VNET/P: Bridging the Cloud and High Performance Computing Through Fast Overlay Networking

Lei Xia, Zheng Cui, John Lange,
Yuan Tang, Peter Dinda, Patrick Bridges

Northwestern University
University of New Mexico
University of Pittsburgh
University of Electronic Science and Technology of China

http://v3vee.org

Overview

- Motivation: Bridging the cloud and HPC resources through virtual networking for HPC applications
  - Current virtual networking performance is NOT sufficient
- Design and optimization of **VNET/P**, a fast virtual overlay networking for such model
  - Applicable to other VMMs and virtual network systems
- Performance evaluation of VNET/P
  - Native/Near-native performance on 1Gbps/10Gbps networks

- *Possible* to extend software-based overlay networks into tightly-coupled environments

Outline

- Model and motivation
- VNET/P: design & optimization
- Performance evaluation
- Conclusions and future work

VNET Model

- A layer 2 virtual overlay network for the user's virtual machines
  - Provide location independence to VMs
  - Carry VMs' traffic via configurable overlay network
- Virtual machines on virtual networks as the abstraction for computing
- Virtual network as a fundamental layer for measurement and adaptation
  - *Monitor* application communication/computation behavior
  - *Adaptive* and *autonomic* mapping of virtual components to physical resources

A. Sundararaj, A. Gupta, P. Dinda, Increasing Application Performance In Virtual Environments Through Run-time Inference and Adaptation, HPDC'05

Bridging the Cloud and HPC

- Hosting HPC applications in VMs is possible
  - Low overhead in CPU/memory virtualization
- Extend *virtual overlay network* from loosely-coupled environments to ***tightly-coupled*** environments
- Seamlessly bridge the cloud and HPC resources
  - Applications can dynamically span to additional cloud resources
  - Virtual networking provides connectivity and mobility

- Performance of virtual overlay network is critical
  - How can it provide high performance inter-VM traffic while VMs are located on the ***same*** data center/cluster?

VNET/U

- VNET implemented at user-level
  - Among the fastest user-level overlay systems

    (78MB/s, 0.98ms)
  - Sufficient for wide-area/loosely-coupled applications
  - Throughput/latency limited by kernel/user transitions
  - Not sufficient for tightly-coupled applications running on cluster/supercomputer with gigabit or 10 gigabit networks

A. Sundararaj, P. Dinda, *Towards Virtual Networks for Virtual Machine Grid Computing*, VEE'04

J. Lange, P. Dinda, *Transparent Network Services via a Virtual Traffic Layer for Virtual Machines*, HPDC'07

VNET/P

- High performance virtual overlay network
  - Targeting for HPC applications in clusters and supercomputers with high performance networks
  - Also applicable to data centers that support IaaS cloud computing

- **High level approach**
  - **Move virtual networking directly into VMM**
  - **Enable optimizations that can only happen inside VMM**

Outline

- Model and motivation
- VNET/P: design & optimization
- Performance evaluation
- Conclusions and future work

Palacios VMM

- OS-independent embeddable virtual machine monitor

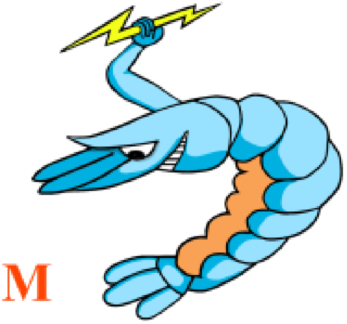
- Open source and freely available
- Host OS: **Linux**, Kitten, Minix …
- Successfully used on supercomputers, clusters (Infiniband and Ethernet), and servers

- VNET/P is in Palacios code base and is publicly available
- Techniques general applicable to other VMMs

