A 100Gbps User Plane Function Prototype Based on Programmable Switch for 5G Network

Cong Zhou, Baokang Zhao, Baosheng Wang
Department of Computer, National University of Defense Technology
ChangSha, China
cleverchou@126.com,bkzhao@nudt.edu.cn,wbs@nudt.edu.cn

ABSTRACT

User Plane Function (UPF) play crucial role in 5G core network, it is of vital importance to improve the UPF performance. In this demo, we provide novel UPF architecture with a programmable switch. The implemented UPF prototype is able to provide up to 100Gbps matching throughput within 1 microsecond latency.

1 INTRODUCTION

Growing 5G applications require a user plane that has a high forward throughput and low packets loss.The User Plane Function (UPF) handles the critical data plane processing of packets between base station and data center, including access control, GTP tunnel encapsulation and decapsulation, forwarding of packet to/from packet data net, maximum bit rate, guaranteed bit rate, per-flow Qos, etc.

UPF play important role in 5G core network, and how to accelerate 5G core networks causes concern. C.A.Shen [4] designs and implements offloading GTP engine realized on the FPGA platform; A.Bose [1] designs and implements several standards compliant UPF prototypes, including DPDK-based UPF; hardware-based offload function UPF and also private packet steering and control plane offload. VPP is also used as an accelerator [5], where DPDK has been achieved for GTP-u tunnel; C.Lee [2] designs SRV6 platform for GPT-u encapsulating/decapsulating function; F.Paolucci [3] implements UPF offload in a P4 switch running on the Behavioral Model version 2 (BMv2).

Above of solutions improved the UPF performance from the view of software-hardware and physical-simulation, however, there is less architecture to integrate them, so that existing work cannot be inherited. In order to break through the above limitations, Our previous work provides a novel architecture, namely CeUPF [6] , to finish software-hardware integration. In this demo, we implement a UPF prototype which is able to provide up to 100Gbps matching throughput within less latency. We also demonstrate the high performance of the prototype with multiple rules and users.

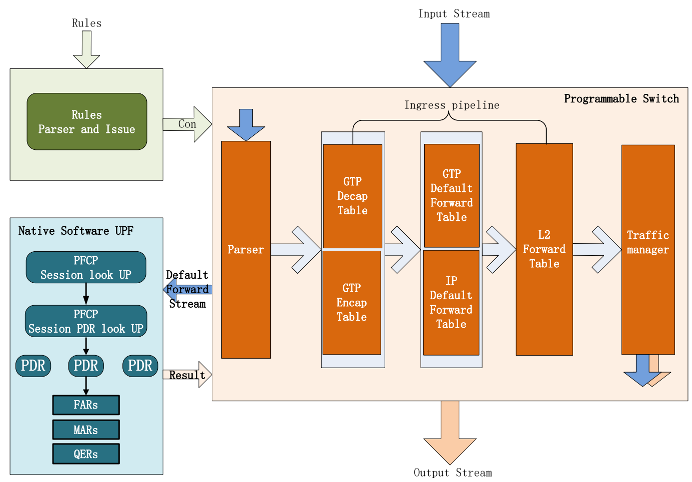

Figure 1: the UPF system architecture

2 UPF SYSTEM ARCHITECTURE

Fig.1 depicts the overview of our implemented UPF system architecture. This demo consist of three parts. a) Rules Parser and Issue module (RPI) connect to programmable switches by the serial port; b) Native Software UPF connects to programmable switches by two QSFP optical fiber port, which is an open source projects for 5G core network, named free5GC, implementing the 5G core networks function defined in 3GPP Release 15; c) Programmable Switch connects base station and data center with other QSFP optical fiber port.

RPI provides a CLI interface for users, parsing user-defined rules, generating legal items, and sending them to the programmable switch. This demo offloads GPT-u encapsulating/decapsulating and forwarding functions into programmable switch. In parser phrase, the programmable switch extracts the header of datagram, including outer source IP, outer destination IP, inner source IP, inner source IP, Tid, then state diagram is constructed. There are five table in the ingress pipeline, GTP default forward table is used for forwarding GTP-u packet to native software UPF and IP default forward table is for IP packet; GTP encap table is used for adding GTP-u header for coming IP packet; GTP decap table is to delete the GTP-u header; L2 forward table is using for changing source MAC address when GTP encap/decap table has applied, then forwarding.

It note that the rule is which packet can be forwarded through programmable switch, and the rules can be defined by matching packet headers. Firstly, programmable switch has no rules, and all of the input stream hit IP/GTP Default Forward table, and sent to native software UPF , are processed as usual. Then these result was back to programmable switch and forwarded. There is a special case, when has no rules to match, it perform the same process. Secondly, enabling programmable switch through loading rule dynamically. Once the rules have been configured, programmable switch becomes an efficient forwarding component. Input packets hit the GTP Encap/Decap Table, perform the corresponding action, then forwarded.

