# 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	
<u>1</u> . Introduction	<u>3</u>
2. Terminology	<u>4</u>
3. Containerized Infrastructure Overview	
4. Networking Models in Containerized Infrastructure	<u>8</u>
<u>4.1</u> . Kernel-space vSwitch Model	
4.2. User-space vSwitch Model	
4.3. eBPF Acceleration Model	
4.4. Smart-NIC Acceleration Model	
<u>4.5</u> . Model Combination	
<u>5</u> . Performance Impacts	<u>14</u>
5.1. CPU Isolation / NUMA Affinity	<u>14</u>
5.2. Hugepages	<u> </u>
5.3. Service Function Chaining	<u>15</u>
<u>5.4</u> . Additional Considerations	<u>     16</u>

CURRENT - version 09	
<u>1</u> . Introduction	3
<u>2</u> . Terminology	Į.
3. Containerized Infrastructure Overview	1
4. Benchmarking Considerations	2
	5
4.2. Additional Configuration Parameters	3
	9
4.3.1. Kernel-space vSwitch Model	2
<u>4.3.2</u> . User-space vSwitch Model	2
4.3.3. eBPF Acceleration Model . Update AFXDP 1	L
4.3.4. Smart-NIC Acceleration Model	3
<u>4.3.5. Model Combinati</u> on	Į.
4.4. Performance Impacts	5
4.4.1. CPU Isolation / NUMA Affinity	5
4.4.2. Hugepages	5
4.4.3. Service Function Chaining	
4.4.4. Additional Considerations	7
- <del>-</del> · · · · · <del>·</del>	2

# 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)

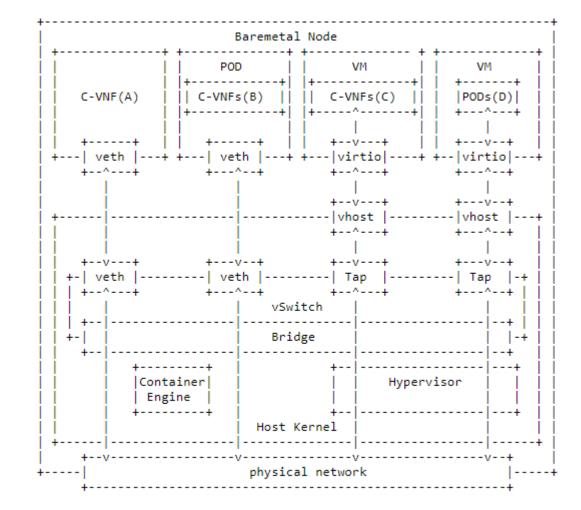


Figure 1: Examples of Networking Architecture based on Deployment Models - (A)C-VNF on Baremetal (B)Pod on Baremetal(BMP) (C)C-VNF on VM (D)Pod on VM(VMP)

