Considerations for Benchmarking Network Performance in Containerized Infrastructure

draft-dcn-bmwg-containerized-infra-09

Minh-Ngoc Tran, Jangwon Lee, Younghan Kim (Soongsil University), Kyoungjae Sun (ETRI), Huynsik Yang (KT),

Draft purpose

- This draft aims to provide additional considerations as specifications to guide containerized infrastructure benchmarking, compared with previous benchmarking methodology of common NFV infrastructure

- The additional considerations include:
  - **Additional deployment scenarios**
  - **Additional configuration parameters**
  - Investigation of **different container networking models** based on the usage of vSwitch and different packet acceleration techniques
  - Investigation of **different deployment settings** (NUMA, hugepages, etc.) **that might make performance impacts** on network performance

Updates Summary (from v8 to v9)

- To increase the draft cohesion and clearly show the purpose of the draft (benchmarking considerations)
- New Benchmarking Considerations section which consists of:
  - Additional Deployment Scenarios (previously in Section3. Containerized Infrastructure Overview)
  - Additional Configuration Parameters of Containerized Infrastructure Benchmarking (Completely new)
  - Networking Models based on usage of vSwitch and acceleration techniques (previous Section 4 + updated eBPF section)
  - Performance Impacts settings (previous Section 5)

version 08

CURRENT - version 09

Update AFXDP

Detailed Updates (1)

Benchmarking Consideration 1

Additional Deployment Scenarios

- Previously inside the *"Containerized Infrastructure Overview"* section
- ETSI-TST-009 defined scenario:
  - BMP2BMP (bare metal container/pod to container/pod)
- 2 proposed additional scenarios:
  - BMP2VMP (baremetal – on VM)
  - VMP2VMP (on VM – on VM)

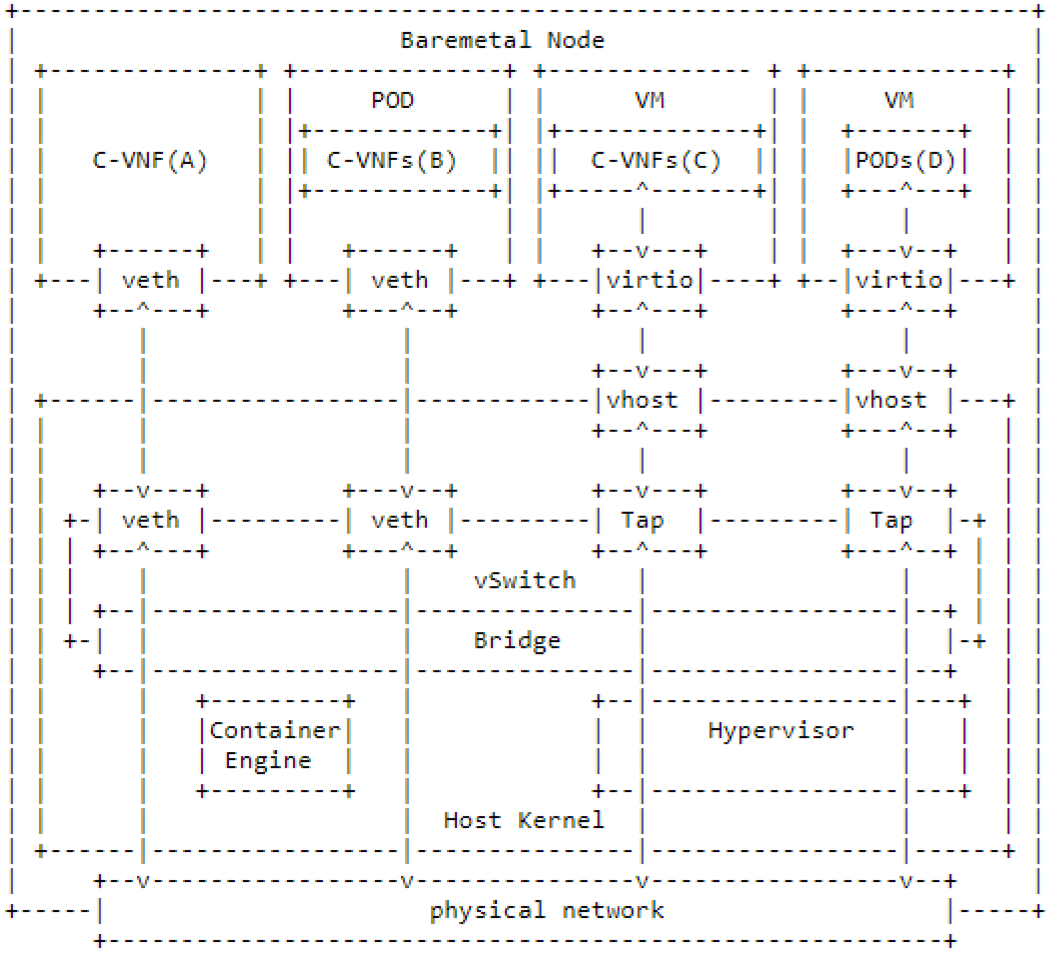


Figure 1: Examples of Networking Architecture based on Deployment Models - (A)C-VNF on Baremetal (B)Pod on Baremetal(BMP) (C)C-VNF on VM (D)Pod on VM(VMP)

Detailed Updates (2)