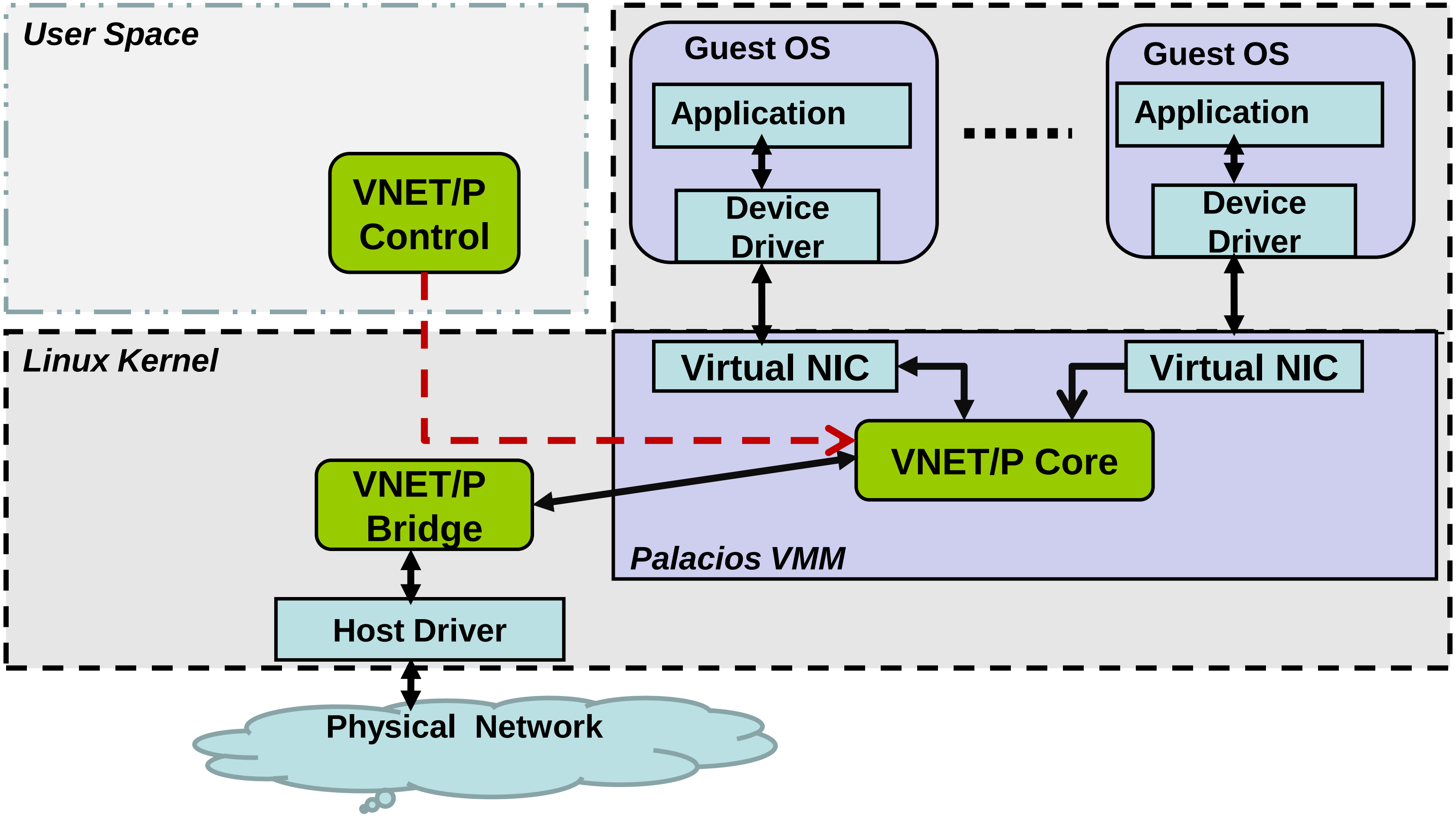
Palacios

An OS Independent Embeddable VMM

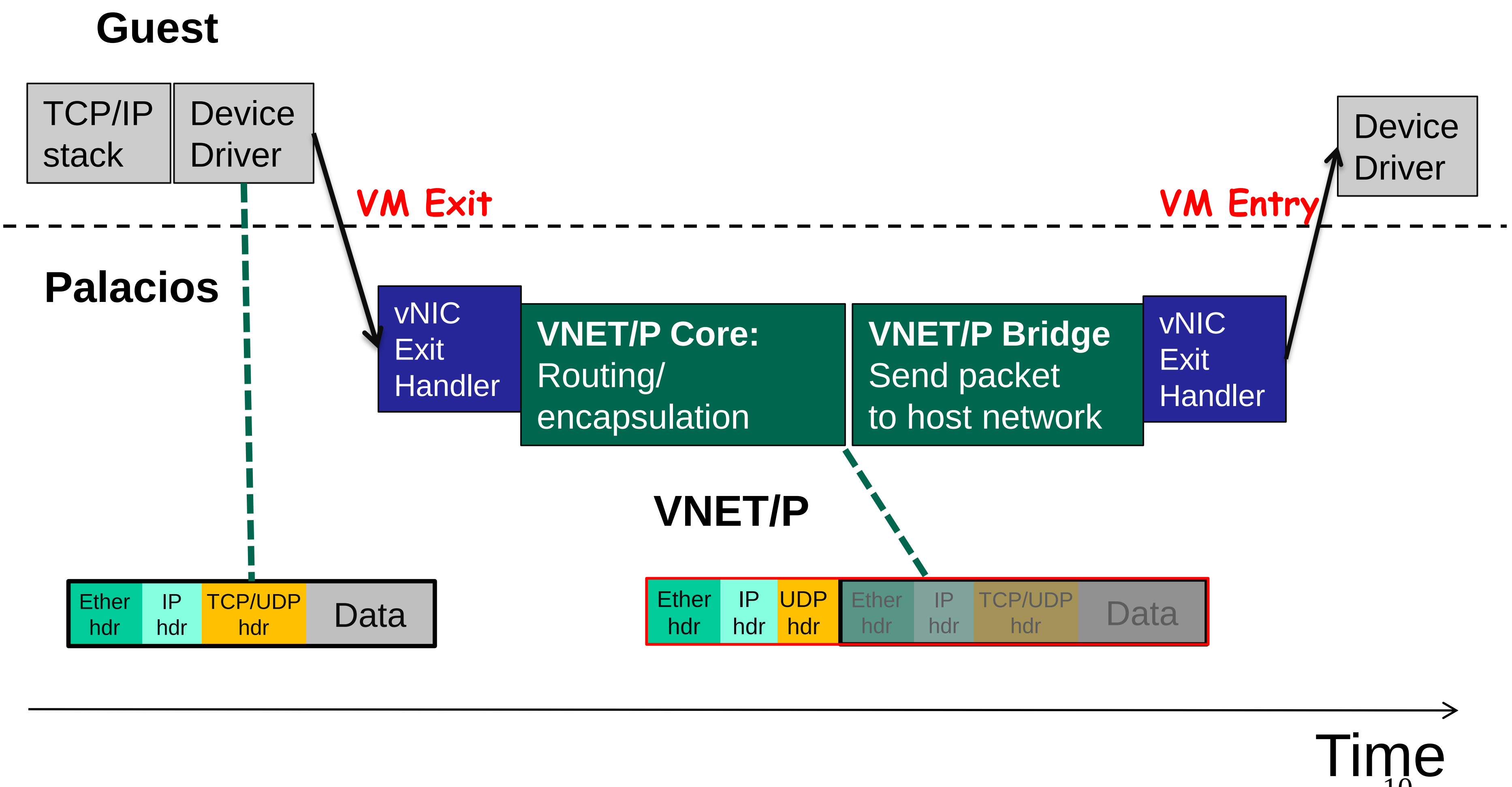
http://www.v3vee.org/palacios



VNET/P Architecture

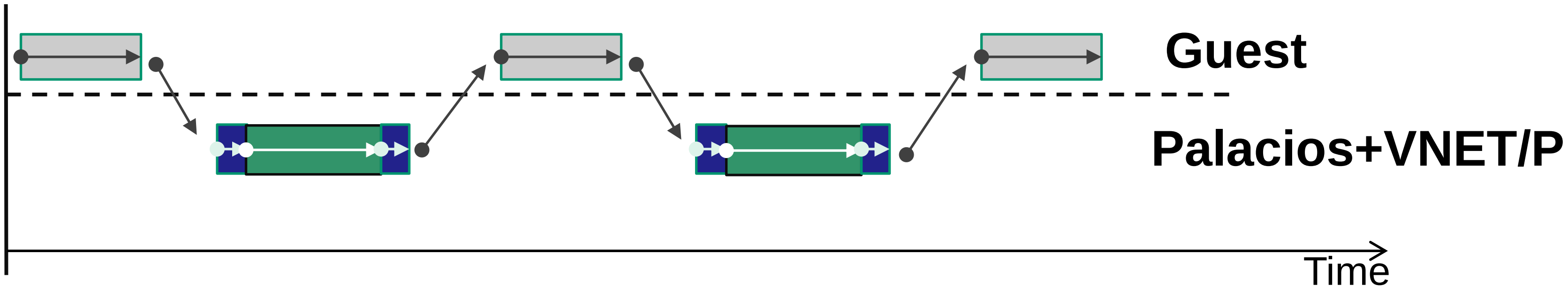


Data Path
(packet transmission)

