TBD: Both In-Kernel and Bypass are needed - A Practical Approach to Low-Latency Virtualized Network Functions

Zuo Xiang*, Frank Gabriel*, Giang T. Nguyen* †, Zhongyi Fan*, and Frank H. P. Fitzek*
*Deutsche Telekom Chair of Communication Networks
†SFB-912 HAEC, Technische Universität Dresden, Germany
Email: {firstname.lastname}@tu-dresden.de

I. Sketch

MARK: Blue is used for hint and comments. [Red is used for issues, problems.]

Storyline: ← we keep the sketch of the storyline currently here only to have a overall view for the draft. This part will be deleted later on

- Introduction
  - 3-4 paragraphs (Broad topic, problem, solution and contributions)
  - Motivation
    - Latency is critical in the next generation of communication networks (e.g. for Tactile Internet)
    - To be solved Problem How to implement low latency VNFs in Virtual Machine(Containers are not considered in this work)?
      1) In kernel space or in user space? Pros and cons?
      2) Which frameworks should be used? Pros and Cons should be analyzed in a separate section.
      3) How to enable low latency without decreasing other features and performance parameters too much? e.g. Flexibility, Scalability for multiple VNFs. Bandwidth, hardware resource usage(related to energy consumption). The chain-based approach proposed in this paper can better meet these requirements.
- Related Work
  - Packet IO frameworks related papers
    - User space: High performance Packet IO
    - Kernel space: XDP and eBPF in practice(2 papers are found).
  - Virtualed Network Functions Approaches
    - Vitualized Network Coding on the Internet. This paper implement NC with DPDK KNI for flexibility.
    - TODO: Find other approaches to achieve low latency VNFs
- Low latency VNF
  - Compare different frameworks and explain why XDP and DPDK are finally chosen for further implementation.
  - Compare chain-based approach and KNI-based approach. Describe OVS-DPDK is the tech which reduce the latency overhead introduced by the chain-based approach, the context switching between user- and kernel space is the main issue that increase the latency with KNI approach

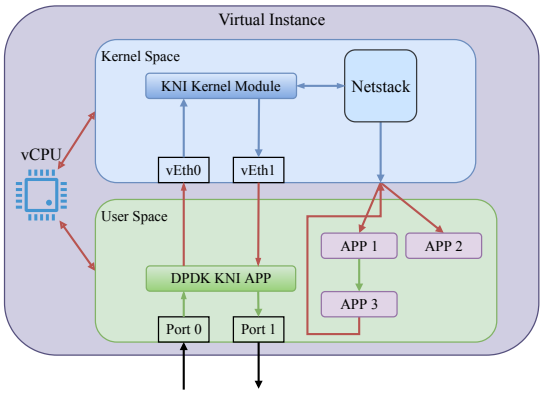

Fig. 1: KNI Approach

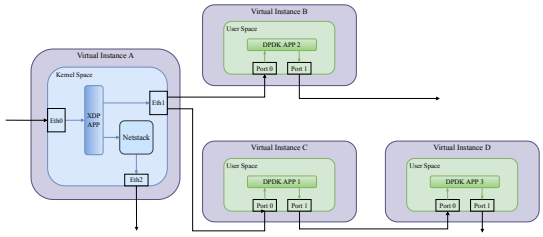

Fig. 2: Chain Approach

  - Describe how to enable our approach on the OpenStack cloud platform. Use SFC, enable multi-queue feature of the Nova.
- Measurement Results
  - End-to-End latency of KNI and XDP+DPDK (1. 1 core for all; 2 cores to make them fair)

  - Bandwidth of KNI and XDP+DPDK. (Different burst size can be a variable here.)
  - CPU usage of KNI and XDP+DPDK on the physical node. Nova compute node

Approach	User	System	IO-Wait	Guest	Total
DPDK KNI 1 vCPU	25.21	23.35	0.04	1.70	50.3
DPDK KNI 2 vCPU	25.23	47.18	0.04	2.88	75.33
XDP + DPDK	25.24	22.97	0.04	2.18	50.43

TABLE I: CPU usage of the compute node (in percent)

- Conclusion
  - If latency counts, our approach is better.
- Future Work
  - Extend SFC-extension to support fully chain-based approach on OpenStack.

ACKNOWLEDGMENT

This work is supported in part by the German Research Foundation (DFG) within the Collaborative Research Center SFB 912 (HAEC).

REFERENCES