3 EXPERIENCE VALIDATION

We uses Spirent TestCenter as traffic generator, which has 2*100G interfaces, and it can simulate standard 5G GTP-u data-gram. The packet size is set 64B, 128B, 256B, 1024B and 1058B respectively following RFC-2544. We continue 60S to send packets in each fixed size.

In programmable switch, the average throughput of GTP-u encapsulation is 86Gbps, because encapsulation will increase GTP-u headers, it is unable to use 100G line speed, meantime the average latency is 0.93us; In native software UPF, the corresponding value are 350Mbps and 59.03us. When validating the decapsulation, in programmable switch, the average throughput 100Gbps and the average latency 0.94us; In native software UPF, the corresponding value are 550Mbps and 70.01us. The various of size validation results are shown in Fig. 2.

The above tests are for 1 user - 1 rule, we design other scenario by increased the number of user and rules, namely, 1 user - 1K rules and 100 users - 1K rules , finally, the metrics results are similar. That is because of multistage pipeline, the programmable switches performance is less affected as user or rules increasing.

Lastly, we also validated this demo functionality using UERANSIM and real mobile phone, the results show that this demo works normally.

4 CONCLUSION

This demo proposed a novel UPF architecture for integrating different types of acceleration schemes, where GTP-u encapsulation/decapsulation and forwarding function were offloaded into programmable switches. The Experience show that this demo has a throughput of 70-100Gbps and a delay of <1 us dealing with the standard 5G data-gram. For the multi users - multi rules scenario, it is also applicable and the function is feasible in the real environment.

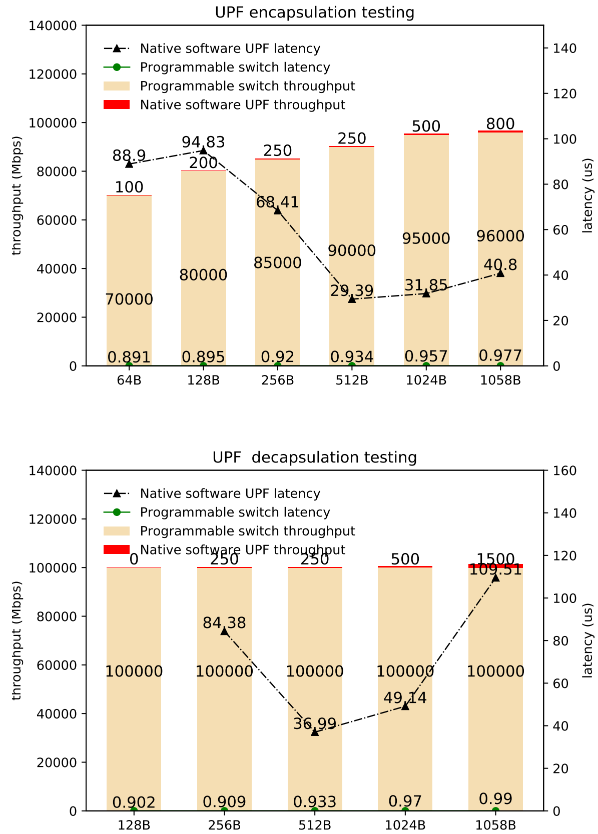

Figure 2: the UPF system validation

REFERENCES

[1] Bose Abhik, Maji Diptyaroop, Agarwal Prateek, Unhale Nilesh, Shah Rinku, and Vutukuru Mythili. 2021. Leveraging Programmable Dataplanes for a High Performance 5G User Plane Function. *APNet* (2021).

[2] Chunghan Lee, Kentaro Ebisawa, Hitoshi Kuwata, Miya Kohno, and Satoru Matsushima. 2019. Performance Evaluation of GTP-U and SRv6 Stateless Translation. *15th International Conference on Network and Service Management* (2019), 1–6.

[3] Francesco Paolucci, Davide Scano, Filippo Cugini, Andrea Sgambelluri, Luca Valcarenghi, Carlo Cavazzoni, Giuseppe Ferraris, and Piero Castoldi. 2021. User Plane Function Offloading in P4 switches for enhanced 5G Mobile Edge Computing. (2021), 1–3.

[4] ChungAn Shen, DingYuan Lee, ChungAn Ku, MengWei Lin, KuoCheng Lu, and ShaoYu Tan. 2019. A Programmable and FPGA-accelerated GTP Offloading Engine for Mobile Edge Computing in 5G Networks. *IEEE Conference on Computer Communications Workshops* (2019), 1021–1022.

[5] VPP. 2021. https://hub.fastgit.org/FDio/vpp. (2021).

[6] Cong Zhou, Baokang Zhao, Baosheng Wang, and Yulei Yuan. 2021. CeUPF: Offloading 5G User Plane Function to Programmable Hardware Base on Co-existence Architecture. *2021 ACM International Conference on Intelligent Computing and its Emerging Applications* (2021), 34–39.