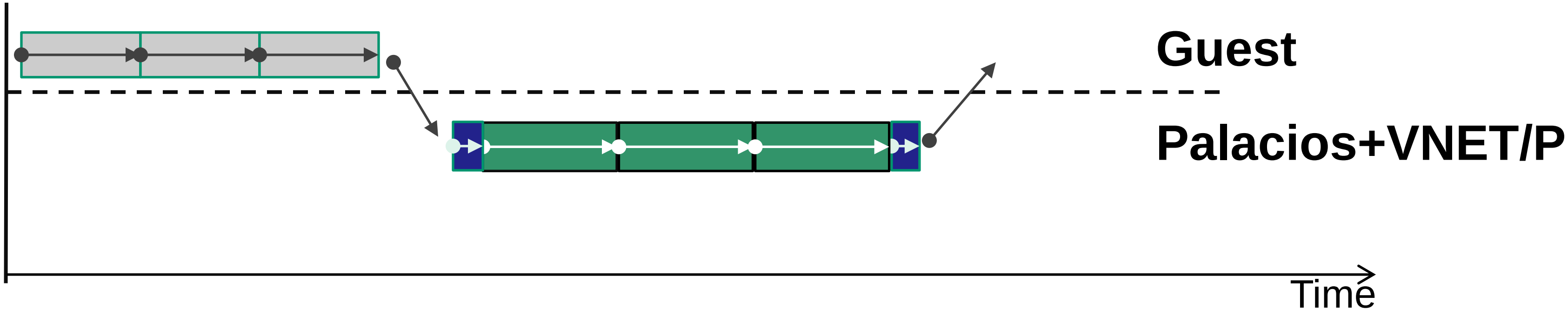


Transmission/Reception Modes

Guest-driven: Enhance latency



VMM-driven: Enhance throughput/Reduce CPU cost



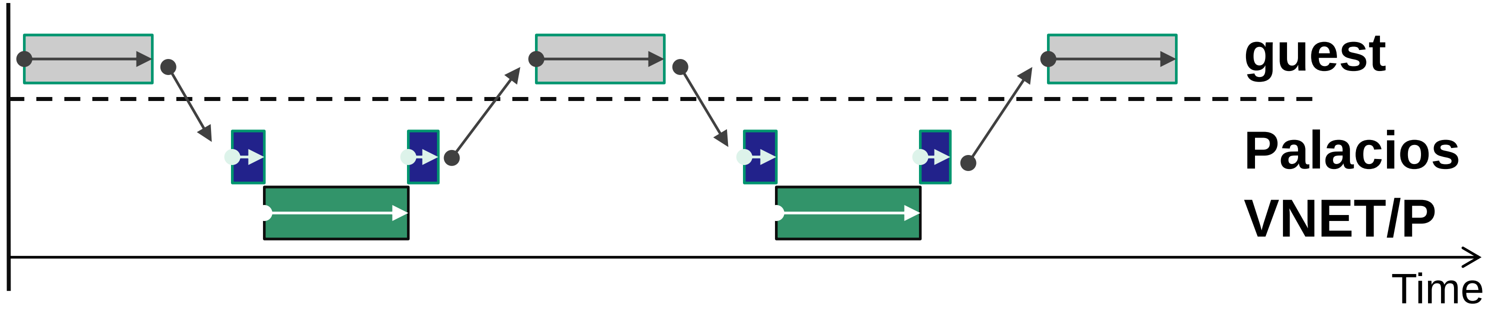
Dynamic Mode Switching

- VNET/P switches between two modes dynamically
  - Depends on the arrival rate of packets from VMs
    - Detected by exit rate due to virtual NIC accesses
  - Low rate: guest-driven mode to reduce the single packet latency
  - High rate: VMM-driven mode to increase throughput

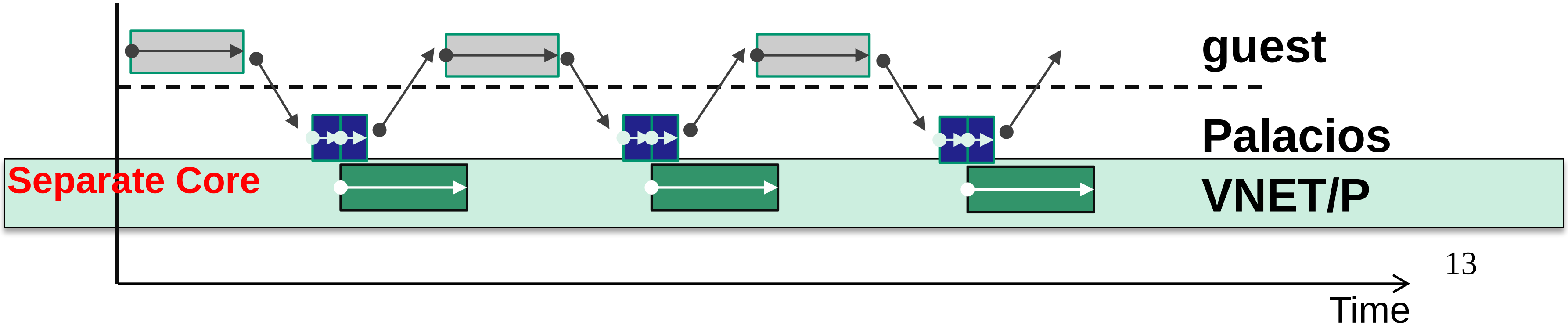
```
rate = # of exits for virtual NIC from last 10ms
if (rate >= THRESOLD && current-mode == GUEST-driven)
        current-mode= VMM-Driven;
else if (rate < THRESOLD && current-mode == VMM-driven)
        current-mode= GUEST-Driven;
else
        do-nothing;
endif
```

Packet Process Offloading Using Dedicated Thread

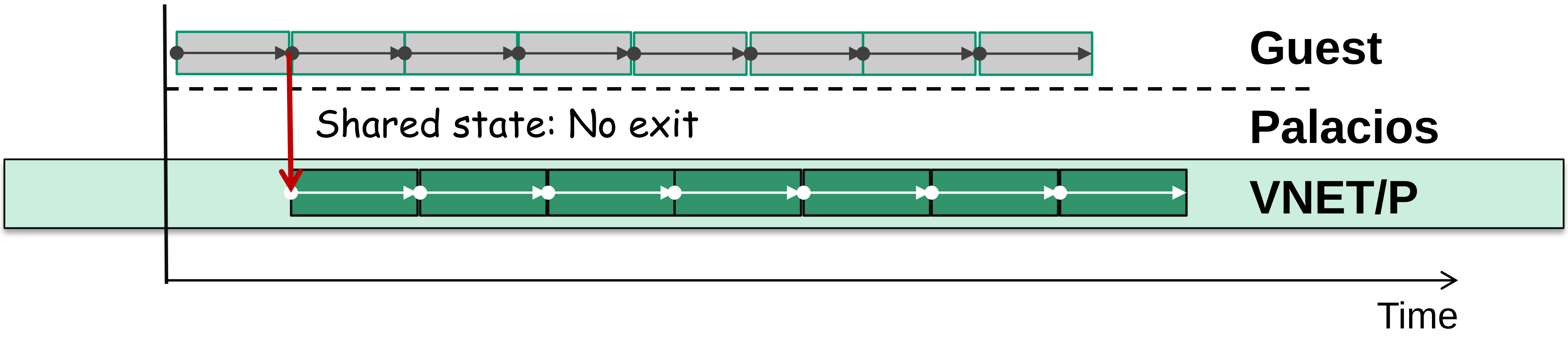
Baseline



Dedicated thread + guest-driven



VMM-driven + Dedicated Thread on Separate Core



- **High throughput mode avoids most exits**
- **VNET/P and Guest process packets in parallel**

Large MTU

- Larger MTU improves throughput and reduces CPU cost
  - Fewer packets are processed for a given amount of data.
  - VNET/P adds to the per-packet processing cost
- Guest MTU
  - Virtio NIC supports up to 64KB MTU
  - Most of other para-NICs support large MTU
- Host MTU
  - 10G usually supports jumbo MTU (9000Bytes)

Implementation

- Code size

Components	LoC
VNET/P Core	1955
VNET/P Bridge	1210
VNET/P Control Backend	1080
Virtio NIC Backend	987
Total	***5232***