# 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

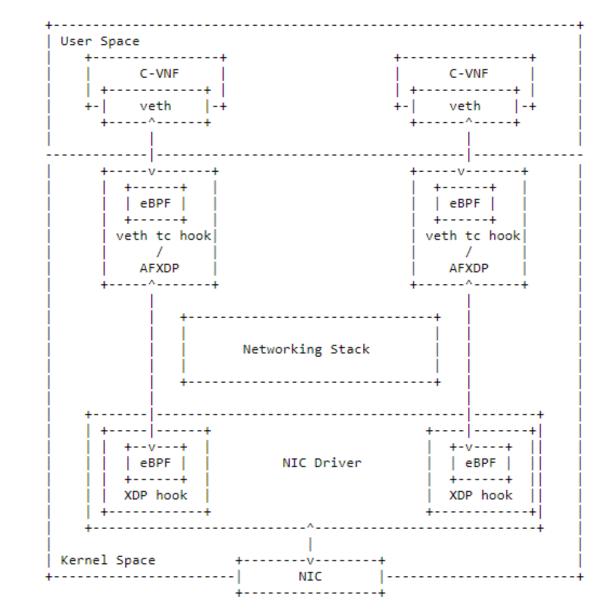


Figure 6: Examples of eBPF Acceleration Model

# Detailed Updates (4)

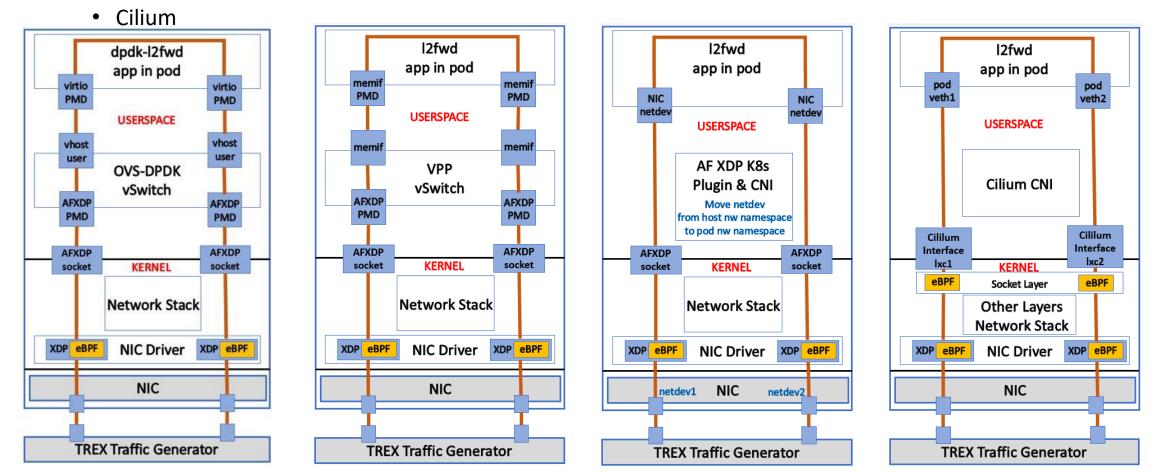
### **Benchmarking Consideration 4**

### **Performance Impacts**

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

## From Hackathon 114-115

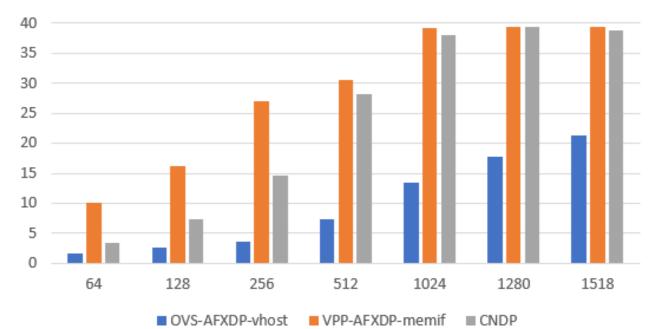
- eBPF Acceleration Model Benchmarking (4 variations)
  - OVS-vhost, VPP-memif vSwitch (AFXDP supported version),
  - Intel Cloud Native Data Plane CNDP,



### 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)

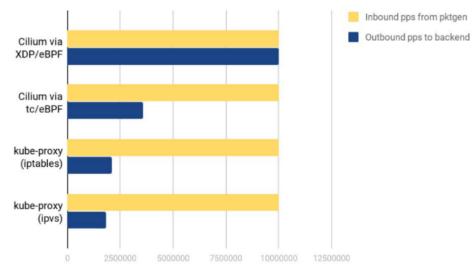


### Throughput (Gbps)

### From Hackathon 114-115

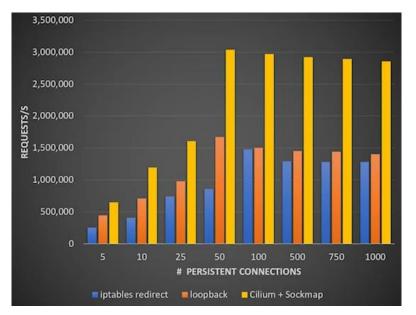
### • Benchmarking Performance Results – eBPF Acceleration Models

- 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" (<u>https://cilium.io/blog/2020/06/22/cilium-18/</u>)
  - East-West: "Istio 1.0: How Cilium enhances Istio with socket-aware BPF programs" (<u>https://cilium.io/blog/2018/08/07/istio-10-cilium/</u>)



#### Forwarding performance of tested Kubernetes node (higher is better)

North-South XDP acceleration vs kernel kube-proxy



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

### Next Steps – Request for feedback

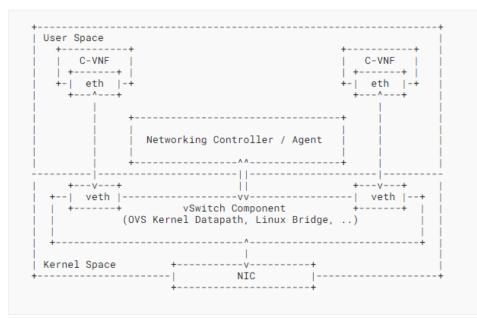
- IETF 115 Hackathon has completed our investigation activities for all proposed benchmarking considerations for containerized infrastructure in our draft.
- We would like to hear any questions and comments from anyone in BMWG that is interested in our draft.
- We will finalize the draft and would like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

