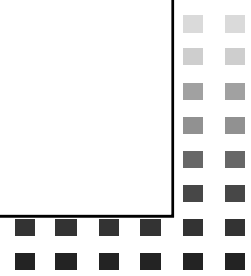




# **BMWG – Containerized Infrastructure Benchmarking**

**IETF 114 Hackathon  
July 23-24, 2022  
Remote**



# Hackathon Plan

- Our draft main goal is to figure out container networking performance impacts by various resource options.
  - Draft:  
Considerations for Benchmarking Network Performance in Containerized Infrastructures  
<https://tools.ietf.org/html/draft-dcn-bmwg-containerized-infra>
  - Two main features
    - Verify container network performance with **various network acceleration models**
    - Verify performance impacts depending on **different configuration settings**

# Hackathon Plan

## What we have done so far in Hackathon

### 1. Models

- ✓ Kernel-space
- ✓ User-space (OVS DPDK, VPP)
- ✓ SmartNIC (SRIOV)
- ✓ Combined (SRIOV-VPP)

### 2. Configuration

- ✓ NUMA (CNF, vSwitch, NIC)
- ✓ Hugepages
- ✓ Service chains (multiple pods)

## In this hackathon

- eBPF Acceleration Model

4.	Networking Models in Containerized Infrastructure . . . . .	8
4.1.	Kernel-space vSwitch Model . . . . .	9
4.2.	User-space vSwitch Model . . . . .	10
4.3.	eBPF Acceleration Model . . . . .	10
4.4.	Smart-NIC Acceleration Model . . . . .	12
4.5.	Model Combination . . . . .	13
5.	Performance Impacts . . . . .	14
5.1.	CPU Isolation / NUMA Affinity . . . . .	14
5.2.	Hugepages . . . . .	15
5.3.	Service Function Chaining . . . . .	15

### BMWG – Containerized Infrastructure Benchmarking

- Champion(s)
  - Younghak Kim <younghak at ssu.ac.kr>
  - Minh-Ngoc Tran <mipearlksa1307 at dcn.ssu.ac.kr>
  - Hokeun Lim <limhk at dcn.ssu.ac.kr>
- Project(s)
  - Benchmarking performance of eBPF acceleration technique
- Specification(s)
  - ➡ <https://datatracker.ietf.org/doc/html/draft-dcn-bmwg-containerized-infra>

# What got done

- Using AF-XDP and OVS-DPDK vSwitch

*NIC ↔ Userspace*

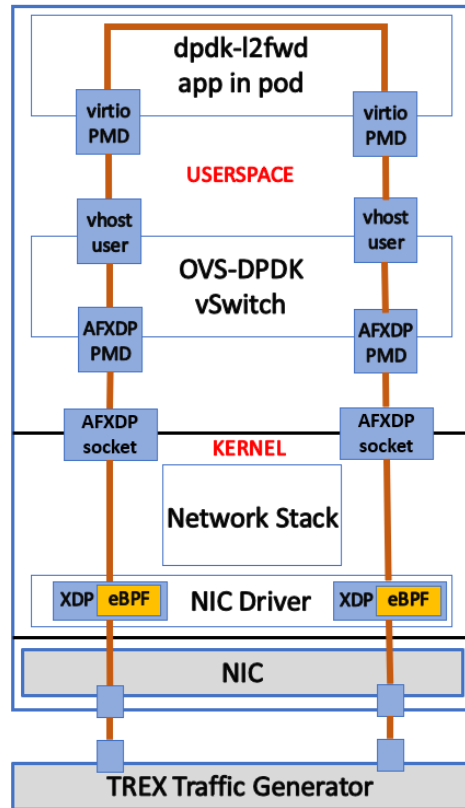
## AF-XDP

- The new socket type available from Linux Kernel > 4.18
- Allows attached eBPF program in XDP hook at kernel NIC driver (native XDP mode) to **transmit packet to userspace bypassing kernel network stack**