  - Mostly VMM-independent code
  - Easy to port to other VMMs

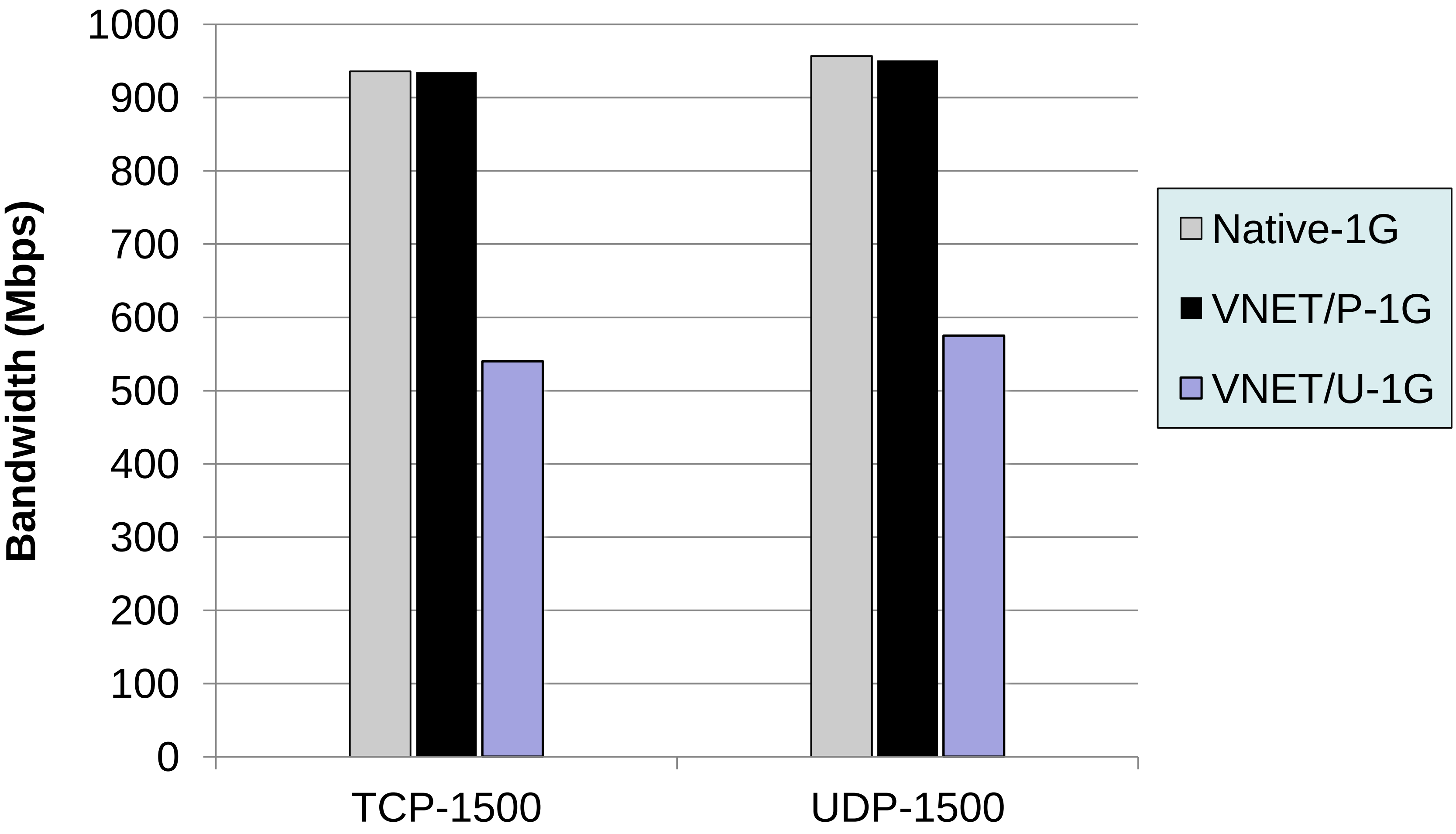
Outline

- Model and motivation
- VNET/P: design & optimization
- Performance evaluation
- Conclusions and future work

Performance Evaluation

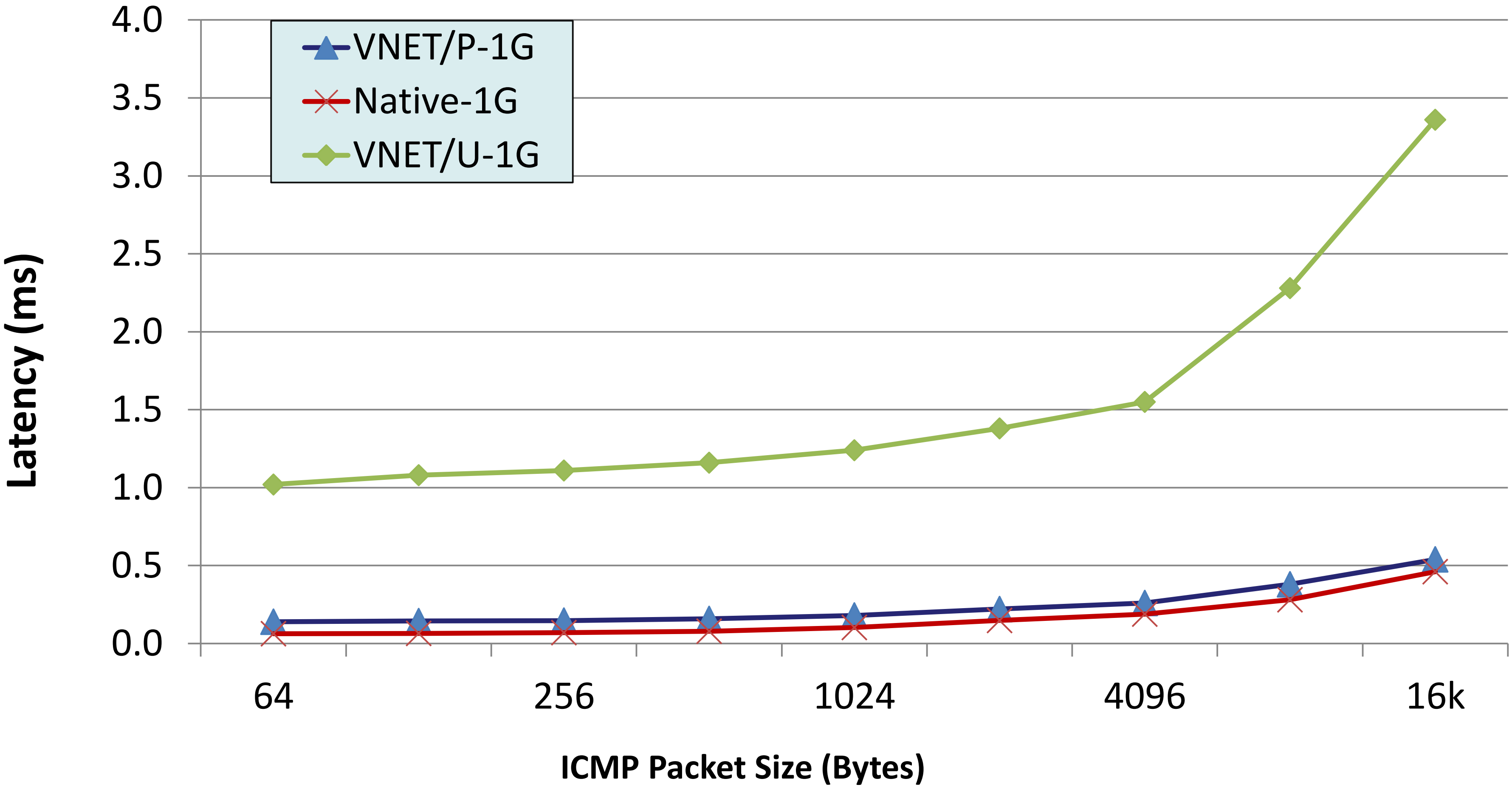
- Micro-benchmarks: Bandwidth and Latency
  - End-to-end performance
  - Multi-node performance
- Application Performance
  - NAS and HPCC
- Comparison
  - **VNET/P**: VMs with Linux and overlay, testing in guests
  - **Native**: Linux on hosts, no VMs, no overlay
  - **VNET/U**: VMs with user-level overlay

