IEEE NFV-SDN 2018 Best Paper Award

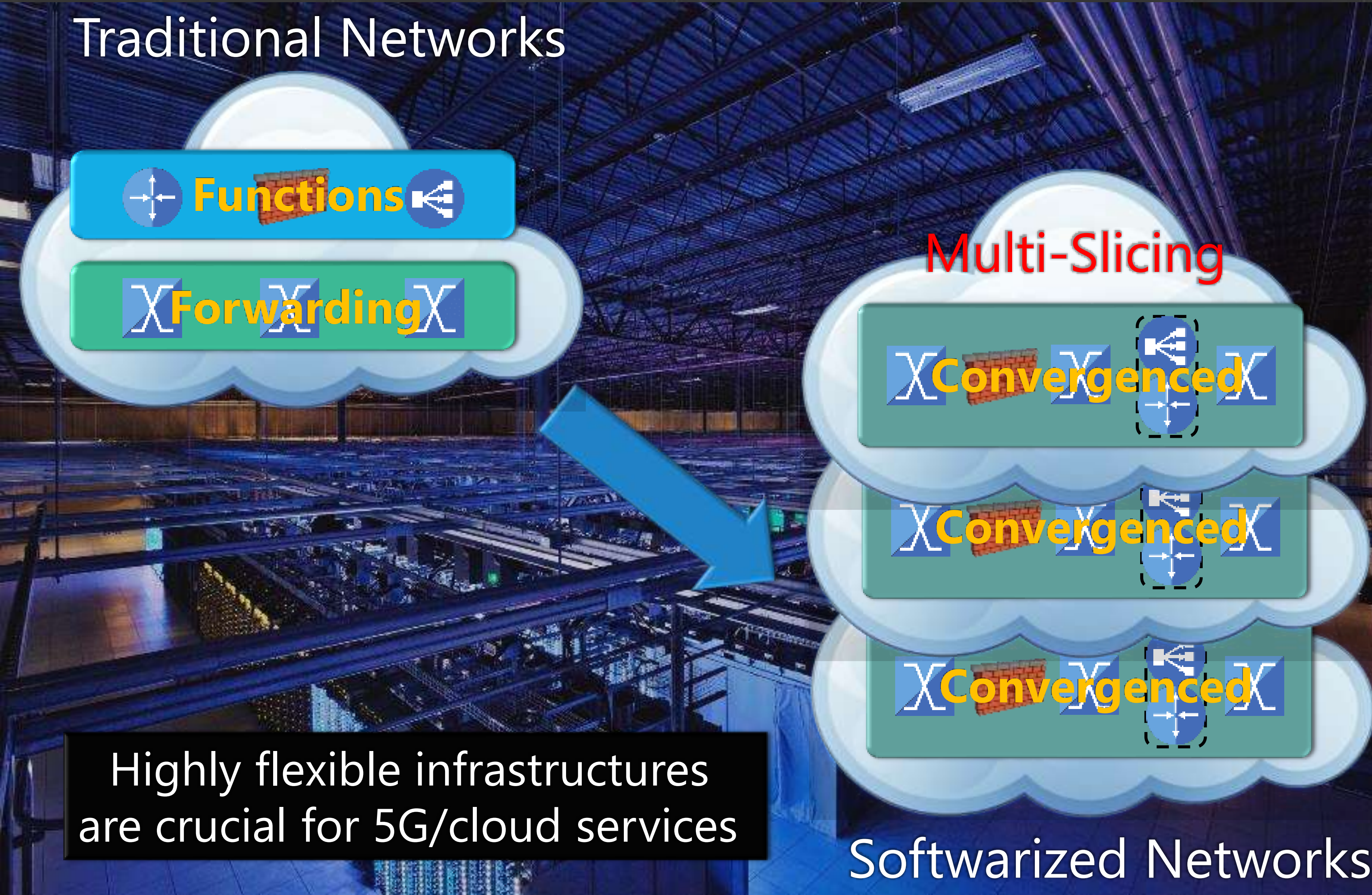
IOVTee: A Fast and Pragmatic Software-based Zero-copy/Pass-through Mechanism for NFV-nodes

Assist. Prof. Ryota Kawashima

Nagoya Institute of Technology, Japan

Contents

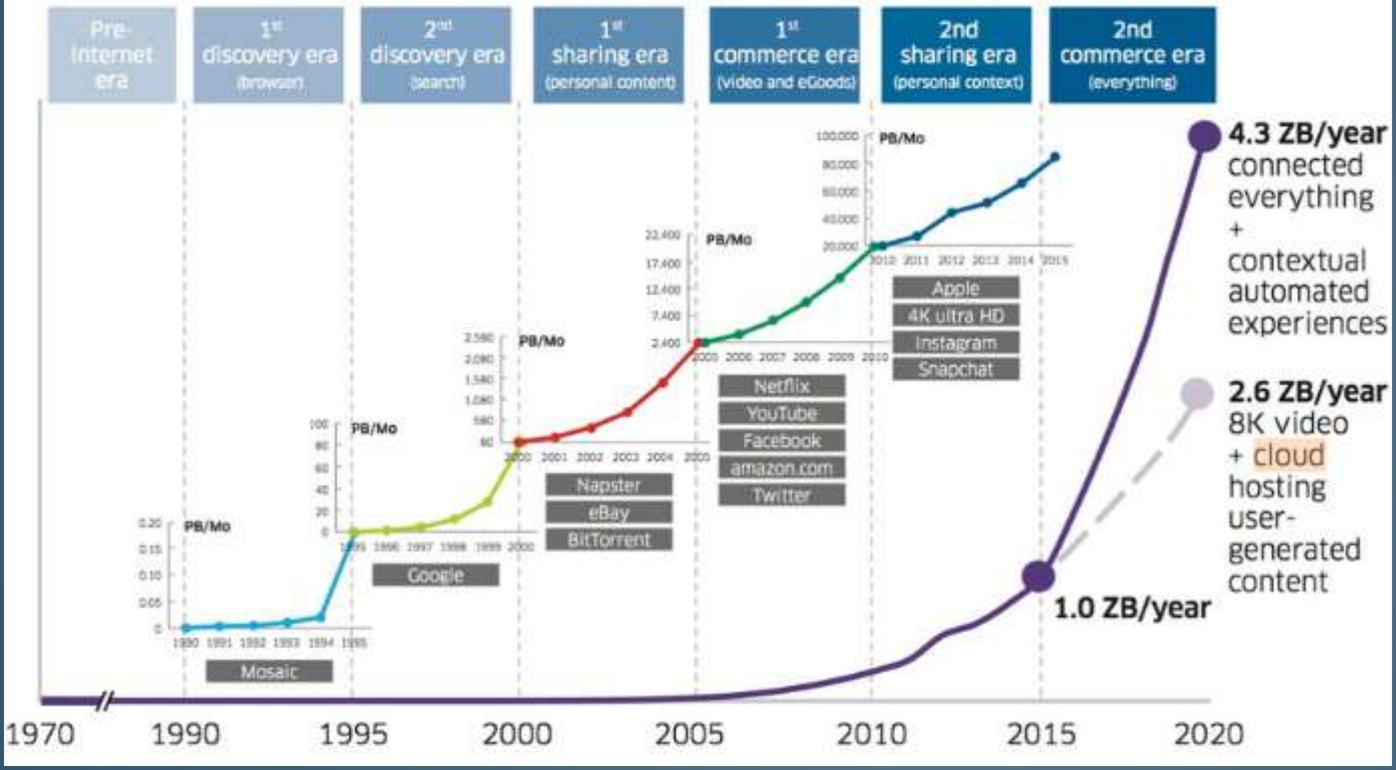
Softwarization for Ultimate Flexibility



Highly flexible infrastructures
are crucial for 5G/cloud services

High-Speed Communications

Core Network Traffic



Marcus K. Weldon, “The Future X Networks”