Benchmarking Consideration 2

Additional Configuration Parameters

- New section
- List of additional parameters for containerized infrastructure
  - Selected Container Runtime
  - Selected Container Network Plugin
  - Selected Packet Acceleration Networking Model
  - Number of C-VNF
  - Memory, NUMA allocation to C-VNF

Detailed Updates (3)

Benchmarking Consideration 3

Networking Models

- Update eBPF Acceleration Model explanation with AFXDP deployment option
- 2 deployment options:
  - XDP hook at NIC - AFXDP: new linux socket that allows a bypass-kernel path
    - Used by: Supported AFXDP vSwitch, Cloud Native Data Plane (CNDP)
  - XDP hook at NIC – traffic control (tc) hook: configured by BPF programs
    - Used by Cilium CNI

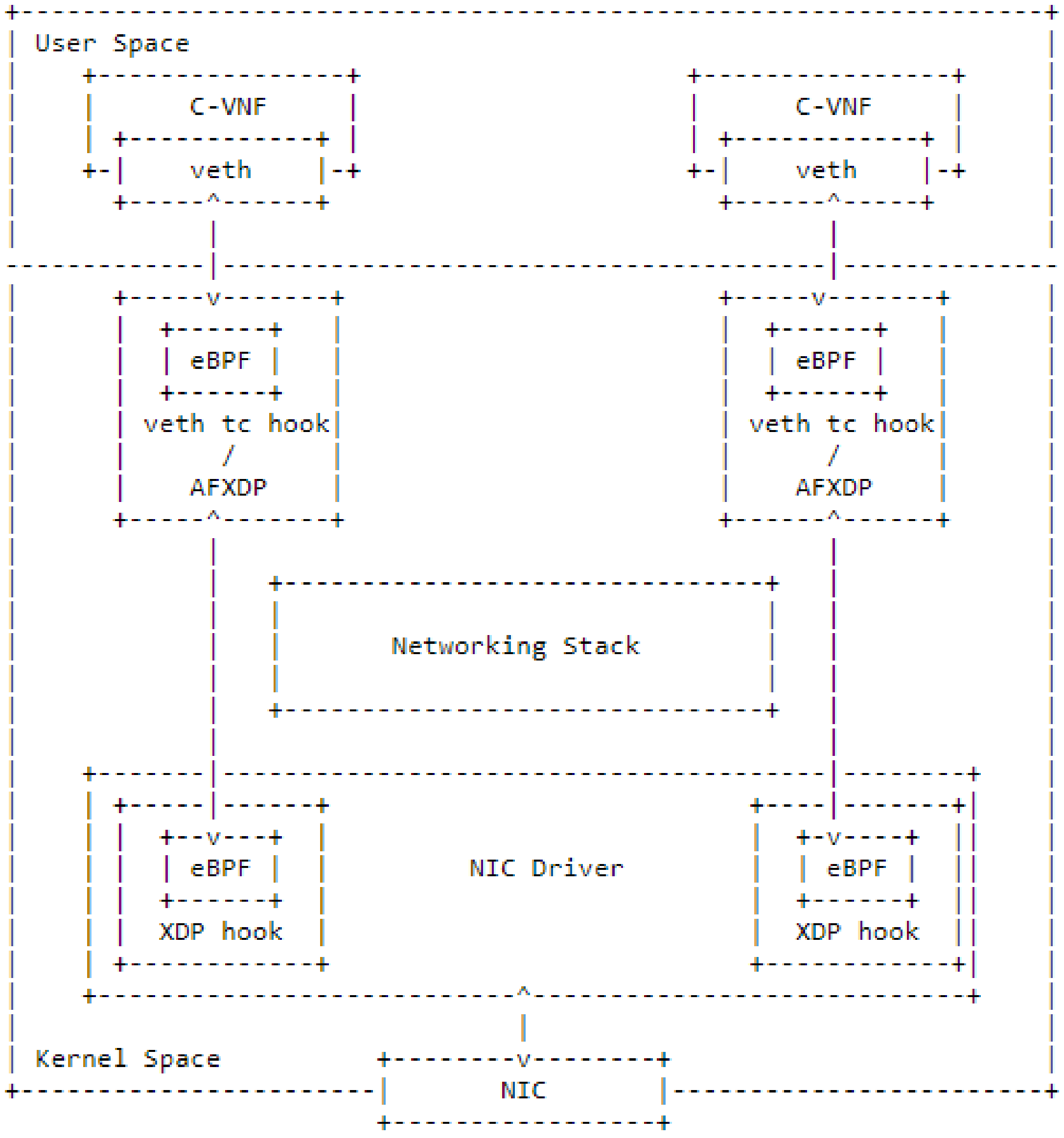


Figure 6: Examples of eBPF Acceleration Model

Detailed Updates (4)