Native Bandwidth on 1Gb Network



VNET/P achieves native bandwidth

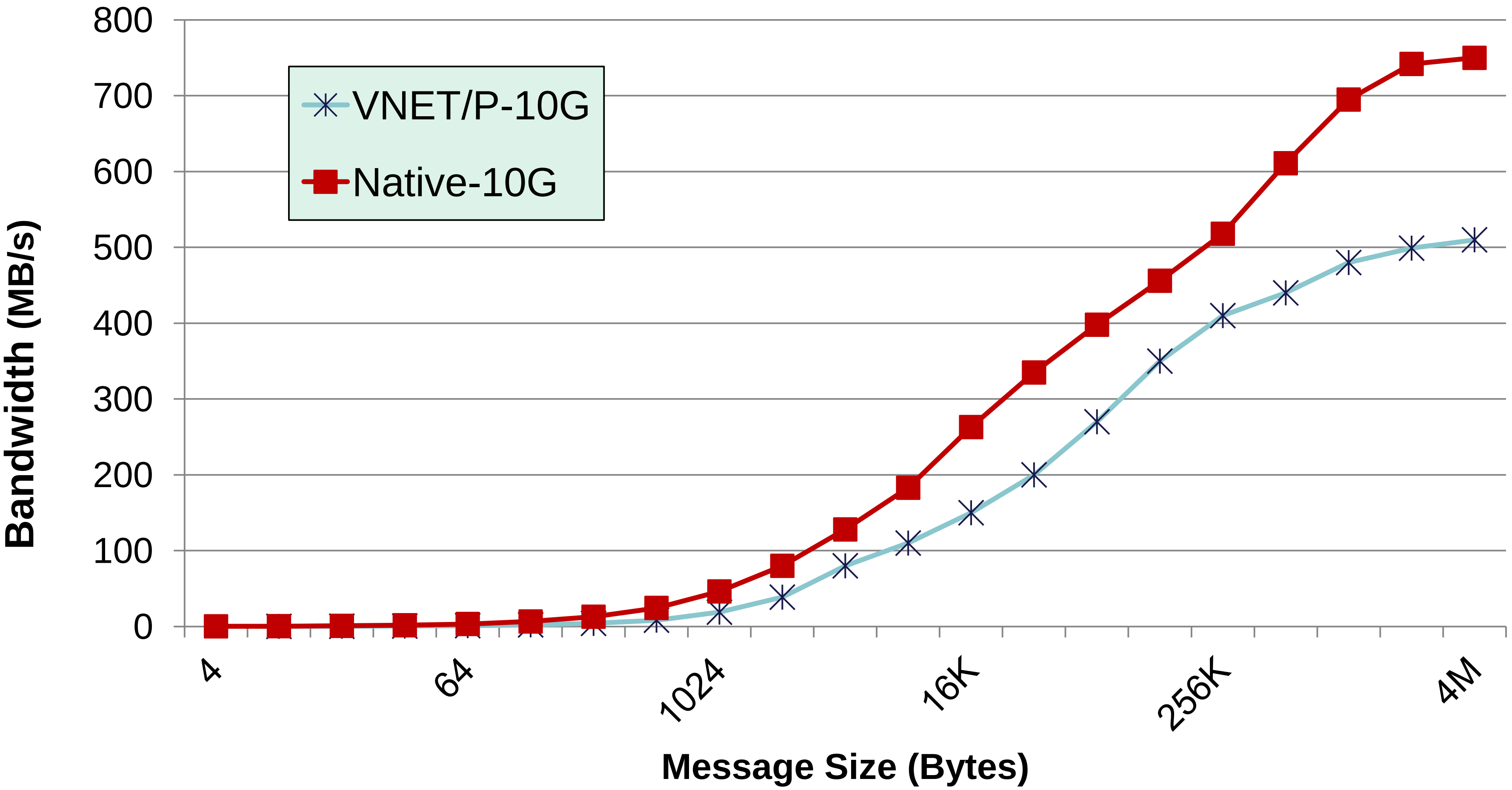
Near-native Round-trip Latency on 1Gb Network