*Userspace ↔ Container*

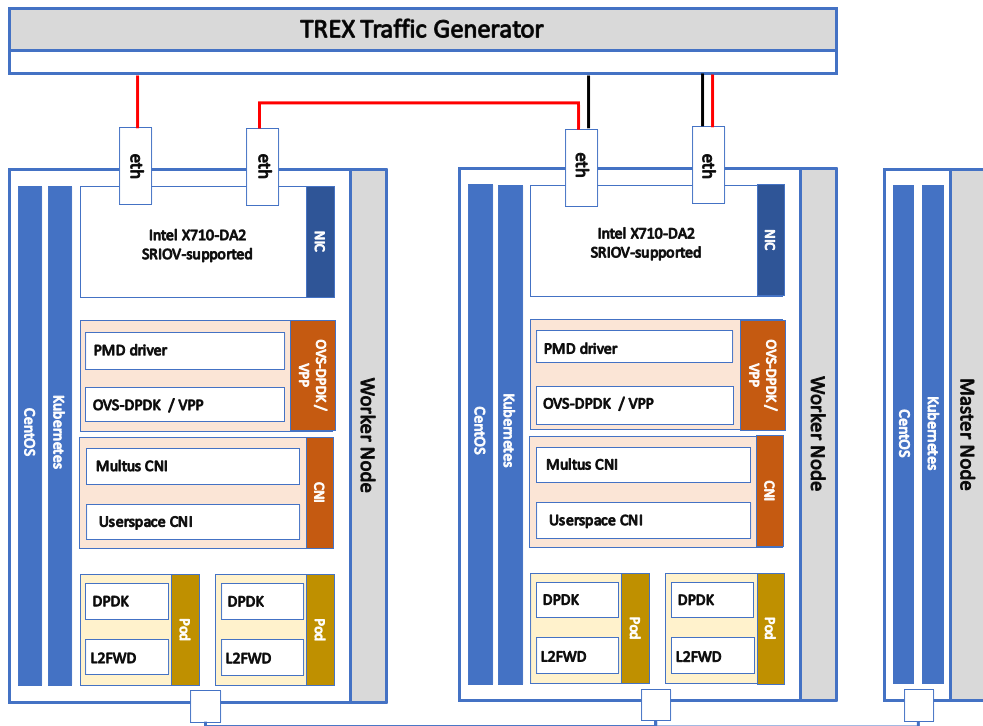
## OVS-DPDK AF-XDP supported version

- An optional built version of OVS-DPDK vSwitch that support AF-XDP
- Can create AF-XDP Poll Mode Driver (PMD) ports that continuously poll packets from AF-XDP sockets**
- Vhostuser ports and Virtio PMDs at application are used to transmit packets between container and host



# What got done

- Benchmarking Testbed – same with previous hackathons



— Single node scenario  
— Multi nodes scenario

- eBPF Supported NIC: Intel X710
- AF-XDP supported kernel: Ubuntu 22.04 (kernel v5.15)
- Pod multi-interfaces: Multus
- vSwitch supported CNI: Userspace CNI

# What got done

- Benchmarking Configuration
- **Hardware – Worker Node**

CPU	Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz 48 CPU cores * 2 NUMA nodes
Memory	256GB: 32GB x 4DIMMs x 2 NUMA nodes @ 2400MHz
NIC	Intel Corporation Ethernet Network Adapter X71-40Gbps
Microcode	0x5003102
Intel NIC Device ID	0x1572
Intel NIC Firmware version	6.01 0x800035cf 1.1747.0
BIOS setting	CPU Power and Performance Policy <Performance> CPU C-state Disabled CPU P-state Disabled Intel(R) Hyper-Threading Tech Enabled Turbo Boost Disabled

- **Traffic Generator : T-Rex (v2.92)**

Name	T-Rex
Version	2.92
Benchmark method	T-Rex Non Drop Rate application (accepted percentage of drop rate is less than 0.1%)

- **Software**

Operating System	Ubuntu 22.04
Linux Kernel Version	5.15
GCC version	gcc version 4.8.5 20150623 (Red Hat 4.8.5-44)
DPDK version	21.11.1
Hugepages	1Gi