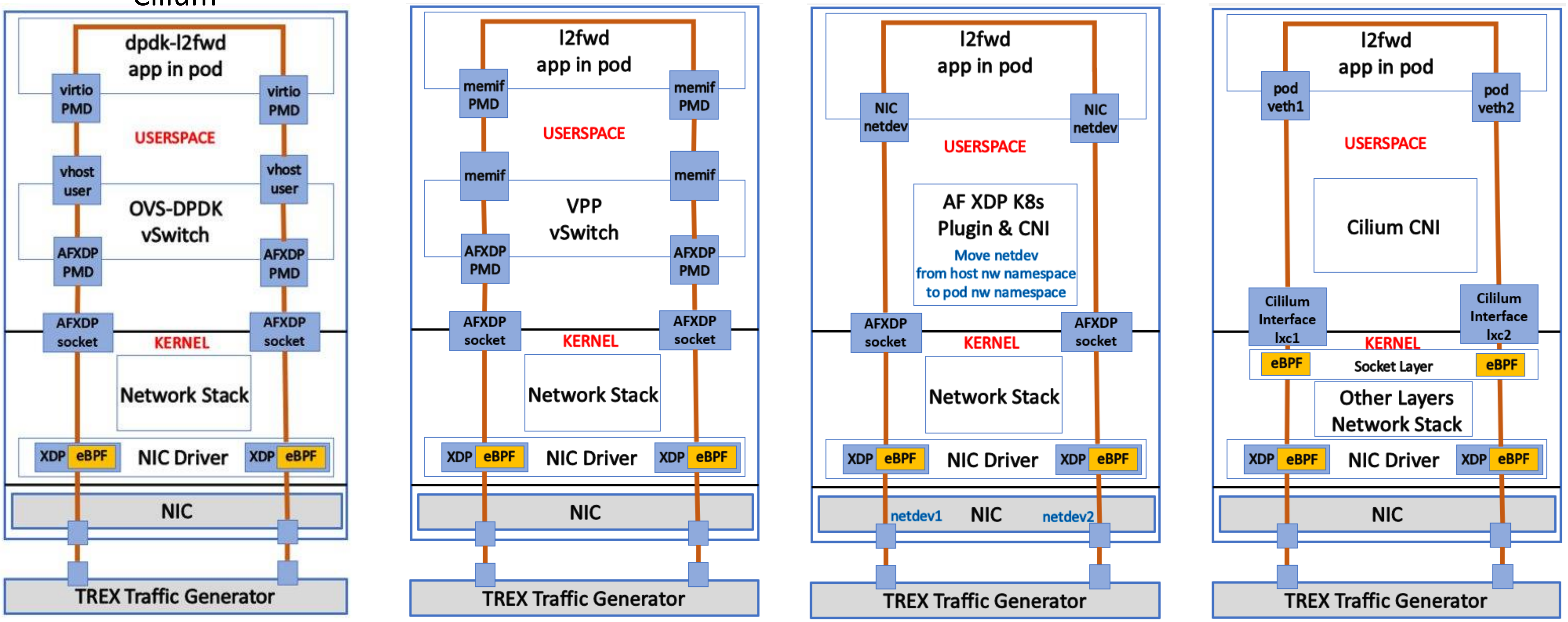
Benchmarking Consideration 4

Performance Impacts

- No changes
- Just move this section into the new Benchmarking Consideration section

From Hackathon 114-115

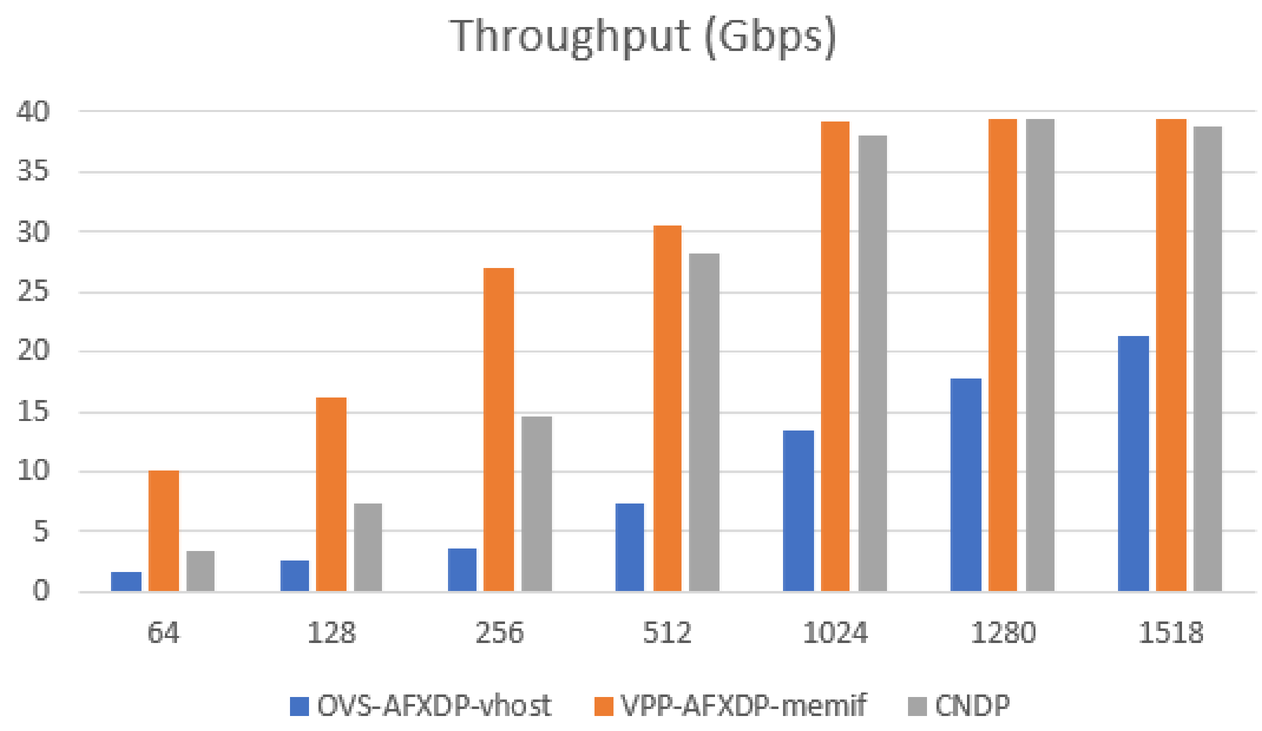
- eBPF Acceleration Model Benchmarking (4 variations)
  - OVS-vhost, VPP-memif vSwitch (AFXDP supported version),
  - Intel Cloud Native Data Plane – CNDP,
  - Cilium