Hardwares

400G Ethernet

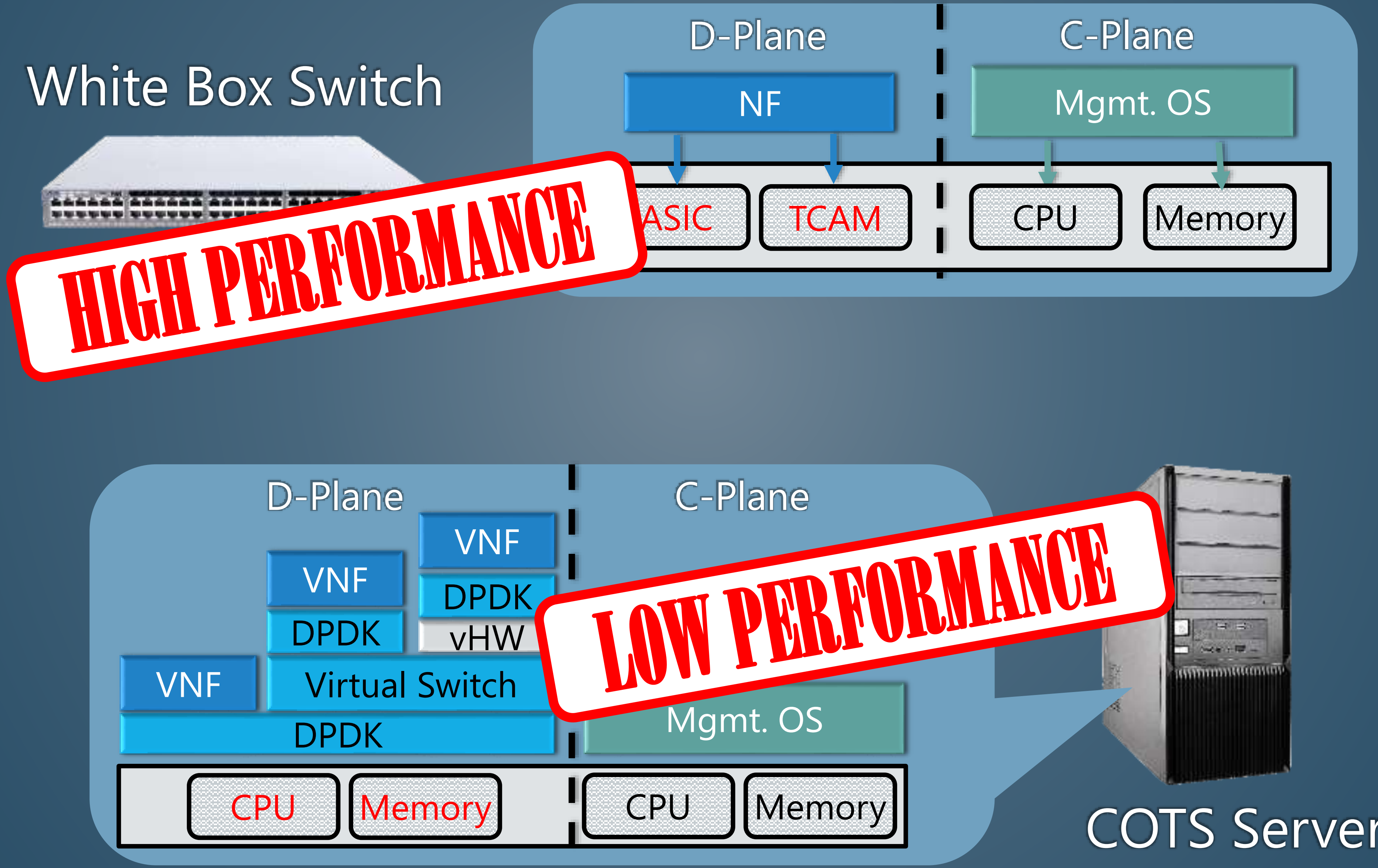
Softwares

DPDK
DATA PLANE DEVELOPMENT KIT

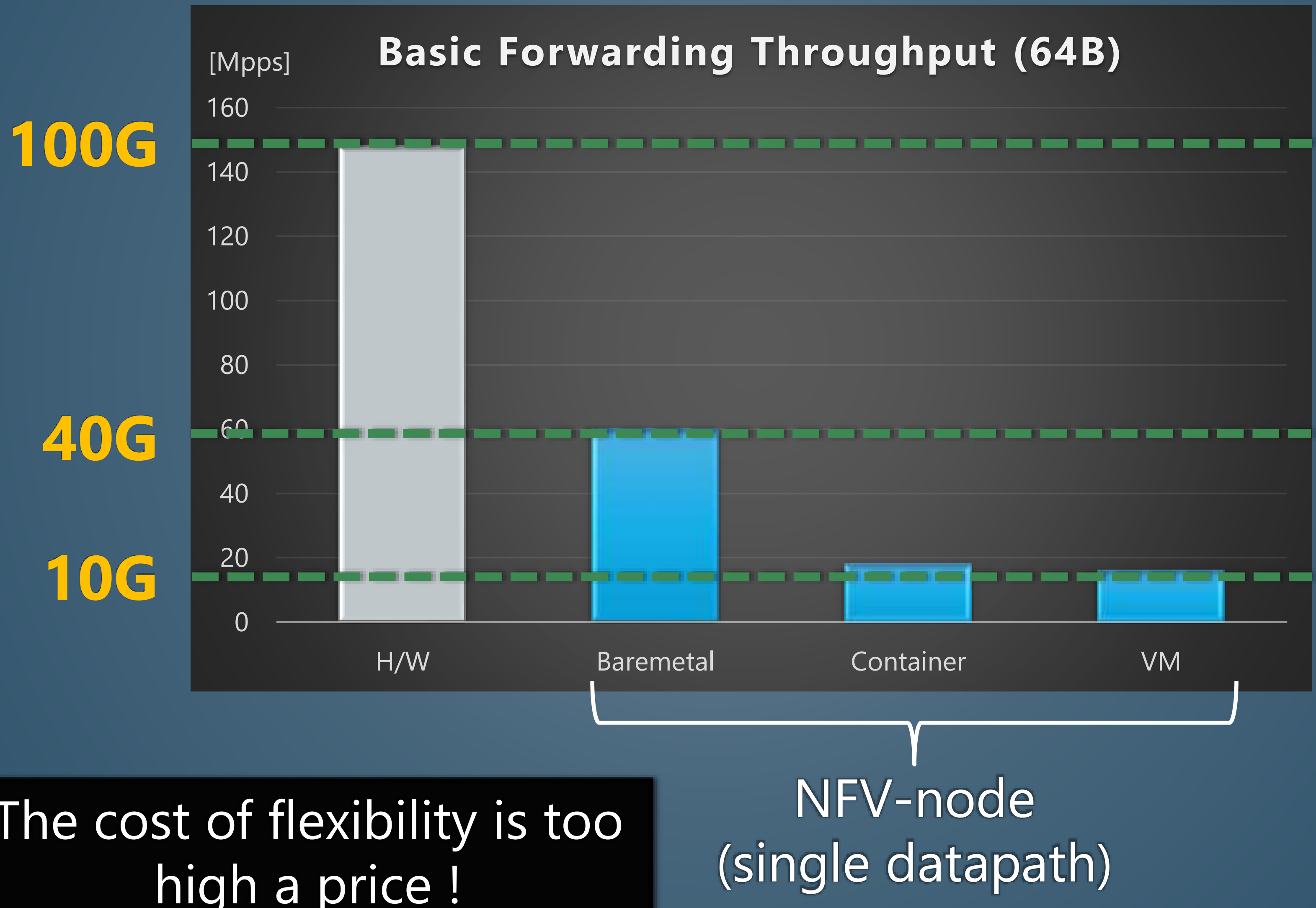
fd.io
The Universal Dataplane

Are software-based approaches viable ?