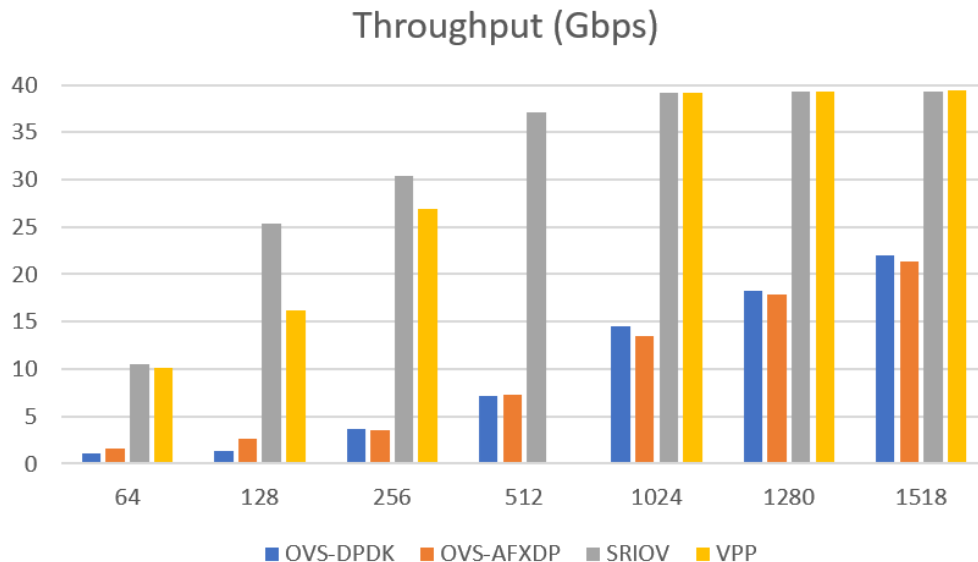
# What we learned

- Benchmarking Performance Results vs OVS-DPDK, SRIOV, VPP (Single Pod)

- OVS-AFXDP catches up with the performance of OVS-DPDK
- But significantly lower behinds SRIOV and VPP
- The reason might be at the limitation of vhostuser-virtioPMD path between container and vSwitch
- VPP uses memif PMD (shared memory packet interface) which is a better performance method

→ This result might not show true performance of AF-XDP-eBPF acceleration model

→ Using AF\_XDP with VPP vSwitch and memif interfaces might significantly improve the performance



# Future Works

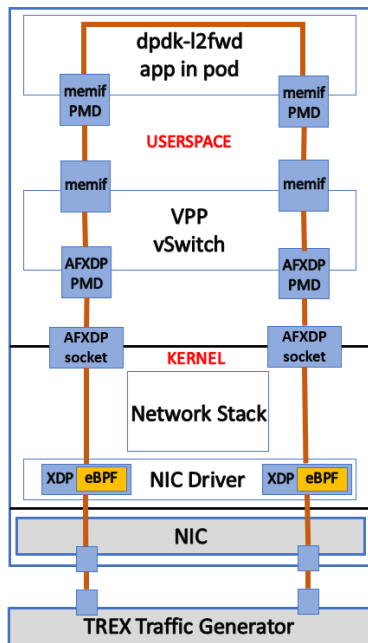
- Performance comparison with other XDP-eBPF acceleration model variations
  - **VPP-AF-XDP**
  - **Cloud Native Data Plane (CNDP)**: A new cloud native userspace framework developed by Intel (first release April 2022) which utilizes AF-XDP and VPP
  - **Cilium**: eBPF based CNI