From Hackathon 114-115

- **Benchmarking Performance Results – eBPF Acceleration Models**

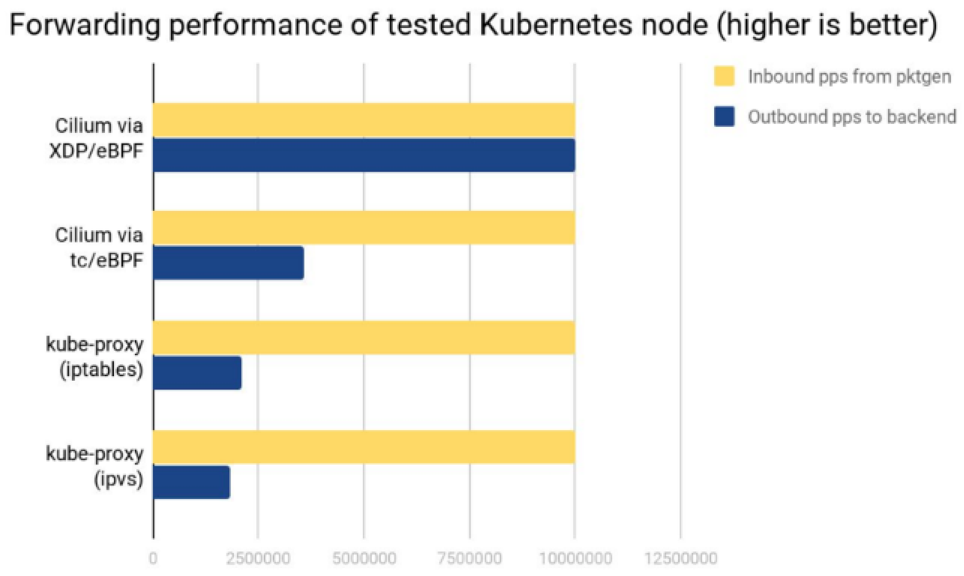
1. VPP-AFXDP outperforms OVS-AFXDP because of memif (shared memory interface) support advantage vs vhost
2. Userspace vSwitch using AFXDP poll mode driver can achieve similar performance vs using DPDK poll mode driver
3. Intel CNDP (poll packets from AFXDP socket to pods by moving netdev from hostname space to pod namespace) can catch up VPP-AFXDP performance with larger size packets (>512)



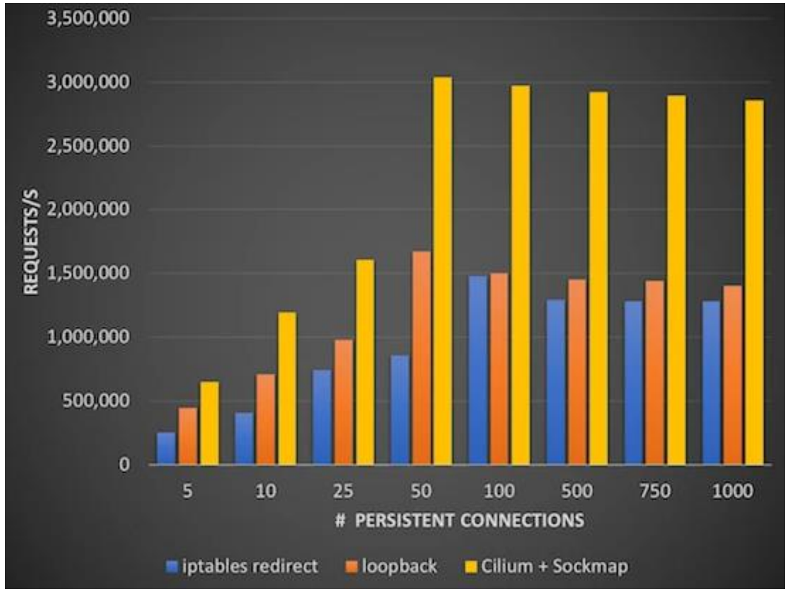
From Hackathon 114-115

- **Benchmarking Performance Results – eBPF Acceleration Models**

4. Cilium use eBPF to accelerate both North-South and East-West traffic (AFXDP: only North-South, East-West via userspace vSwitch)
   Cilium Performance can be referred from Cilium's own benchmarking results
   - North-South: "Cilium 1.8 Release Blog" (https://cilium.io/blog/2020/06/22/cilium-18/)
   - East-West: "Istio 1.0: How Cilium enhances Istio with socket-aware BPF programs" (https://cilium.io/blog/2018/08/07/istio-10-cilium/)