White Box Switches vs. COTS Servers



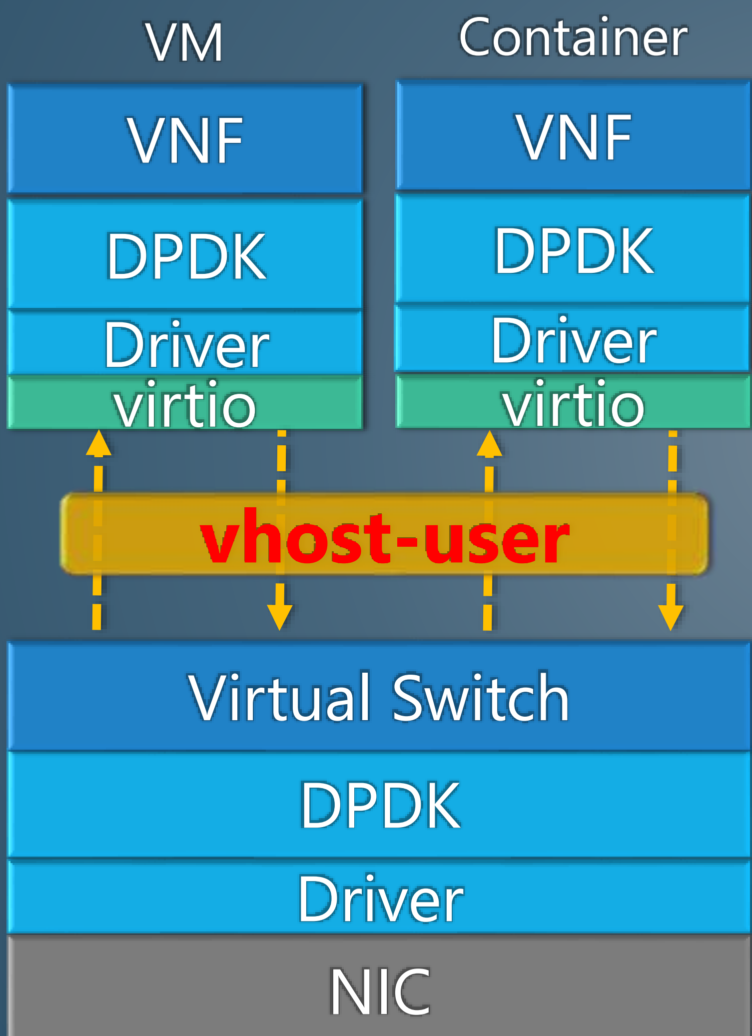
The Reality of NFV-nodes (COTS Servers)



The cost of flexibility is too high a price !

Contents

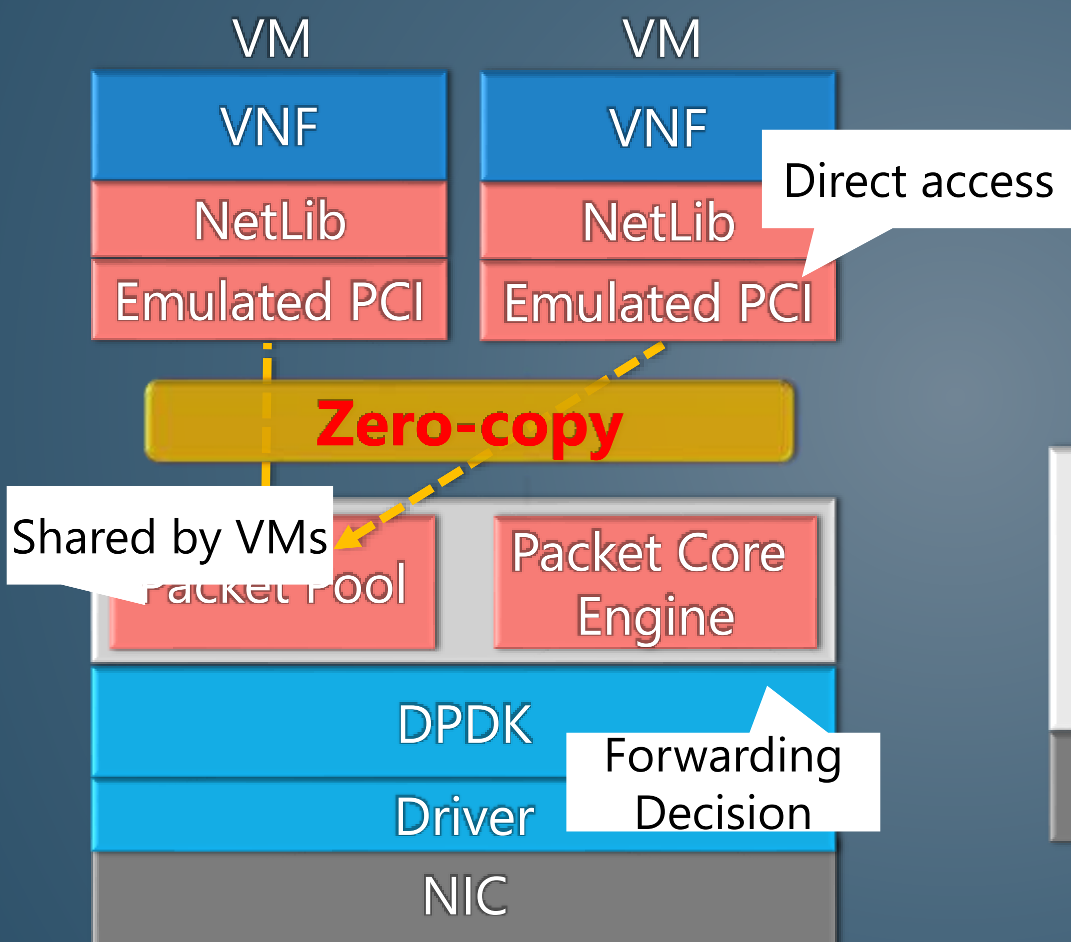
vhost-user (de-facto)



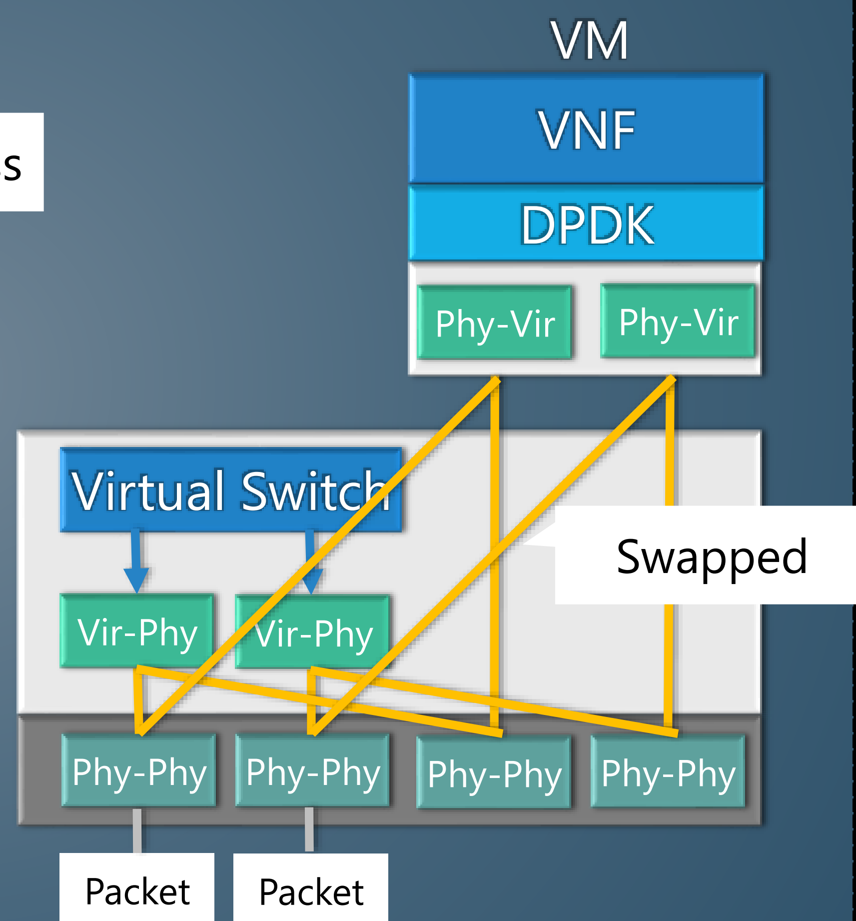
- User-space to User-space
- Packet copy (each direction)
- Tx zero-copy (optional)