Ping Round-trip Latency

VNET/P achieves 2 times of native latency

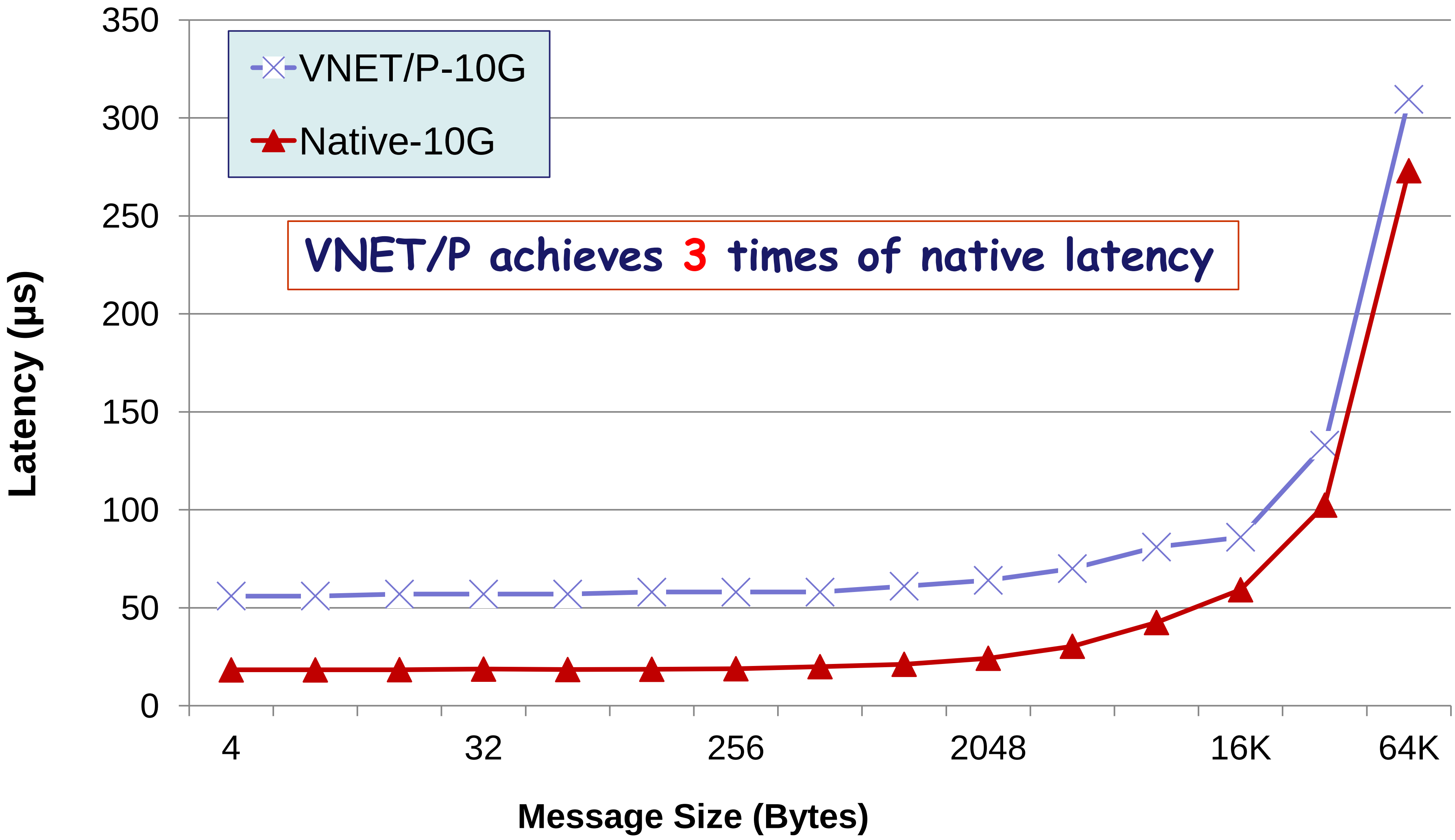
High Bandwidth on 10Gb Network



IMB One-way Bandwidth

VNET/P achieves around 65%-70% of native bandwidth

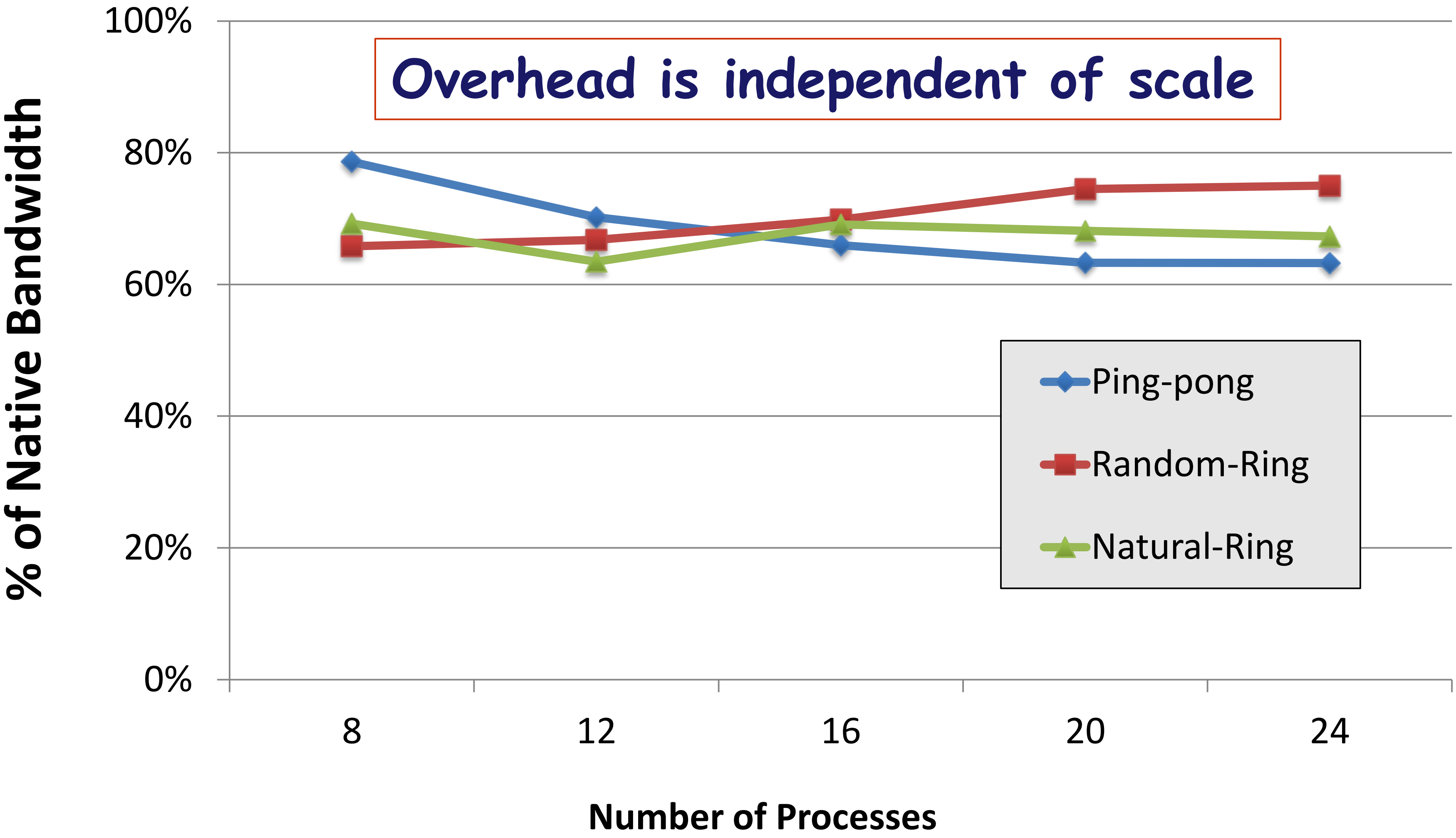
Low Latency on 10Gb Network