North-South XDP acceleration vs kernel kube-proxy



East-West eBPF socket layer acceleration vs kernel kube-proxy

Next Steps – Request for feedback

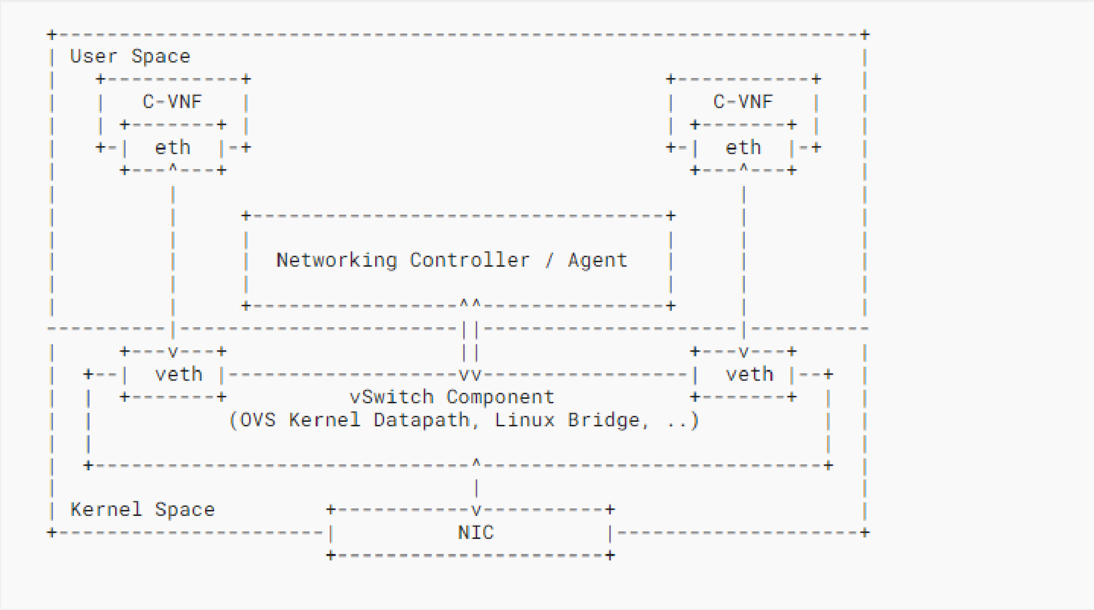
- IETF 115 Hackathon has completed our investigation activities for all proposed benchmarking considerations for containerized infrastructure in our draft.
- We would like to hear any questions and comments from anyone in BMWG that is interested in our draft.
- We will finalize the draft and would like to gather reviews for WG adoption.

Backup Slides

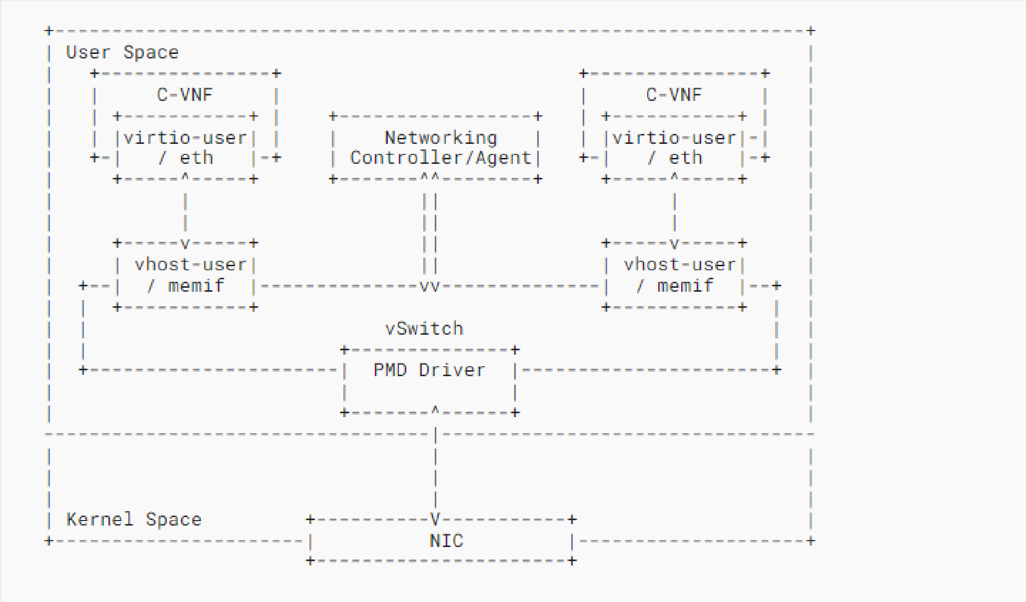
Networking Models

- Kernel-space vSwitch
- User-space vSwitch
- eBPF Acceleration Model
- Smart-NIC Acceleration Model
- Model Combination