Differences at packet transmission routes between XDP socket and container

Differences at East-West (E-W) traffic handling

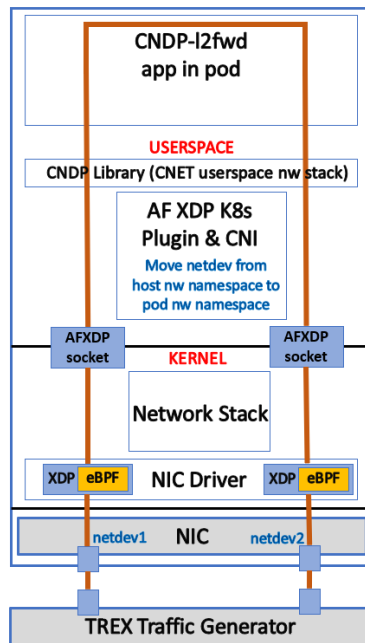


# Future Works



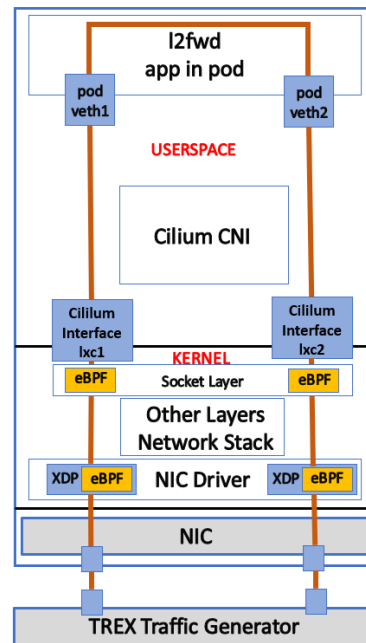
## VPP-AFXDP

Use vSwitch pmd and memif



## CNDP

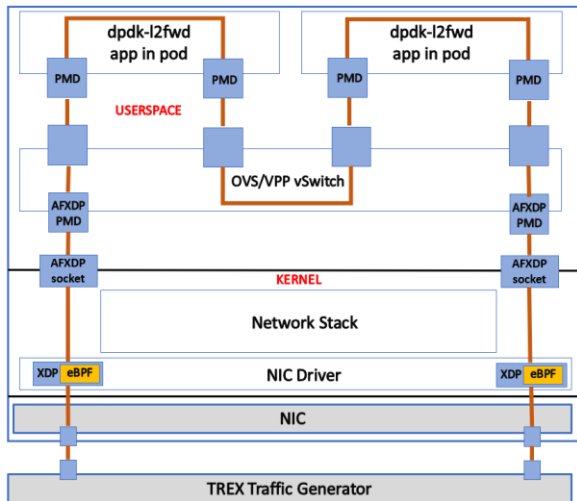
Create afxdp socket at pod namespace



## Cilium

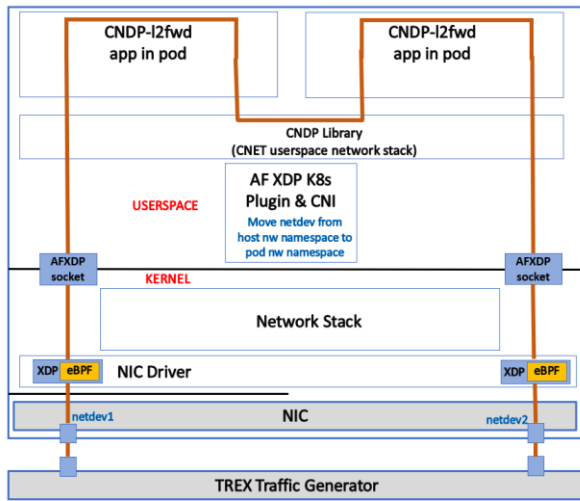
Normal xdp at NIC driver and Socket Layer, CNI veth pair with pod

# Future Works



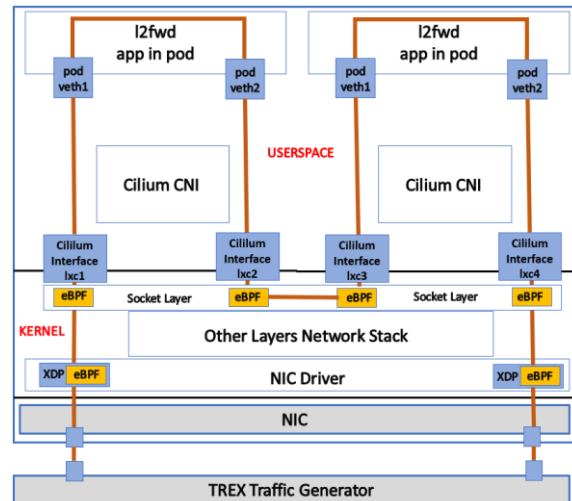
## OVS/VPP-AFXDP

E-W using vSwitch



## CNDP

E-W using VPP network stack  
(Assumption: not yet described by official docs)



## Cilium

E-W using eBPF at NW Socket Layer

# Wrap Up

## Team members:

**Younghan Kim (SSU)**

**Minh Ngoc Tran(SSU)**

**Thanh Nguyen Nguyen (SSU)**

**Jangwon Lee (SSU)**

**Hokeun Lim (SSU)**

## Git repo:

<https://github.com/SSU-DCN/bmwg-container-networking>

## Remote Hackathon from Seoul

Internet Infra System Technology  
Research Center – Soongsil University  
(IISTR- SSU)