# Backup Slides

## Networking Models

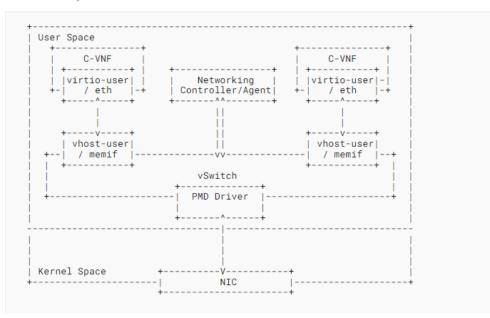
- Kernel-space vSwitch
- User-space vSwitch

- eBPF Acceleration Model
- Smart-NIC Acceleration Model
- Model Combination



#### 4.3.1. Kernel-space vSwitch Model



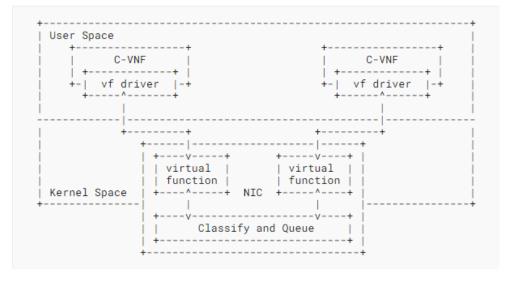


## Networking Models

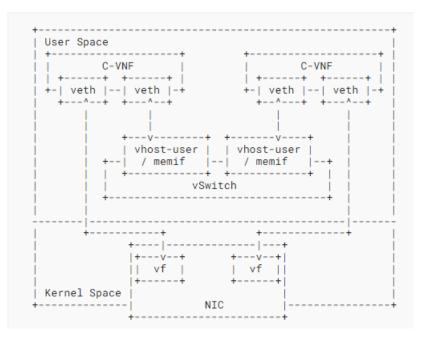
- Kernel-space vSwitch
- User-space vSwitch

4.3.4. Smart-NIC Acceleration Model

- eBPF Acceleration Model
- Smart-NIC Acceleration Model
- Model Combination



#### 4.3.5. Model Combination



## Performance Impacts

 $\circ$  Hugepages

NUMA & CPU Isolation

**O Service Function Chaining** 

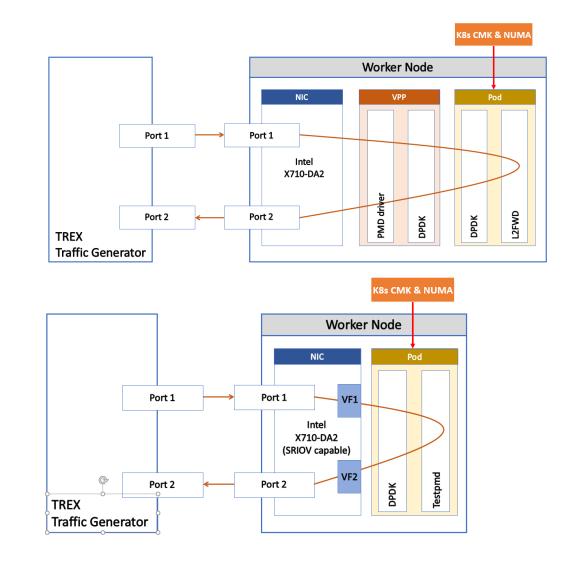
- In NFV environment, physical network port is commonly connected to multiple VNFs rather than single VNF
- Aspects needed to be considered when benchmarking service function changing
  - $\circ~$  Number of VNFs
  - Different network acceleration technologies (which provide VNF to VNF networking)

### $\circ$ 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.

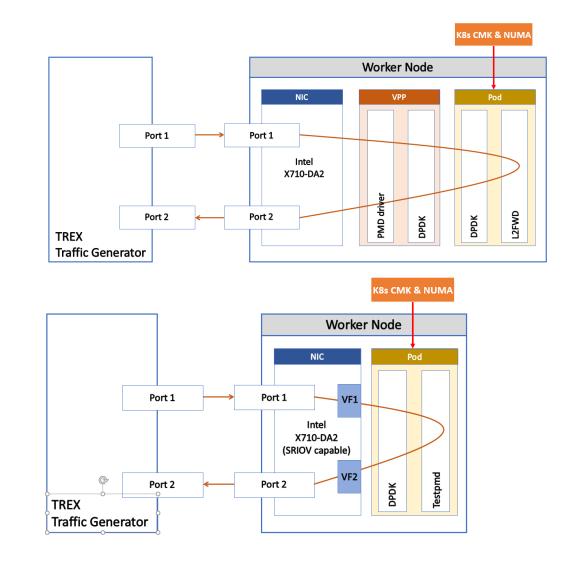
## Benchmarking Experiences (Contiv-VPP + SRIOV)