4.3.1. Kernel-space vSwitch Model



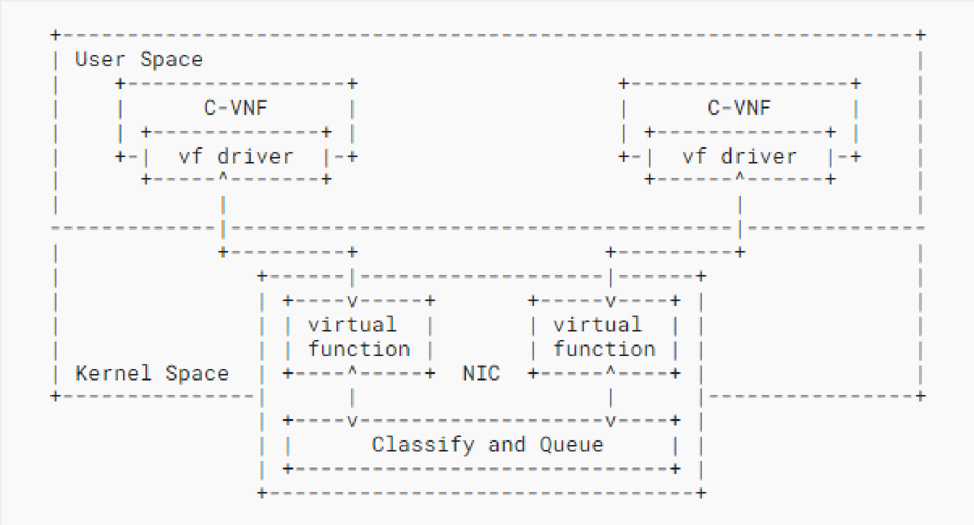
4.3.2. User-space vSwitch Model



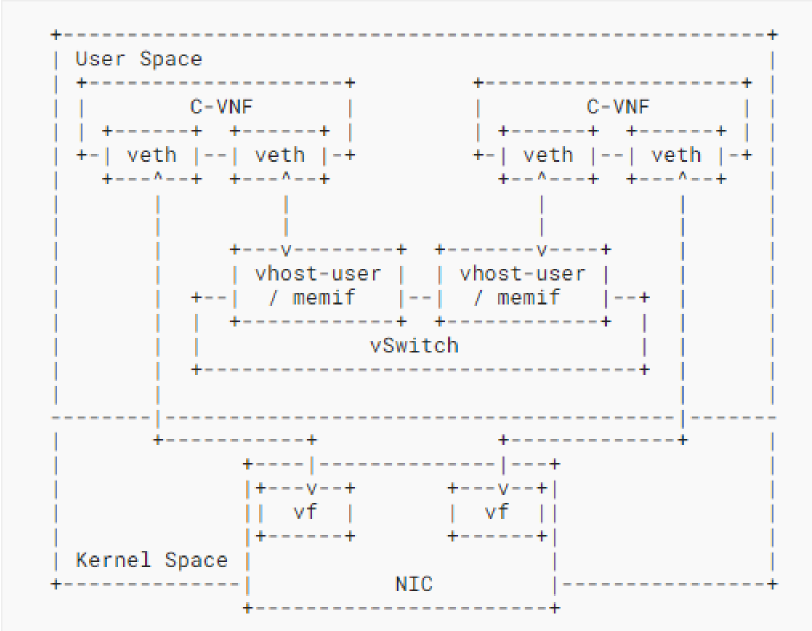
Networking Models

- Kernel-space vSwitch
- User-space vSwitch
- eBPF Acceleration Model
- Smart-NIC Acceleration Model
- Model Combination

