PRAVEGA: Scaling Private 5G RAN via eBPF+XDP

Udhaya Kumar Dayalan, Ziyan Wu, Gaurav Gautham, Feng Tian, Zhi-Li Zhang
{dayal007,wu000598,gauta044,tianx399}@umn.edu,zhzhang@cs.umn.edu
Department of Computer Science & Engineering, University of Minnesota - Twin Cities, USA

ABSTRACT

We exploit eBPF+XDP to scale and accelerate software packet processing in (O-RAN compliant) disaggregated 5G RAN (Radio Access Network). We argue that the Central Unit User Plane (CU-UP) component is likely the bottleneck in the 5G RAN user plane data path and therefore focuses on optimizing its performance. We propose an eBPF/XDP-based framework, PRAVEGA, and discuss additional options for further improvements.

CCS CONCEPTS

• **Networks** → **Network layer protocols**; **Network architectures**; **Mobile networks**; **Network design principles**;

KEYWORDS

5G, RAN, 5G Throughput, O-RAN Central Unit, eBPF, XDP

ACM Reference Format:
Udhaya Kumar Dayalan, Ziyan Wu, Gaurav Gautham, Feng Tian, Zhi-Li Zhang. 2023. PRAVEGA: Scaling Private 5G RAN via eBPF+XDP. In *1st Workshop on eBPF and Kernel Extensions (eBPF '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3609021.3609303

1 INTRODUCTION

Unlike previous generations of cellular networks, one distinctive feature of 5G is the adoption of network function virtualization (NFV) and "cloud-native" architecture. For example, 5G core networks are specified as a collection of interactive network functions such as User Plane Function (the sole function of 5G core user/data plane), Access Management Function (AMF) and Session Management Function (SMF) that are part of the 5G core control plane. Furthermore, both 3GPP and the Open-RAN (O-RAN) Alliance have defined a new 5G radio access network (RAN) architecture that *disaggregates* the conventional 5G gNode (gNB) into three components: *Central Unit* (CU), *Distributed Unit* (DU) and *Radio Unit* (RU), with open interfaces[1] for communications among them (as well as with 5G core functions). CU is further split into CU-UP (CU user plane) and CU-CP (CU control plane). We refer the reader to 3GPP specifications [4–6] and O-RAN specifications [12] for details.

The disaggregation of 5G RAN makes it possible to implement most of the 5G RAN protocol stack in software. This is particularly the case for CU, as it performs the upper layers of the 5G RAN protocol stack – Service Data Adaptation Protocol (SDAP) and Packet Data Convergence Protocol (PDCP) – and does not require specialized radio signal processing hardware. Softwarization or "cloudification" of 5G RAN is especially appealing to many industrial use cases, where *private* 5G [7] deployment is common. Software implementation makes it easier to tailor the private 5G deployment to specific use cases, incorporate and upgrade certain features to better support such use cases as needed. However, as running the 5G RAN protocol stack on commodity servers is in general much slower than on dedicated hardware appliances, *scaling (private) 5G RAN software implementation poses a key challenge*, especially for use cases that require a large number of simultaneous connections and high bandwidths such as industrial IoT (Internet of Things) and video surveillance systems for massive warehousing.

In this short workshop paper we explore the feasibility and benefits of leveraging eBPF+XDP in scaling CU-UP. We focus on CU-UP as each CU-UP may connect to multiple UPFs and multiple DUs, and must process downlink packets from the 5G core network (UPFs) to user equipment (UE) and uplink packets from UE to the 5G core network quickly to meet the bandwidth demand and minimize latency. As the connections between CU-UP and UPF and between CU-UP and DU are Ethernet-based, we can exploit eBPF (extended Berkeley Packet Filter) [1, 13] and XDP (eXpress Data Path) [8] for kernel extension/bypassing and packet processing optimization. The **key contributions** of our paper are summarized below.

• We advance an initial design of a scalable (private) 5G RAN architecture using eBPF+XDP, focusing particularly on CU-UP.

• We also discuss additional options to further accelerate packet processing to scale 5G RAN implementation to meet bandwidth and latency demands. While our proposed designs are applicable to public 5G RAN, PRAVEGA is intended for private 5G deployments to meet the needs for customization and ease-of-upgrade.

2 PRAVEGA DESIGN

In this section, we present the initial design of PRAVEGA.

