities. For instance, ciphering (which is more computationally intensive) may not be needed given the predominantly encrypted application traffic. Despite so, there are cases where a subset of users require PDCP features that are not available on the hardware-accelerated CU. Thus, full-fledged software CU fallbacks are still necessary to maintain comprehensive support at the CU.

To that end, we present CUP$_4$, a hierarchical CU design with a fast path using programmable switches and a slow path using GPPs. CUP$_4$ enables the offloading of user-plane traffic, whenever possible, at the CU to support mobile users at a greater scale – one CUP$_4$ instance can easily support orders more DUs. CUP$_4$ provides added flexibility for network operators to delegate users who are latency sensitive (e.g., URLLC) as well as throughput sensitive to benefit from the line-rate packet processing capability of programmable switches.

2 CUP$_4$: PROTOTYPE IMPLEMENTATION

Our 5G standalone (SA) testbed uses the open-source OpenAirInterface (OAI) [15] 5G RAN [10] and the OAI 5G Core [9]. We deploy them across three commodity servers (with dual 16-core Intel Xeon Gold 6326 CPUs @ 2.9 GHz, with the power governor set to performance mode) connected to an Intel Tofino2 [7] switch (see Fig. 2). The radio front-end used was the USRP B210 USB Software Defined Radio from Ettus Research [23]. We used two UEs equipped with Quectel RM500Q-GL (Qualcomm X55) 5G modules [22].

Our current CUP$_4$ prototype does not modify the OAI CU. Instead, we passively monitor[1] both DU ⇔ CU and CU ⇔ UPF bidirectional traffic using an offloading daemon (~300 lines of Python) to identify traffic that can be offloaded to the fast path. In particular, the GTP Tunnel Endpoint Identifier (TEID), N-PDU number (NPDU), Sequence Number (SeqNum) in the GTP header, and the QoS Flow Indicator (QFI) in the GTP extension header are modified by the fast path based on how it is done at the slow path. We implement the CU functionality on an Intel Tofino2 [7] programmable switch in ~400 lines of P4 [6] as the fast path. Similar to the slow path, the fast path does the necessary header modifications, GTP tunnel encap/decap, and sequencing as configured by the daemon into the match-action tables on the programmable switch. The sequence numbers are kept track of using register arrays. In addition, the fast path rule entries are also configured with aging timers to automatically remove inactive rules, i.e., when a UE goes idle.

As our approach *transparently* offloads traffic from the OAI CU software slow path onto the Tofino2 ASIC fast path, the RRC layer at the CU-CP is unaware of the fast path. This results in the RRC eventually treating the offloaded flows at the fast path as "idle" and subsequently releasing the radio connection, thus disconnecting the user. To solve that, the CUP$_4$ fast path periodically mirrors a copy of the offloaded traffic of a user to the slow path to "refresh" the RRC timer.

[1]We disable ciphering protection at the PDCP for the digital radio bearers (DRB) and thus are able to observe the interactions in the clear.

Table 1: CU processing latency.

Latency	Fast Path (in μs)	Slow Path (in μs)
Median	0.52	40.05
99%-ile	0.60	15,040.45
99.9%-ile	0.61	34,000.19

3 PRELIMINARY EVALUATION

First, we perform functional verifications on CUP$_4$ in our 5G testbed. With live user traffic, when the offloading daemon is enabled, we verify that CUP$_4$ transparently offloads user traffic from the slow path onto the fast path with no disruption. Then, we measure the processing latency of the software OAI CU (slow path) and CUP$_4$ (fast path). Packets are timestamped using the programmable switch and mirrored to an external collector for analysis. Traffic is generated with the UE running an iPerf3 client transmitting a 1 GB file over UDP to an edge-located iPerf3 server connected to the 5G core. Preliminary evaluations (see Table 1) show that the slow path has a noticeably higher processing latency (up to 80X difference, at the median) and exhibits a significantly long tail latency (up to 25, 000X, at the 99^{th}-percentile) as compared to the Intel Tofino2-based fast path.

4 FUTURE WORK

CUP$_4$ presents an early prototype for a hardware-accelerated CU by introducing a fast path that offloads the user plane (CU-UP) transparently onto programmable switches. As a proof-of-concept, CUP$_4$ currently only supports the key functionalities of the CU, e.g., GTP encap/decap, and PDCP sequencing. We continue to explore how CUP$_4$ can support more PDCP and SDAP functions in the Tofino2-based fast path, e.g., transmission buffering, header compression [8], reordering [14, 24], and deduplication [13] while being integrated with the CU-CP over the standardized E1 interface [4] alongside the software OAI CU implementation. Notwithstanding, larger-scale evaluations, and more complex RAN topologies (e.g., more DUs) will be looked into. We believe CUP$_4$ is the necessary step forward towards the 5G 1 ms end-to-end latencies grand ambition.

ACKNOWLEDGEMENT

We thank the reviewers for their invaluable feedback. This research is supported by the National Research Foundation, Singapore, and Infocomm Media Development Authority under its Future Communications Research & Development Programme.

REFERENCES

[1] 3GPP. 2023. TS 29.281 v17.4.0: General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U).

[2] 3GPP. 2023. TS 38.323 v17.4.0: NR; Packet Data Convergence Protocol (PDCP) specification.

[3] 3GPP. 2023. TS 38.331 v17.4.0: NR; Radio Resource Control (RRC); Protocol specification.