Zero-copy Approaches

NetVM*



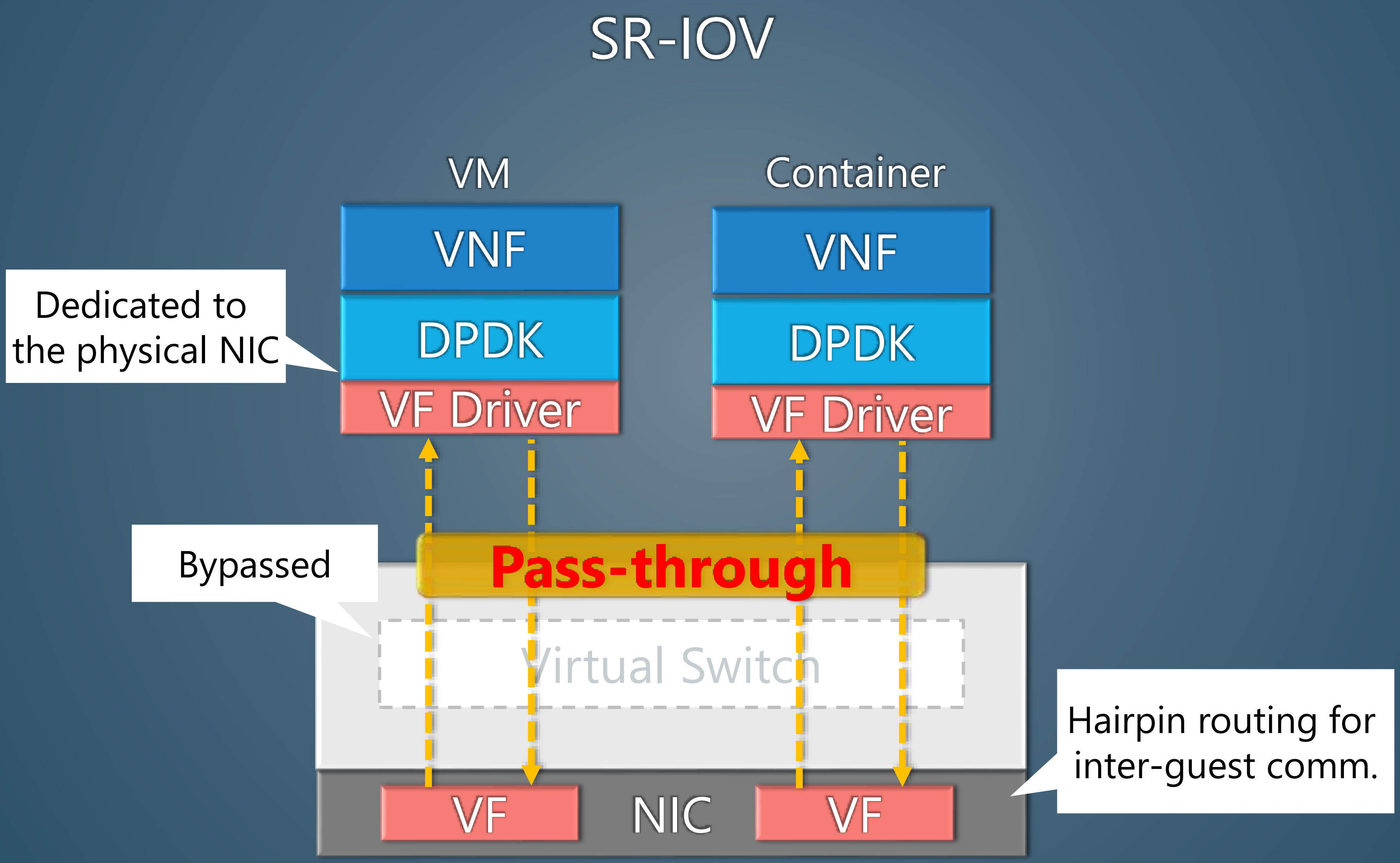
Zcopy-vhost**



* J. Hwang, et al., "NetVM: High Performance and Flexible Networking Using Virtualization on Commodity Platforms", IEEE TNSM, vol. 12, no. 1, pp. 34-47, 2015

** D. Wang, et al., "Zcopy-vhost: Eliminating Packet Copying in Virtual Network I/O", Proc. IEEE LCN, pp. 632-639, 2017

(H/W) Pass-through Approaches



Problem Statements

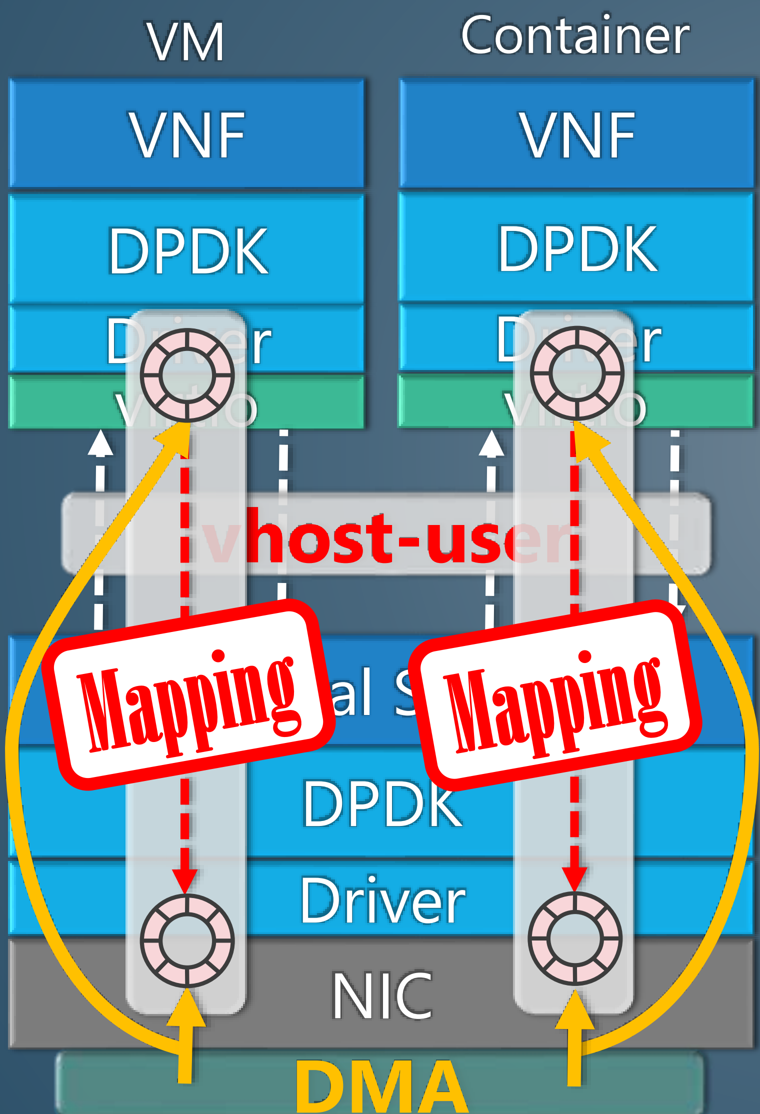
Concerns	Description	Methods
Security	Exposing the host memory	NetVM, IVSHMEM
Transparency	VNFs are aware of the host environment	NetVM, IVSHMEM, Zcopy-vhost, SR-IOV
Portability	The method broadly depends on other system components	Zcopy-vhost
Traceability	Internal behaviors are hidden	SR-IOV
Container	Container-based VNFs are not supported	IVSHMEM, Zcopy-vhost