2.1 Kernel Based CU-UP

In this design, the GTP-U packets are completely handled in the kernel space without being sent to the user space. Fig. 1 illustrates the proposed design for the pure kernel based CU-UP. The Data Path Layer are configured by the Management Layer using eBPF Maps [9]. The detailed workflow of this design is shown in Fig. 2. The key components of this design are discussed below:

Parser. The *'Parser'* component gets called for each packet received in the NIC. *'Parser'* filters and send GTP-U user plane packets to *'Classifier'* and pass the other type of packets to the network stack.

Classifier. The *'Classifier'* which processes the SDAP layer, retrieves the PDU session and QFI value from each packet. Next, it validates the PDU session information and assign the respective

[1] CUs, DUs and RUs that support O-RAN standard interfaces are often referred to as O-CU, O-DU and O-RU.

eBPF '23, September 10, 2023, New York, NY, USA

ACM ISBN 979-8-4007-0293-8/23/09.
https://doi.org/10.1145/3609021.3609303

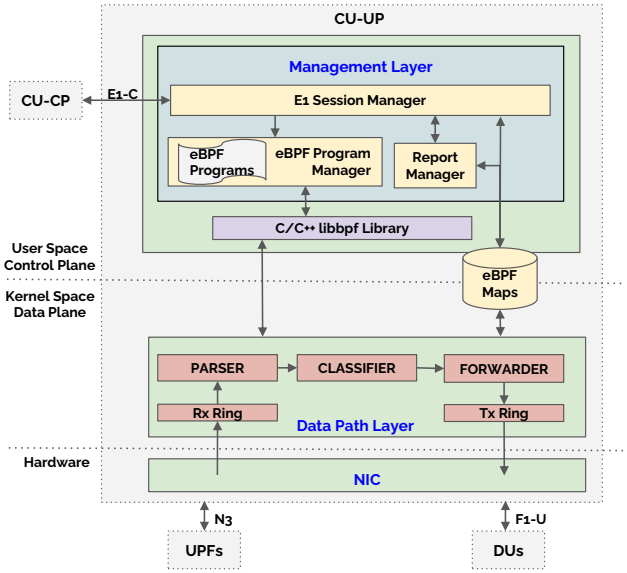

Figure 1: Pure Kernel eBPF Based CU-UP

DRB for each packet based on the QFI-DRB Mapping. Finally, it forwards the packet to the *'Forwarder'*.

Forwarder. The *'Forwarder'* which processes the PDCP layer, consists of a set of DRBs per UE. Each packet passes through the respective DRB undergoing integrity protection, ciphering and sequence numbering. Finally, it routes the modified GTP-U packet to the respective DU based on the established PDU session.

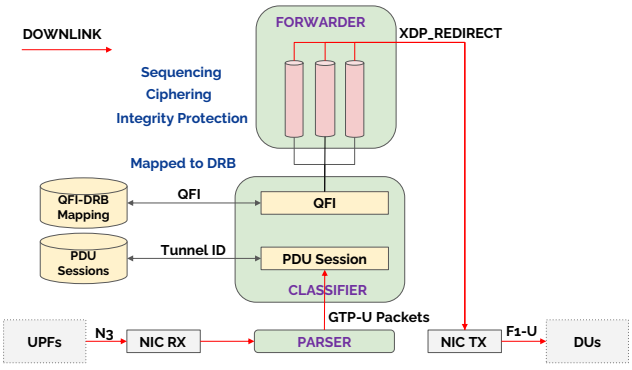

Figure 2: Pure Kernel eBPF Based CU-UP Flow - Downlink

2.2 Offloading Ciphering/Deciphering

Ciphering/Deciphering are optional CPU-intensive operations in the PDCP layer. For example, voice-traffic in the mobile network may be subject to ciphering while non-sensitive data do not require ciphering for the optimization of network resources and latency. The ciphering algorithms such as SNOW 3G and ZUC [2, 3] are CPU-intensive and can be a bottleneck if not designed carefully.

To further mitigate the overhead of the CPU-intensive operations, we use Bluefield 2 (BF2) [11] to offload expensive ciphering/deciphering operations. The BF2 is comprised of two key components: Mellanox ConnectX6-based switching data plane and 8