4.3.4. Smart-NIC Acceleration Model



4.3.5. Model Combination

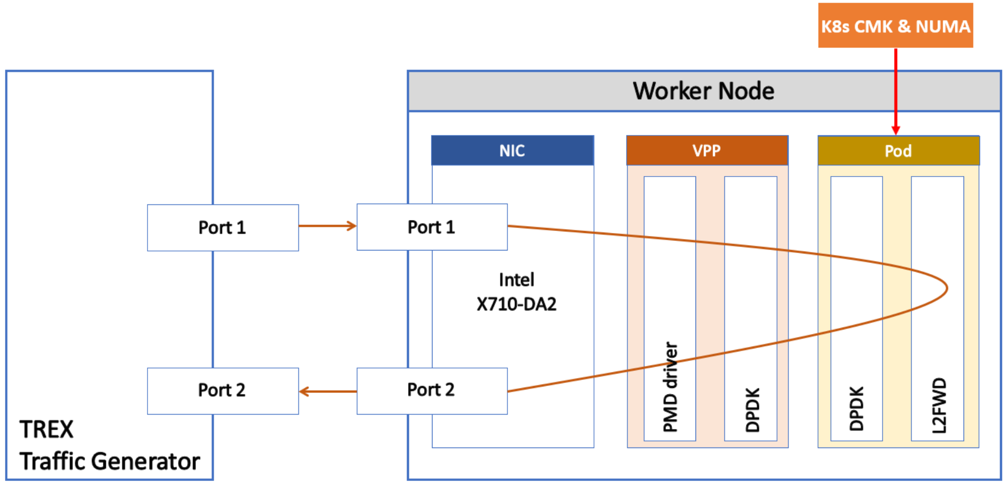


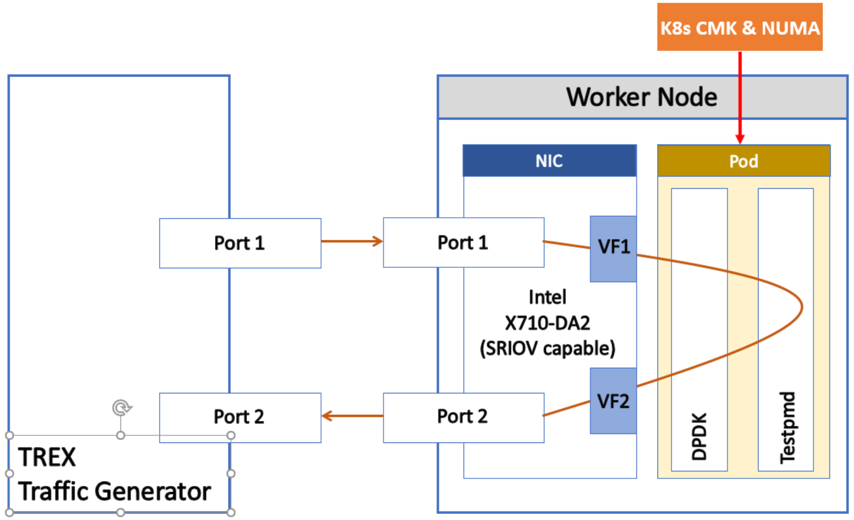
Performance Impacts

- Hugepages
- NUMA & CPU Isolation
- Service Function Chaining
  - In NFV environment, physical network port is commonly connected to multiple VNFs rather than single VNF
  - Aspects needed to be considered when benchmarking service function changing
    - Number of VNFs
    - Different network acceleration technologies (which provide VNF to VNF networking)
- Inter-node networking
  - As defined in ETSI-NFV-IFA-038, different inter-node networking technologies may affect container network performance between nodes
    - Tunnel end point (VXLAN), Border Gateway Protocol (BGP), Layer 2 underlay, direct using dedicated NIC, load balancer.

Benchmarking Experiences (Contiv-VPP + SRIOV)

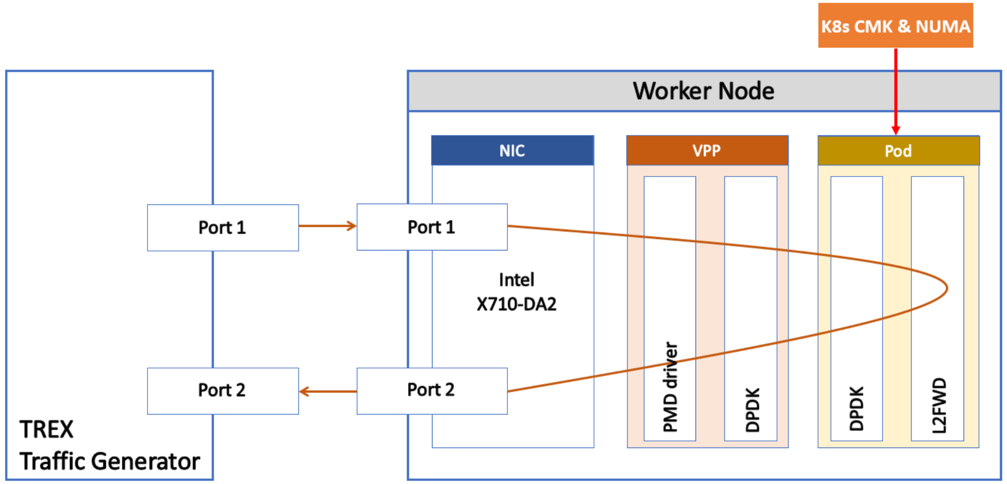
- Test performance of user-space based model and SmartNIC (VPP and SRIOV)
- Figure out impact of CPU isolation (using CMK – CPU Manager for Kubernetes) and NUMA to network performance
  - Without CMK
  - CMK-shared mode (2 pods share 2 CPUs)
  - CMK-exclusive mode (1 dedicated CPU/pod)

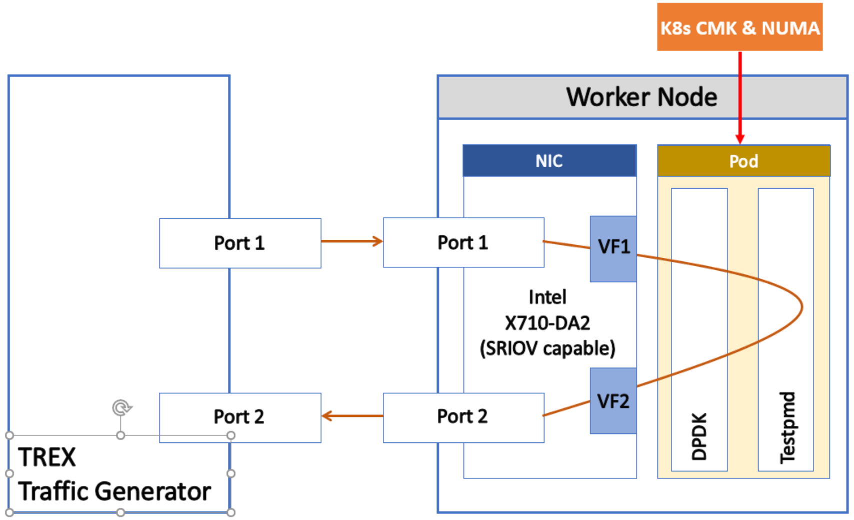




Benchmarking Experiences (Contiv-VPP + SRIOV)

- Test performance of user-space based model and SmartNIC (VPP and SRIOV)
- Figure out impact of CPU isolation (using CMK – CPU Manager for Kubernetes) and NUMA to network performance
  - Without CMK
  - CMK-shared mode (2 pods share 2 CPUs)
  - CMK-exclusive mode (1 dedicated CPU/pod)





Benchmarking Experiences (Contiv-VPP + SRIOV)

What we learned

- VPP and SRIOV has nearly the same performance

CPU Isolation:

- CPU Isolation (CMK) significantly improves throughput
- Exclusive mode is better than Shared mode