The existing methods have pragmatic problems

A yet another practical approach is needed

Contents

Proposed Approach (IOVTee)

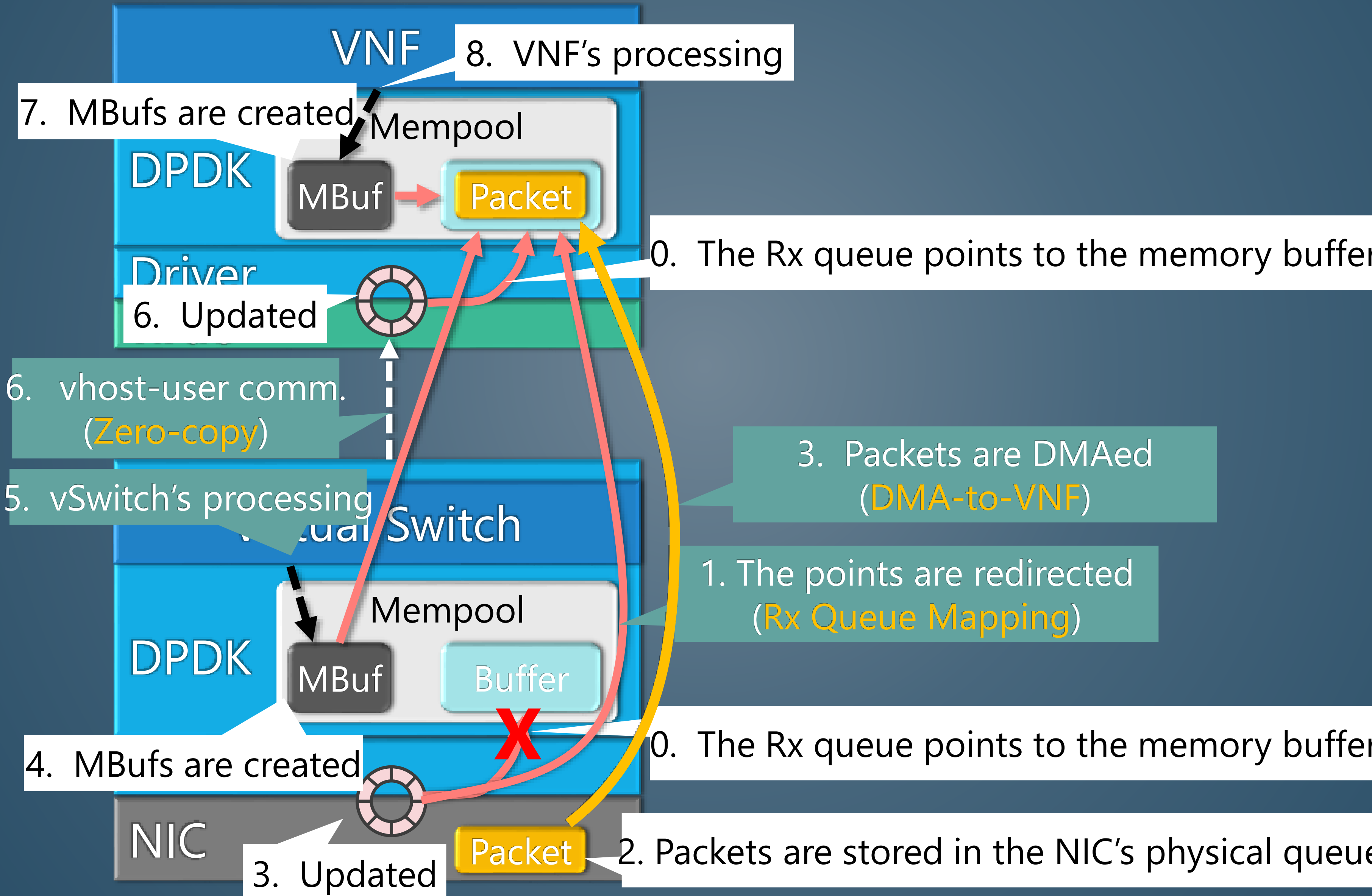


Rx Queue Mapping
(DMA-to-VNF)

Packet Processing on the Host

Vhost-user Interface

Step-by-Step Description



Are Problems Resolved ?

Concerns	Description	Resolved?
Security	Host memory is NOT exposed to VNFs	✅
Transparency	IOVTee is completely hidden by the vhost-user interface	✅
Portability	IOVTee is implemented only within the host DPDK internals	✅
Traceability	IOVTee is a complete software-based approach	✅
Container	The vhost-user interface is not changed	✅

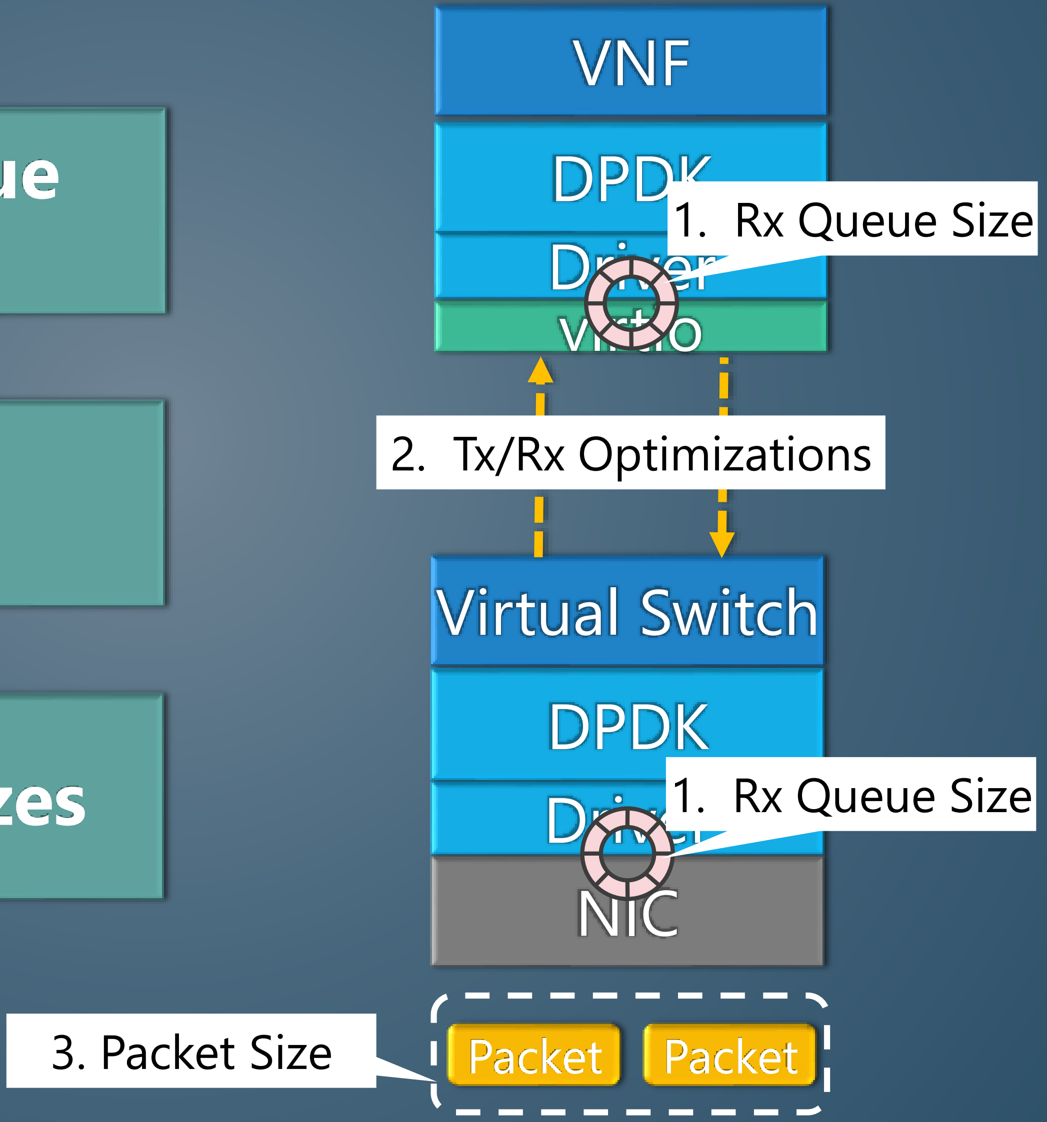
IOVTee is a pragmatic zero-copy/pass-through mechanism