VNET/P achieves 3 times of native latency

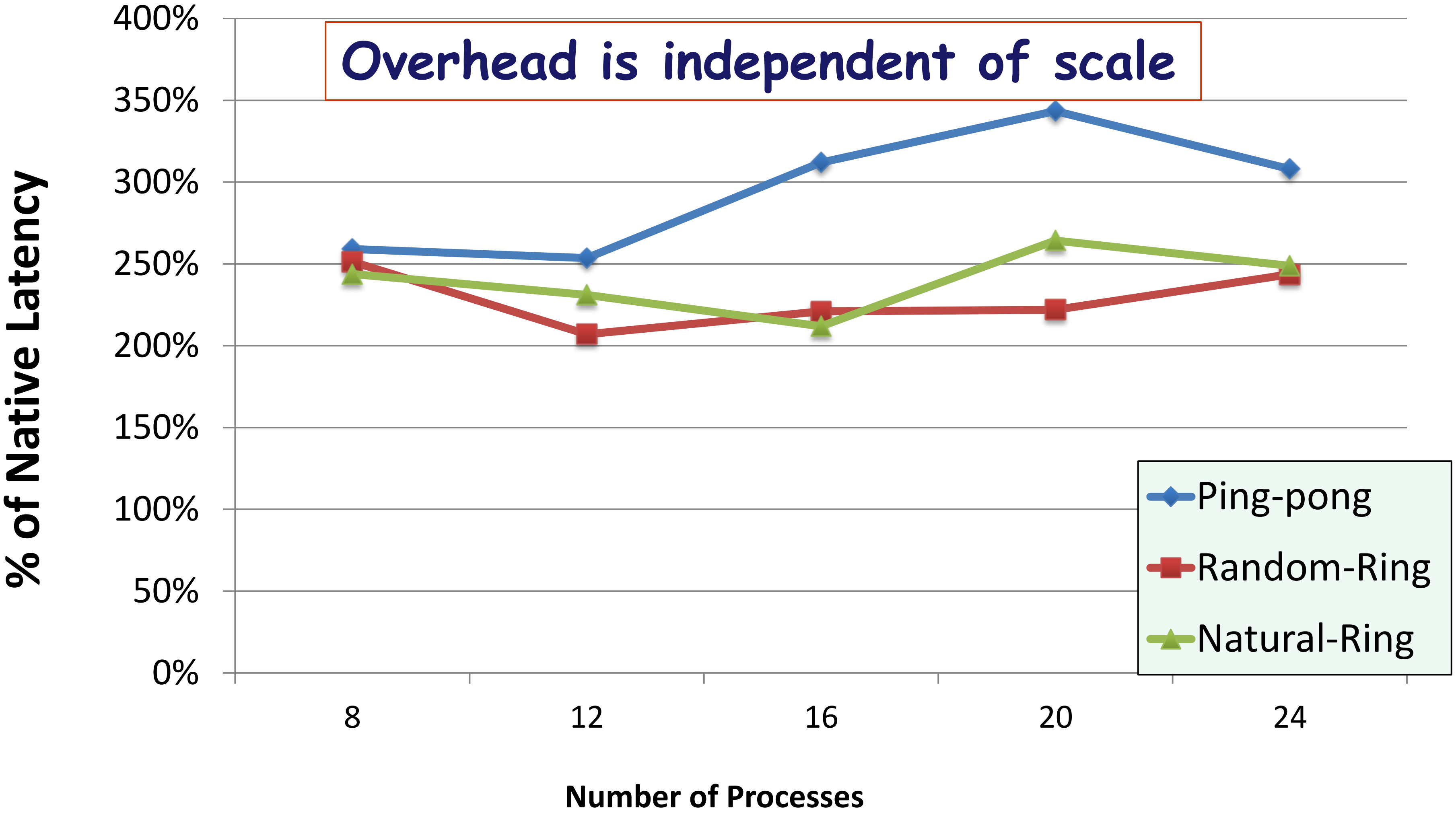
IMB Pingpong Latency

Scalable High Bandwidths (10Gbps)



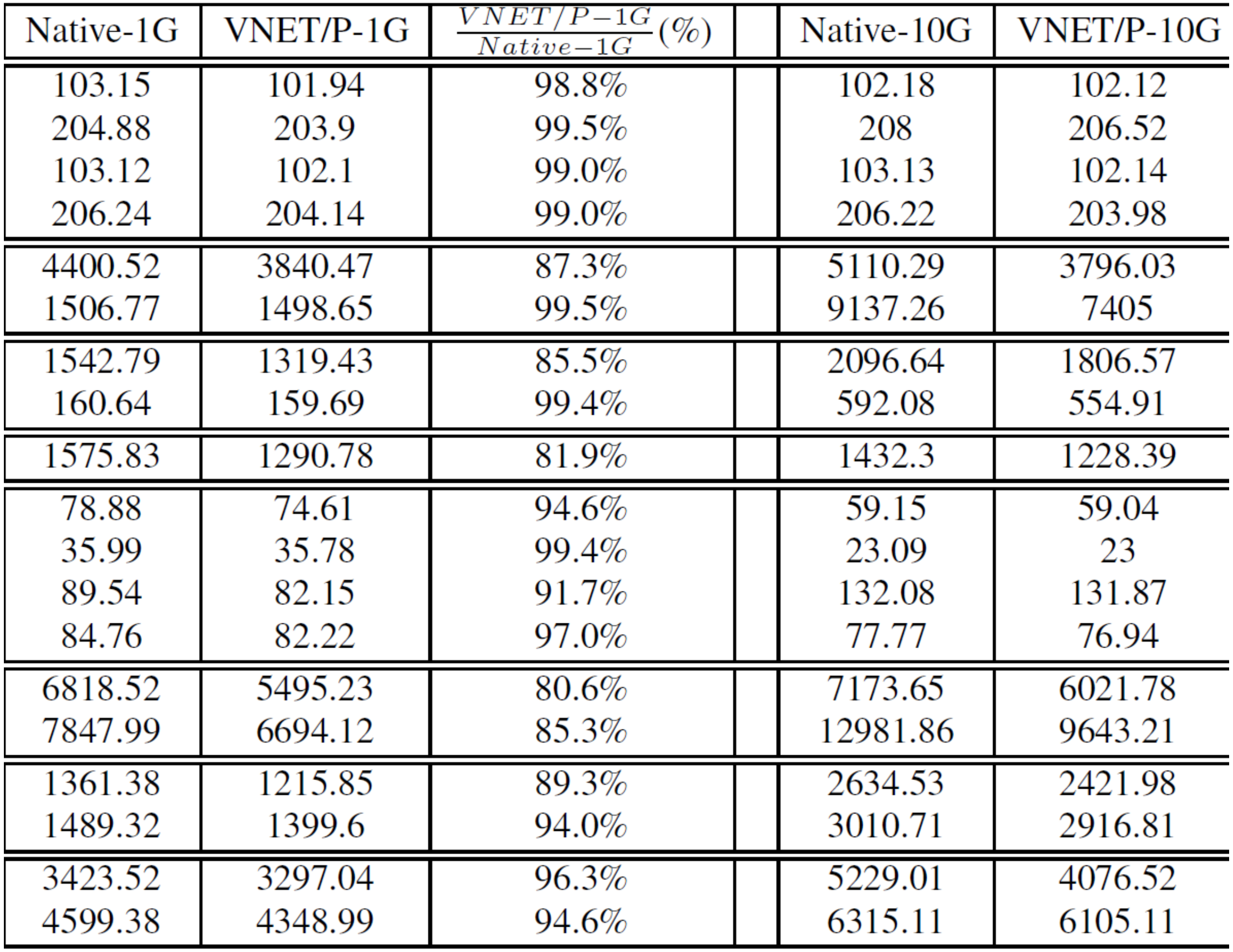
Bandwidth by HPCC Communication Benchmark

Scalable Low Latencies (10Gbps)