NUMA alignment:

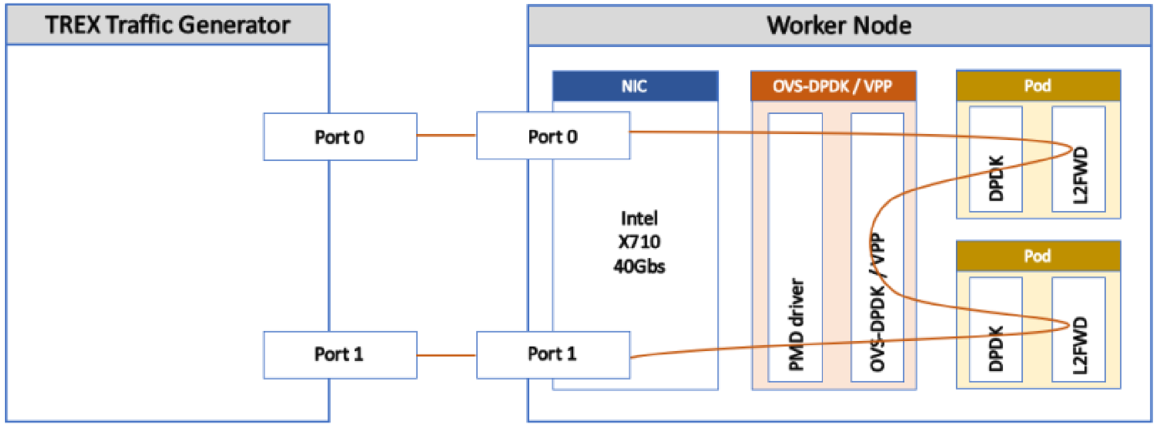
- Assigning CPU in the same NUMA node is better than in different NUMA nodes

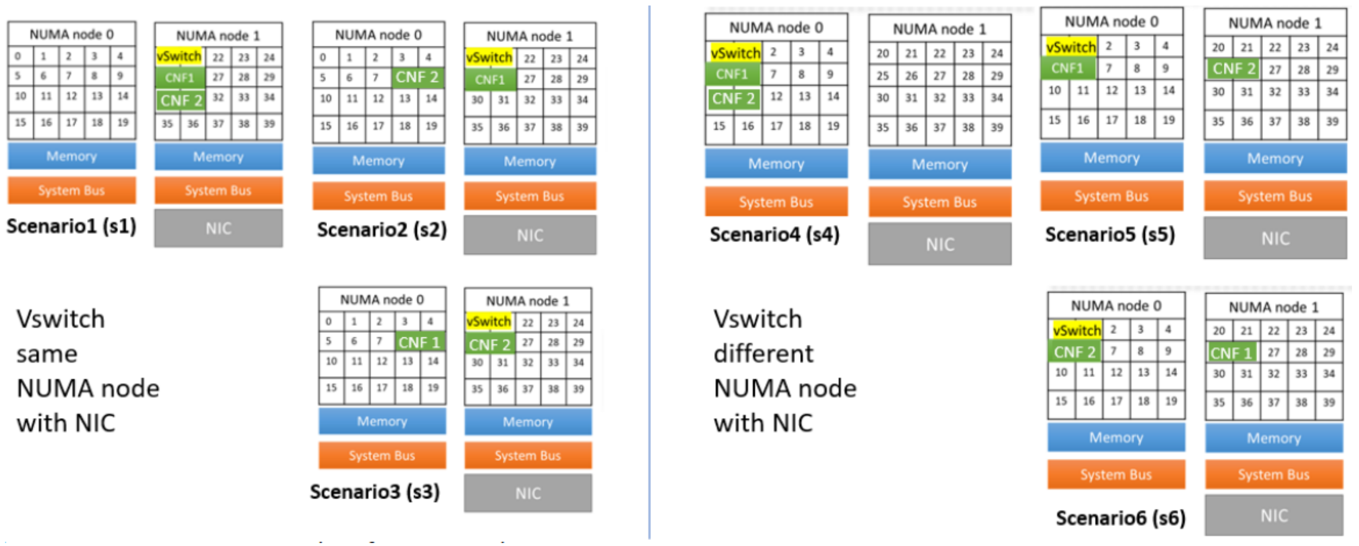
Model	NUMA Mode (pinning)	Result(Gbps)
Maximum Line Rate	N/A	3.1
	same NUMA	9.8
Without CMK	N/A	1.5
CMK-Exclusive Mode	same NUMA	4.7
	Different NUMA	3.1
CMK-shared Mode	same NUMA	3.5
	Different NUMA	2.3

CPU Isolation and NUMA location impact in VPP test with 10G Intel X710-DA2 NIC

Benchmarking Experiences (Multi-pods)

- Test performance of VPP in service function chain scenario (2 pods)
- Figure out impact of NUMA allocation over CNF, vSwitch, NIC
  - 6 scenarios
    - vSwitch same with NIC
      - vSwitch same with input CNF and vice versa
    - vSwitch different with NIC
      - vSwitch same with input CNF and vice versa





Benchmarking Experiences (Multi-pods)

What we learned

NUMA alignment:

- **vSwitch and NIC** in different nodes slightly degrade performance in 1024+ packet size
- **CNFs and vSwitch** in different nodes degrade performance by 10-15%
- **Input CNF and vSwitch** in different node has better performance