What about the performance ?

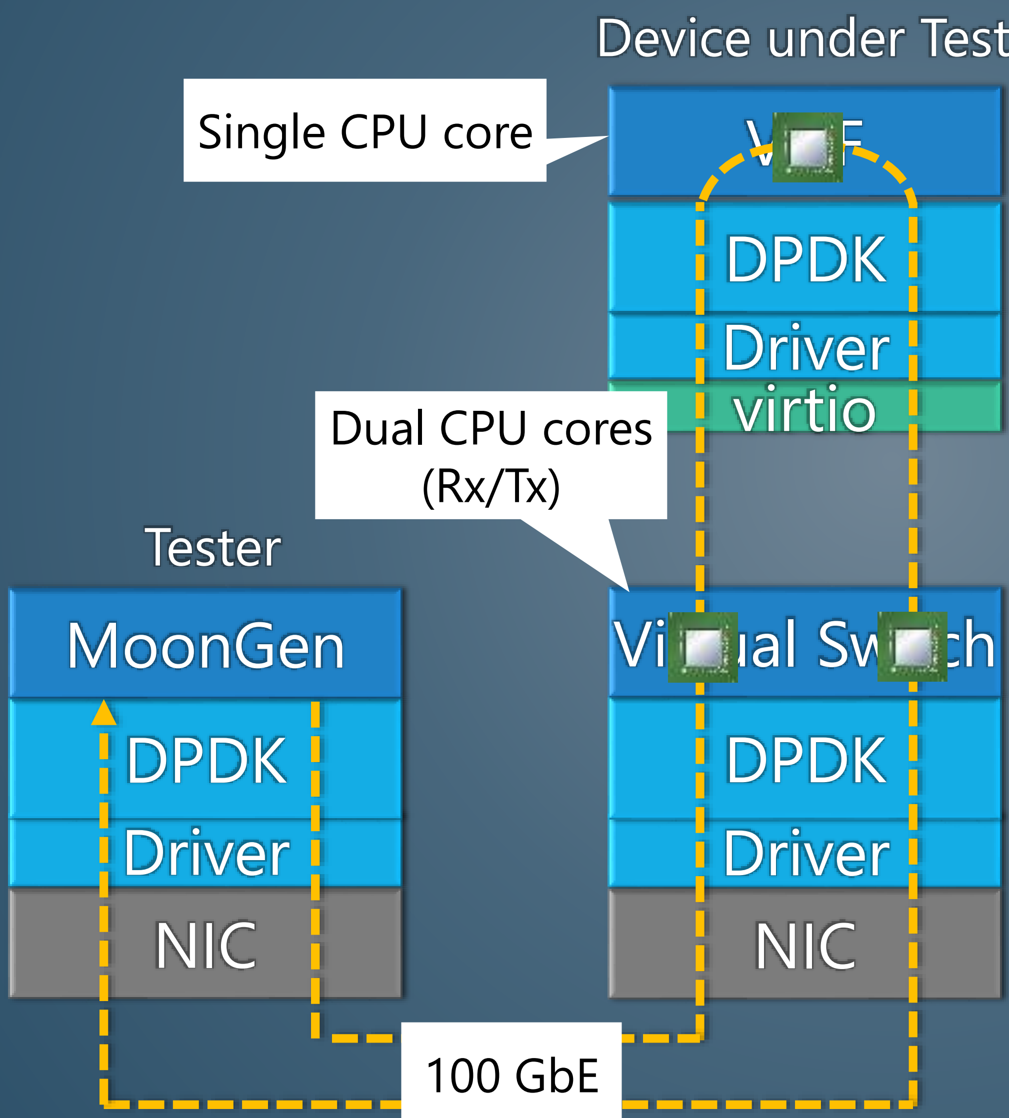
Contents

Three Experiments

1. **Various Rx Queue Sizes**
2. **Various Tx/Rx Optimizations**
3. **Various Packet Sizes**



Environment

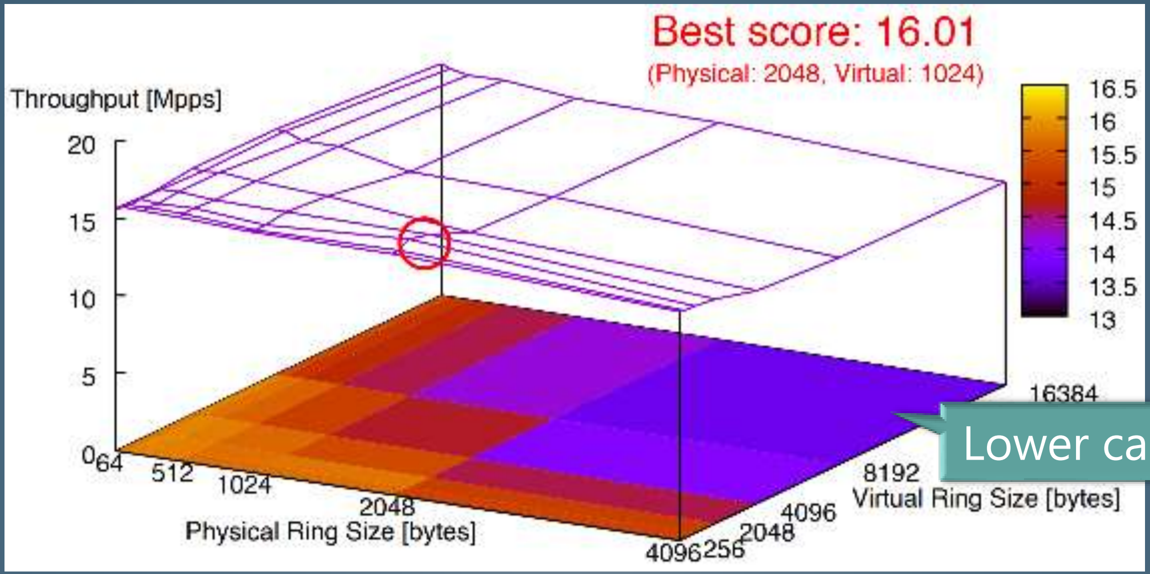


Item	Spec.
OS	CentOS 7.5
DPDK	v18.02
VNF	Testpmd
vCPU	4 cores
Memory	8 GB
vNIC	virtio-net/mlx5