[4] 3GPP. 2023. TS 38.401 v17.4.0: NG-RAN; Architecture description.

[5] Tom Barbette, Georgios P. Katsikas, Gerald Q. Maguire, and Dejan Kostić. 2019. RSS++: Load and State-Aware Receive Side Scaling. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. 318–333.

[6] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. 2014. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 87–95.

[7] Intel Corporation. 2023. Intel Tofino 2. https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-2-series.html [Accessed: Mar. 2023].

[8] Mikael Degermark, Carsten Burmeister, Anton Martensson, Thomas Wiebke, Akihiro Miyazaki, Thima Koren, Takeshi Yoshimura, Hideaki Fukushima, Rolf Hakenberg, Khiem Le, Haihong Zheng, Hans Hannu, Zhigang Liu, Krister Svanbro, Lars-Erik Jonsson, and Carsten Bormann. 2001. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. RFC 3095. https://doi.org/10.17487/RFC3095

[9] EURECOM. 2023. OPENAIR-CN-5G: An implementation of the 5G Core network by the OpenAirInterface community. https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed [tag: v1.5.0].

[10] EURECOM. 2023. OpenAirInterface. https://gitlab.eurecom.fr/oai/openairinterface5g [tag: 2023.w12].

[11] Rostand A. K. Fezeu, Eman Ramadan, Wei Ye, Benjamin Minneci, Jack Xie, Arvind Narayanan, Ahmad Hassan, Feng Qian, Zhi-Li Zhang, Jaideep Chandrashekar, and Myungjin Lee. 2023. An In-Depth Measurement Analysis Of 5G MmWave PHY Latency And Its Impact On End-to-End Delay. In *Passive and Active Measurement*. 284–312.

[12] ITU. [n. d.]. ITU-R FAQ on INTERNATIONAL TELECOMMUNICATIONS (IMT). https://www.itu.int/en/ITU-R/Documents/ITU-R-FAQ-IMT.pdf [Accessed: Mar. 2023].

[13] Stefan Johansson. 2021. Packet Deduplication in P4. https://opennetworking.org/wp-content/uploads/2021/05/2021-P4-WS-Stefan-Johansson-Slides.pdf [Accessed: June 2023].

[14] Raj Joshi, Chahwan Song, Xin Zhe Khooi, Nishant Budhdev, Ayush Mishra, Mun Choon Chan, and Ben Leong. 2023. Masking Corruption Packet Losses in Datacenter Networks with Link-local Retransmission. In *Proceedings of the ACM SIGCOMM 2023 Conference*.

[15] Florian Kaltenberger, Aloizio P Silva, Abhimanyu Gosain, Luhan Wang, and Tien-Thinh Nguyen. 2020. OpenAirInterface: Democratizing innovation in the 5G Era. *Computer Networks* 176 (2020), 107284.

[16] Florian Kaltenberger, Hongzhi Wang, and Sakthivel Velumani. 2021. Performance evaluation of offloading LDPC decoding to an FPGA in 5G baseband processing. In *WSA 2021; 25th International ITG Workshop on Smart Antennas*. VDE, 1–4.

[17] Anupa Kelkar and Chris Dick. 2021. Aerial: a GPU hyper-converged platform for 5G. In *Proceedings of the SIGCOMM'21 Poster and Demo Sessions*. 79–81.

[18] Jialin Li, Naveen Kr. Sharma, Dan R. K. Ports, and Steven D. Gribble. 2014. Tales of the Tail: Hardware, OS, and Application-Level Sources of Tail Latency. In *Proceedings of the ACM Symposium on Cloud Computing*. 1–14.

[19] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, Feng Qian, and Zhi-Li Zhang. 2021. A Variegated Look at 5G in the Wild: Performance, Power, and QoE Implications. In *Proceedings of the ACM SIGCOMM 2021 Conference*. 610–625.

[20] Tian Pan, Nianbing Yu, Chenhao Jia, Jianwen Pi, Liang Xu, Yisong Qiao, Zhiguo Li, Kun Liu, Jie Lu, Jianyuan Lu, et al. 2021. Sailfish: Accelerating cloud-scale multi-tenant multi-service gateways with programmable switches. In *Proceedings of the ACM SIGCOMM 2021 Conference*. 194–206.

[21] O-RAN Project. 2022. O-RAN Architecture Overview. https://docs.o-ran-sc.org/en/latest/architecture/architecture.html [Accessed: June 2023].

[22] Quectel. [n. d.]. 5G RM50xQ series. https://www.quectel.com/product/5g-rm50xq-series [Accessed: June 2023].

[23] Ettus Research. [n. d.]. USRP B210 USB Software Defined Radio (SDR). https://www.ettus.com/all-products/ub210-kit/ [Accessed: June 2023].

[24] Chahwan Song, Xin Zhe Khooi, Raj Joshi, Inho Choi, Jialin Li, and Mun Choon Chan. 2023. Network Load Balancing with In-network Reordering Support for RDMA. In *Proceedings of the ACM SIGCOMM 2023 Conference*.

[25] Cuidi Wei, Ahan Kak, Nakjung Choi, and Timothy Wood. 2022. 5GPerf: Profiling Open Source 5G RAN Components under Different Architectural Deployments. In *Proceedings of the ACM SIGCOMM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases*. 43–49.

[26] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of the ACM SIGCOMM 2020 Conference*. 479–494.