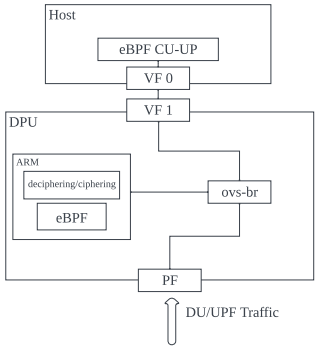

Figure 3: Offloading Cryptic Operations to SmartNICs

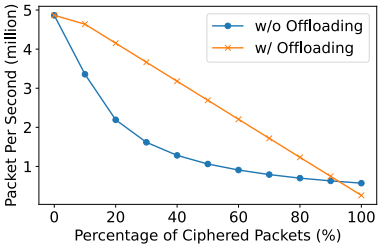

Figure 4: Evaluation on Offloading Ciphering with BF2

Arm A72 cores that can achieve speeds upto 2.75GHz. With the ConnectX6 handling packet reception from network ports, it can either directly transfer them to the host or re-route them to the Arm CPUs. As shown in our design Fig. 3, we use different flow steering policies configuring through the ASAP OVS offloading [10] for flows with/without ciphering/deciphering needs. For flows that need crypto operations, the flows will be steered to eBPF program running in the ARM cores in the Bluefield to save cycles of the CPU in the hosts. Otherwise, the flows will be sent to the network stack.

To illustrate the benefits, we implement a prototype with the offloading scheme, with an experiment that playing a number of flows with a varying proportion (P_E) of encrypted traffics that compares against the one without offloading. As shown in Fig. 4, the version without the offloading scheme,the throughput goes down as P_E increases because of the increased CPU usage takes up the resources for processing packets in a timely manner, resulting in dropped packets. In the offloading scheme, we can see the performance greatly increased because of the more CPU cycles spent on packet processing instead of ciphering, although for the case that P_E = 100% the SmartNIC itself becomes a bottleneck.

3 CONCLUSION & FUTURE WORK

In this paper, we designed PRAVEGA, an eBPF/XDP based fast processing CU-UP. We have also discussed other potential design options. Besides further optimization on offloading to the SmartNICs, we will explore techniques using dedicated cryptographic hardware and further incorporation on other O-RAN components.

ACKNOWLEDGMENTS

This research was supported in part by the NSF under Grants CNS-1814322, CNS-1836772, CNS-1831140, CNS-1901103, CNS-2106771 and CNS-2128489, and Seed Grants from University of Minnesota MnRI, CTS and CSE and an unrestricted gift from InterDigital.

REFERENCES

[1] 2023. eBPF - Introduction, Tutorials Community Resources. (2023). Retrieved March 2023 from https://ebpf.io/

[2] 3rd Generation Partnership Project. 2006. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 UIA2. Document 2: SNOW 3G Specification. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2396. (2006).

[3] 3rd Generation Partnership Project. 2011. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 128-EIA3. Document 2: ZUC Specification. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2400. (2011).

[4] 3rd Generation Partnership Project. 2020. Release 15. https://www.3gpp.org/release-15. (April 2020).

[5] 3rd Generation Partnership Project. 2020. Release 16. https://www.3gpp.org/release-16. (July 2020).

[6] 3rd Generation Partnership Project. 2021. Release 17. https://www.3gpp.org/release-17. (March 2021).

[7] Cisco. 2023. Cisco Private 5G Solution Overview. (2023). Retrieved May 2023 from https://www.cisco.com/c/en/us/products/collateral/wireless/private-5g/private-5g-service-so.html

[8] Toke Høiland-Jørgensen, Jesper Dangaard Brouer, Daniel Borkmann, John Fastabend, Tom Herbert, David Ahern, and David Miller. 2018. The EXpress Data Path: Fast Programmable Packet Processing in the Operating System Kernel. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '18)*. Association for Computing Machinery, New York, NY, USA, 54–66. https://doi.org/10.1145/3281411.3281443

[9] Prototype Kernel. 2023. eBPF maps. (2023). Retrieved April 2023 from https://prototype-kernel.readthedocs.io/en/latest/bpf/ebpf_maps.html

[10] NVIDIA. [n. d.]. ASAP OVS Offload - NVIDIA Support for TripleO Ussuri Application Notes - NVIDIA Networking Docs. ([n. d.]). Retrieved May 2023 from https://docs.nvidia.com/networking/display/TAN10/ASAP+OVS+Offload

[11] NVIDIA. [n. d.]. BlueField Data Processing Units (DPUs) | NVIDIA. ([n. d.]). Retrieved May 2023 from https://www.nvidia.com/en-us/networking/products/data-processing-unit/

[12] O-RAN Alliance. 2023. O-RAN Alliance. (2023). Retrieved March 2023 from https://www.o-ran.org/

[13] Marcos A. M. Vieira, Matheus S. Castanho, Racyus D. G. Pacífico, Elerson R. S. Santos, Eduardo P. M. Câmara Júnior, and Luiz F. M. Vieira. 2020. Fast Packet Processing with EBPF and XDP: Concepts, Code, Challenges, and Applications. *ACM Comput. Surv.* 53, 1, Article 16 (feb 2020), 36 pages. https://doi.org/10.1145/3371038