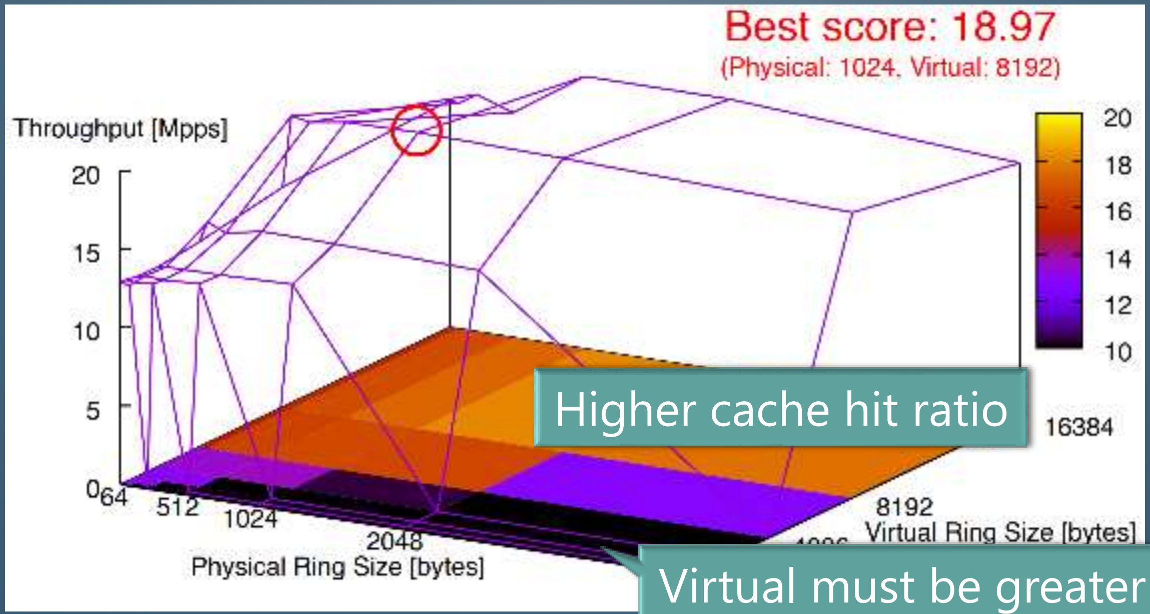
Item	Spec.
OS	CentOS 7.5
DPDK	v18.02
vSwitch	Testpmd
VMM	KVM (QEMU 2.11.0)
CPU	Intel Core i9-7900X 3.3 GHz (10 cores w/o HT)
Memory	32 GB (DDR4-2133)
NIC	Mellanox ConnectX-5 EN