Latency by HPCC Communication Benchmark

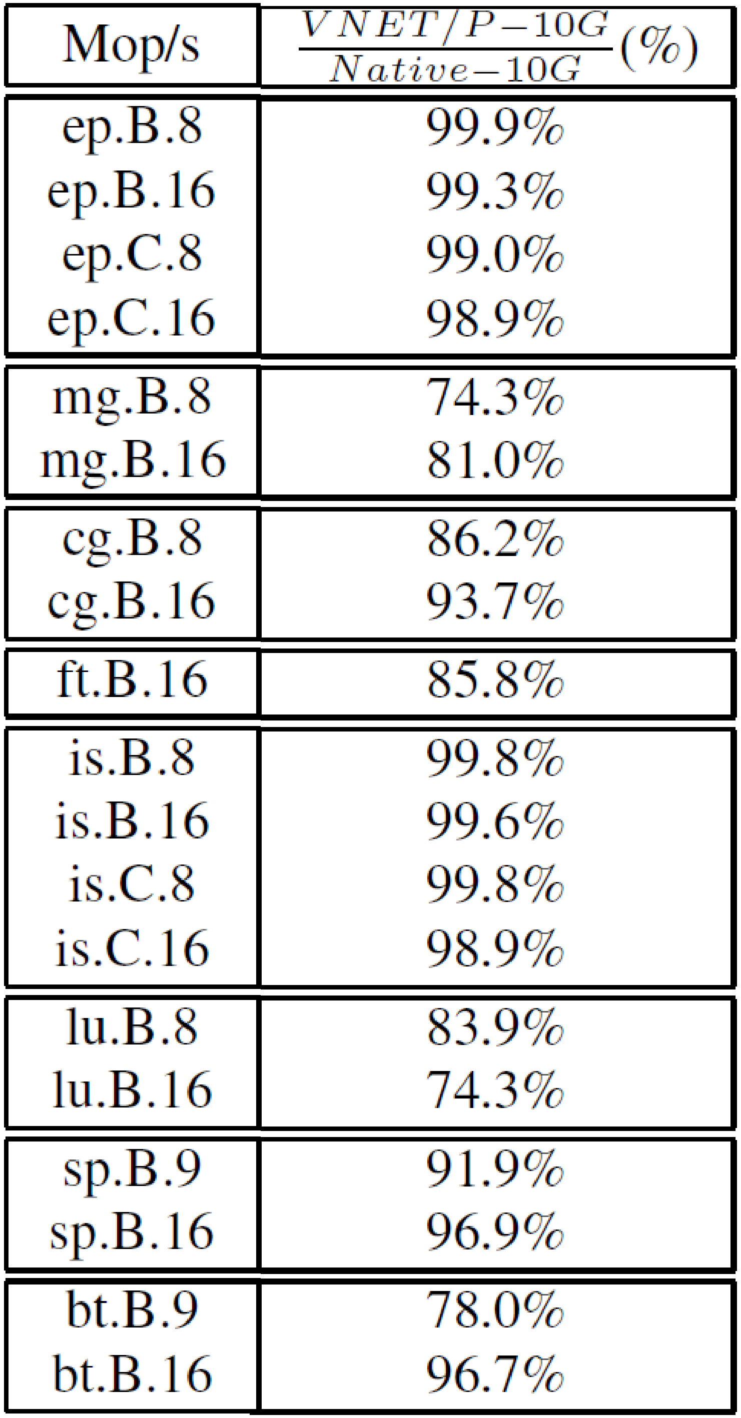
Application Performance



Native-1G	VNET/P-1G	$\frac{VNET/P-1G}{Native-1G}(\%)$		Native-10G	VNET/P-10G
103.15	101.94	98.8%		102.18	102.12
204.88	203.9	99.5%		208	206.52
103.12	102.1	99.0%		103.13	102.14
206.24	204.14	99.0%		206.22	203.98
4400.52	3840.47	87.3%		5110.29	3796.03
1506.77	1498.65	99.5%		9137.26	7405
1542.79	1319.43	85.5%		2096.64	1806.57
160.64	159.69	99.4%		592.08	554.91
1575.83	1290.78	81.9%		1432.3	1228.39
78.88	74.61	94.6%		59.15	59.04
35.99	35.78	99.4%		23.09	23
89.54	82.15	91.7%		132.08	131.87
84.76	82.22	97.0%		77.77	76.94
6818.52	5495.23	80.6%		7173.65	6021.78
7847.99	6694.12	85.3%		12981.86	9643.21
1361.38	1215.85	89.3%		2634.53	2421.98
1489.32	1399.6	94.0%		3010.71	2916.81
3423.52	3297.04	96.3%		5229.01	4076.52
4599.38	4348.99	94.6%		6315.11	6105.11

NAS Benchmarks

App ance



Mop/s	$\frac{VNET/P-10G}{Native-10G}(\%)$
ep.B.8	99.9%
ep.B.16	99.3%
ep.C.8	99.0%
ep.C.16	98.9%
mg.B.8	74.3%
mg.B.16	81.0%
cg.B.8	86.2%
cg.B.16	93.7%
ft.B.16	85.8%
is.B.8	99.8%
is.B.16	99.6%
is.C.8	99.8%
is.C.16	98.9%
lu.B.8	83.9%
lu.B.16	74.3%
sp.B.9	91.9%
sp.B.16	96.9%
bt.B.9	78.0%
bt.B.16	96.7%

Native-1G	VNE	VNET/P-10G
103.15	10	102.12
204.88	20	206.52
103.12	10	102.14
206.24	20	203.98
4400.52	384	3796.03
1506.77	149	7405
1542.79	131	1806.57
160.64	15	554.91
1575.83	129	1228.39
78.88	74	59.04
35.99	35	23
89.54	82	131.87
84.76	82	76.94
6818.52	549	6021.78
7847.99	669	9643.21
1361.38	121	2421.98
1489.32	13	2916.81
3423.52	329	4076.52
4599.38	434	6105.11

Outline

- Model and motivation
- VNET/P: design & optimization
- Performance evaluation
- Conclusions and future work

Future Work

- Further performance improvements
  - More specific optimizations to achieve native performance *(in submission)*
    - Optimistic interrupts,
    - Cut-through forwarding
    - Noise isolation
  - Move VNET up to guest through guest code injection *(to appear in ICAC'12)*
- Extend VNET/P on other high performance interconnects (Infiniband, SeaStar, etc)
  - Provide Ethernet abstraction for HPC application on different physical networks

VNET on Various Interconnects

- VNET on **InfiniBand**
  - Already works
  - Currently via IPoIB framework
    - 4.0Gbps bw/Native IPoIB 6.5Gbps
  - Pursuing high performance and leverage advanced hardware nature
- VNET over **Gemini**
  - In progress

Summary

- Current virtual networking is not fast enough for tightly-coupled environments
  - Bridge cloud and HPC resources for HPC applications
- VNET/P: high performance virtual overlay networking for tightly-coupled parallel systems
  - Overlay networking directly implemented into VMM
  - Native performance on 1Gb network
  - Close to native performance on 10Gb network
- Software-based overlay network can be extended into tightly-coupled environments

- Thanks, Questions??

- **Lei Xia**
  - Ph.D candidate, Northwestern University
  - *lxia@northwestern.edu*
  - *http://www.cs.northwestern.edu/~lxi990*

- V3VEE Project: http://v3vee.org