- 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:**

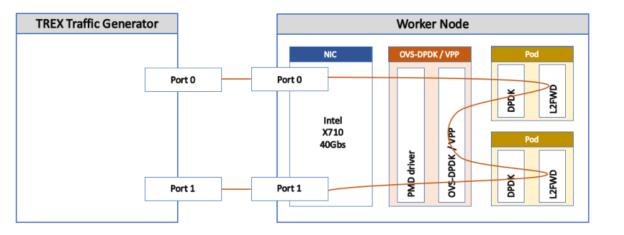
• Assigning CPU in the same NUMA node is better than in different NUMA nodes

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



different

with NIC

NUMA node

NUMA node 0 NUMA node 1									NUMA node 0						NUMA node 1								
	1	2	3	4	-	VSI	vitch		23	24	0	1	2	3	4			itch					
5	6	7	8	9	-		NF1	27	28	29	5	6	7	-	F 2		CN	_	27	23	29		
10	11	12	13	1	4	C	NF 2	32	33	34	10	11	12	13	14		30	31	32	33	34		
15	16	17	18	1	9	35	36	37	38	39	15	16	17	18	19		35	36	37	38	39		
	м	emo	ry				M	emo	iry			М	emo	iry			Memory						
	Syst	tem I	Bus				Sys	tem	Bus		System Bus						System Bus						
Sce	ena	rio	1(	s1	1)			NIC			Sc	ena	ario	<b>5</b> 2	(s2)	)			NIC				
V			-h								-	-		ode	-	]		NUM			-		
V	sw	ito	ch								0	1	2	3	4		vSw	itch	22	23	24		
	sw		ch								-	-		3	-			itch			-		
sa		e		10	bd	e					0	1	2	3 CN	4 F 1		vSw CN	<mark>itch</mark> F 2	22 27	23 28	24 29		
sa N	am	ne MA	A r		od	е					0 5 10	1 6 11 16	2 7 12	3 CN 13 18	4 F 1 14		vSw CN 30	itch F 2 31 36	22 27 32	23 28 33 38	24 29 34		
sa N	am UN	ne MA	A r		bc	e					0 5 10	1 6 11 16 N	2 7 12 17	3 CN 13 18 ory	4 F 1 14		vSw CN 30	itch F 2 31 36 M	22 27 32 37	23 28 33 38	24 29 34		
sa N	am UN	ne MA	A r		bc	e					0 5 10	1 6 11 16 N Sys	2 7 12 17 17	3 CN 13 18 Dry Bus	4 F 1 14 19		vSw CN 30	itch F 2 31 36 M Sys	22 27 32 37 emo	23 28 33 38 TY Bus	24 29 34		

NUI	NUMA node 0 NUMA node 1								NUMA node 0						NUMA node 1							
vSwitc	2	3	4	20	21	22	23	24	vSv	vitch	2	3	4		20	21	22	23	2			
CNF1	7	8	9	25	26	27	28	29	CN	F1	7	8	9		CN	F 2	27	28	2			
CNF 2	12	13	14	30	31	32	33	34	10	11	12	13	14		30	31	32	33	3			
15 16	17	18	19	35	36	37	38	39	15	16	17	18	19		35	36	37	38	3			
N	Memory					Memory						Memory						Memory				
Sy	System Bus					tem	Bus			System Bus					System Bus							
Scen			NIC	:		Sc	Scenario5 (s5)						NIC									
Vsw	vito	h								NUN		ode	0	]	<u> </u>			de 1				

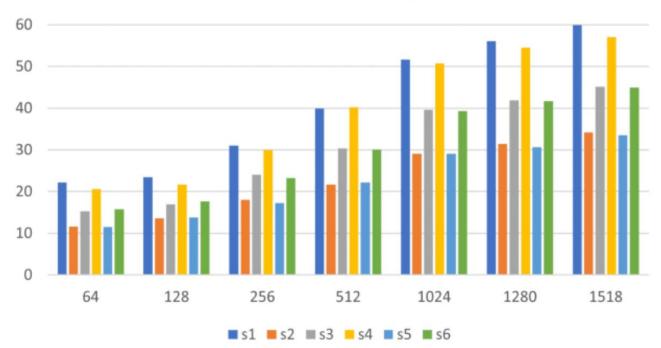
1	NUM	A no	de (	)	1	NUM	IA no	de 1	ι
vSw	vitch	2	3	4	20	21	22	23	24
CN	F 2	7	8	9	CN	F 1	27	28	29
10	11	12	13	14	30	31	32	33	34
15	16	17	18	19	35	36	37	38	39
	М	emo	ry			М	emo	iry	
	Syst	tem	Bus			Sys	tem	Bus	
Sc	ena	ario	o6 (	s6)			NIC		

# 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



VPP (%line rate)