Exp1: Physical/Virtual Ring Sizes

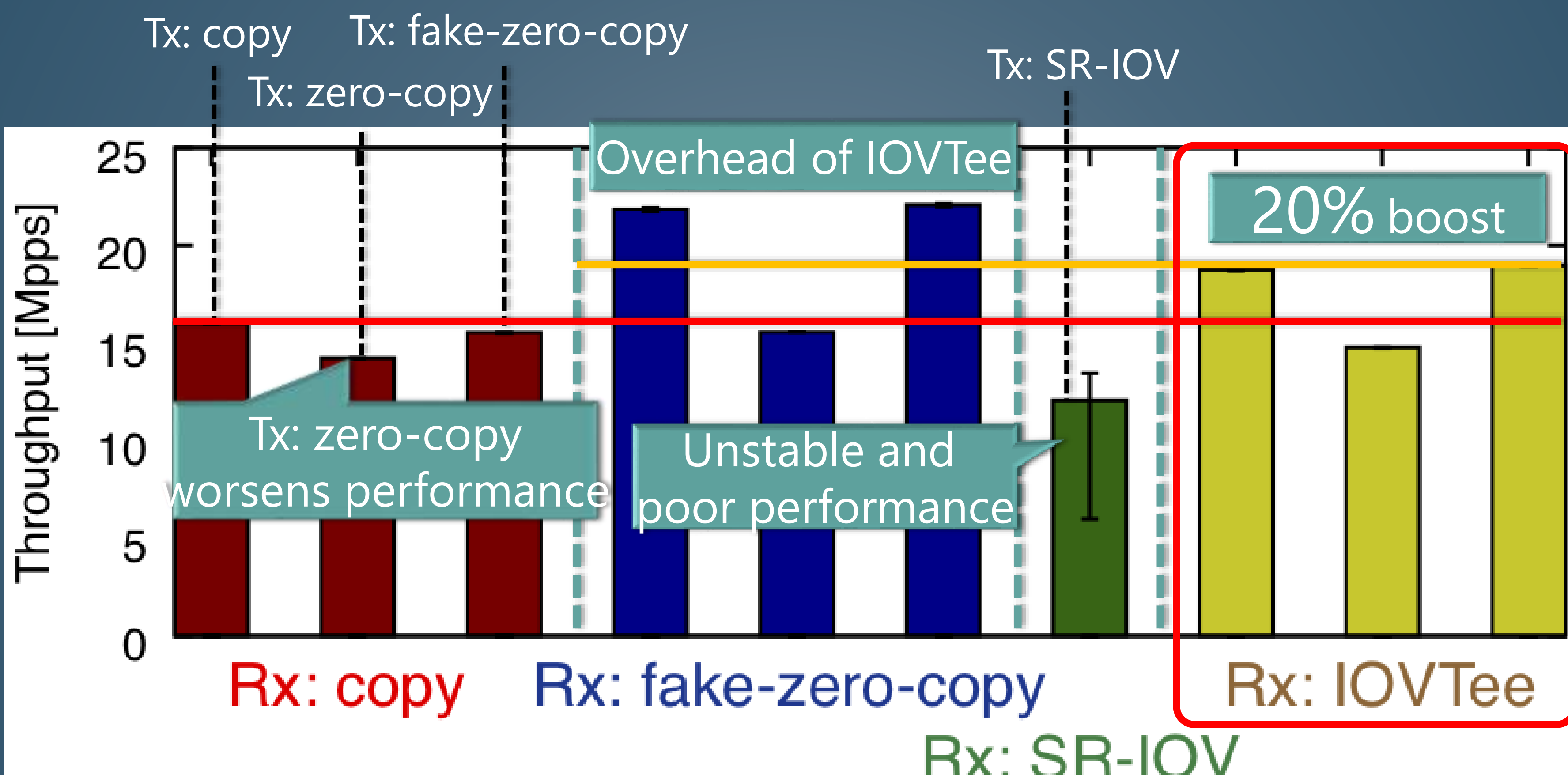
Default vhost-user



IOVTee



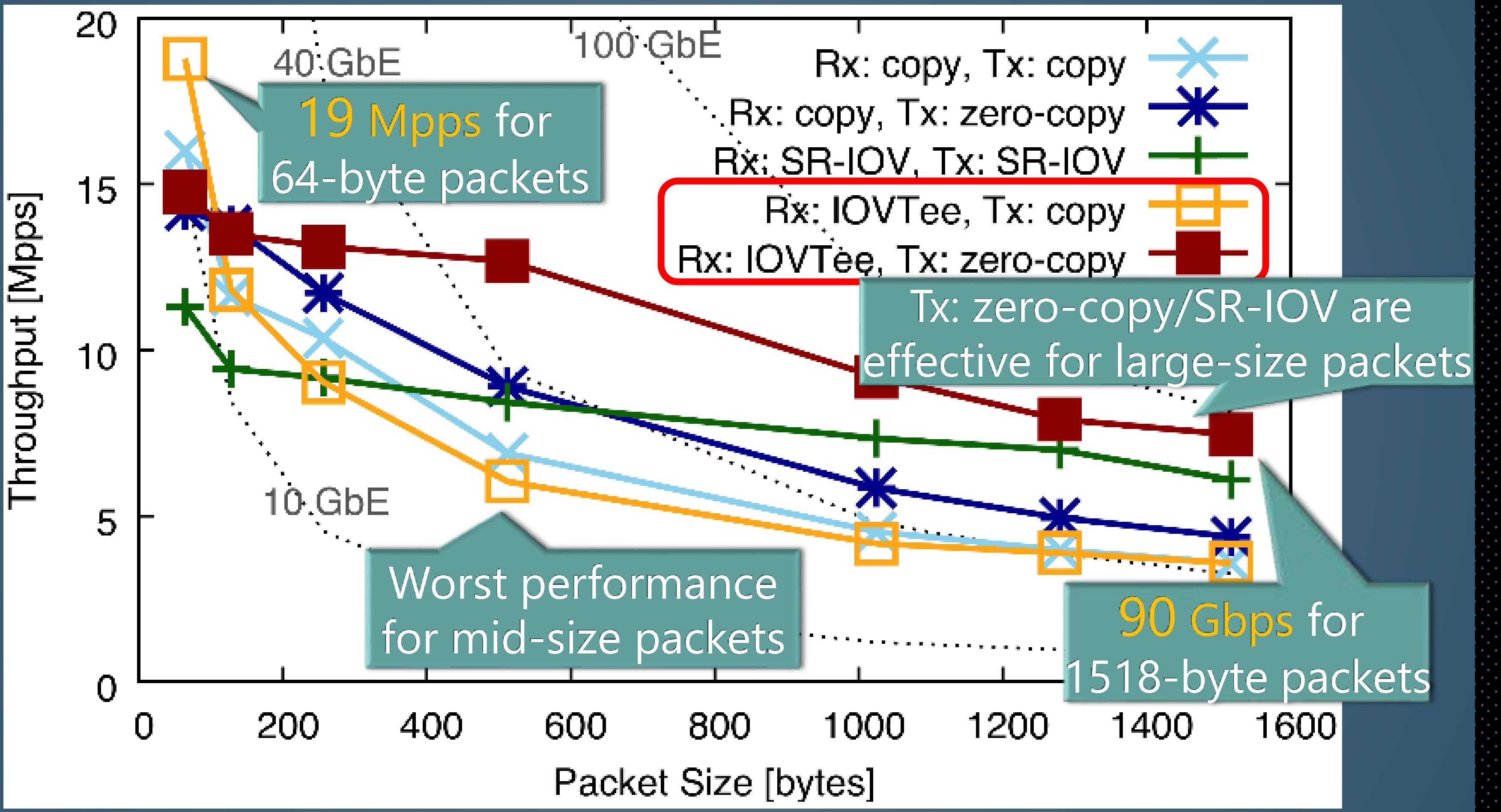
Exp2: Tx/Rx Zero-copy Methods



Zero-copy for Rx path is effective

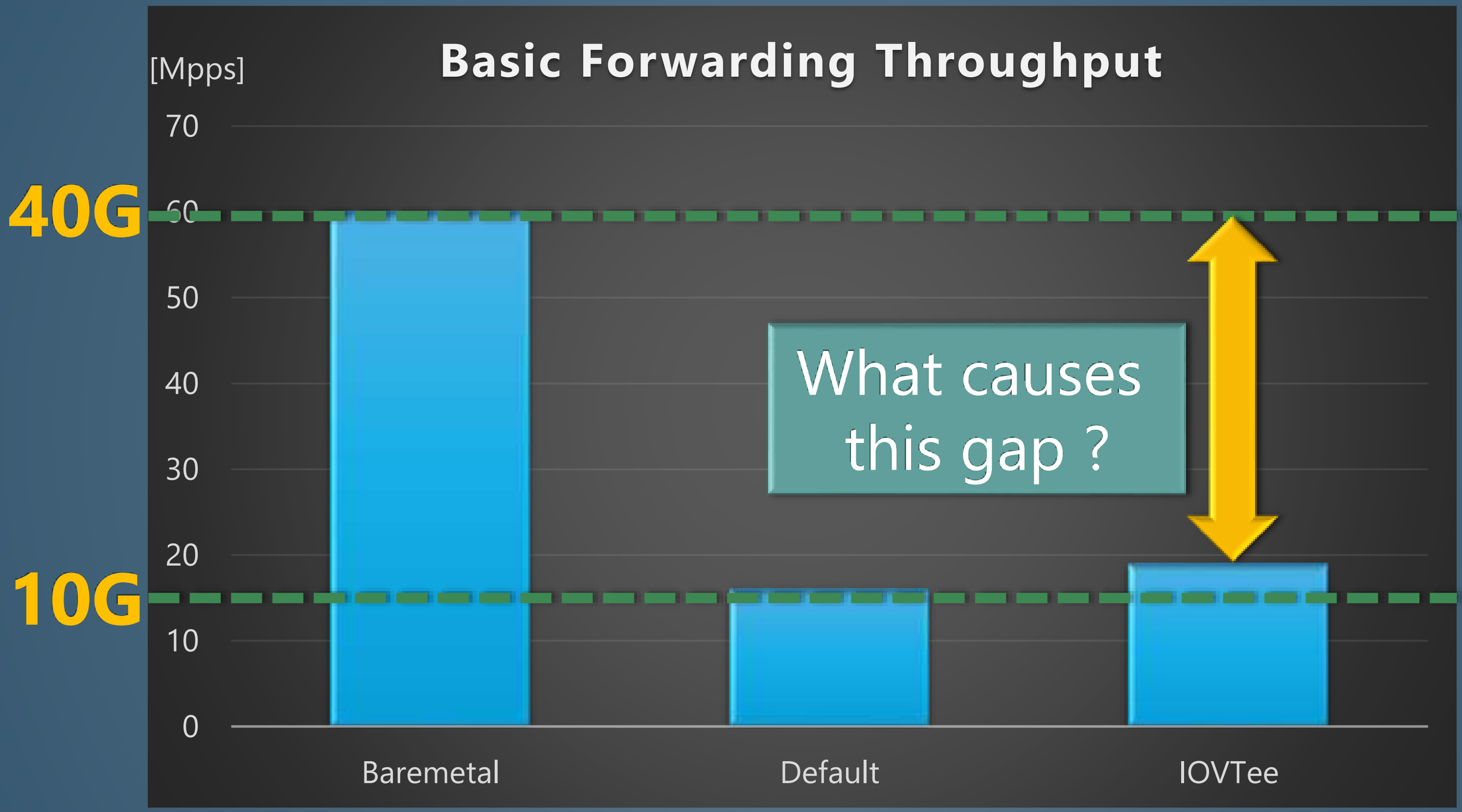
Current implementation could be further optimized

Exp3: Packet Sizes



IOVTee is superior for any packet size

Baremetal vs. Virtual Machine



(64-byte packets, Single datapath)

Zero-copy is effective, but not enough

Contents

Summary

IOVTee: A yet another Zero-copy/Pass-through method

Fast

19 Mpps (64-byte)
90 Gbps (1518-byte)

Pragmatic

Security Transparency
Portability Traceability
Container

Future Work

- Further optimizing current implementation
- Identifying actual performance bottleneck