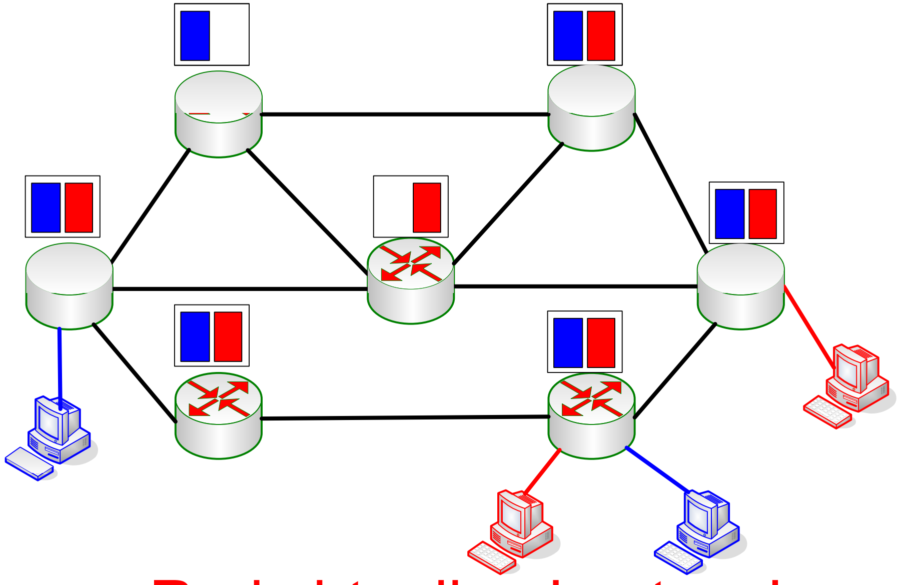
EUROPA: Efficient User-Mode Packet Forwarding in Network Virtualization

Yong Liao, Dong Yin, Lixin Gao

Univ. of Massachusetts, Amherst

Network Virtualization Platform

- Concurrent virtual networks running in the same substrate
  - – VINI, Trellis, VRouter
  - – NP, NetFPGA
- Requirements for such a platform
  - – Flexibility & isolation
  - – Performance



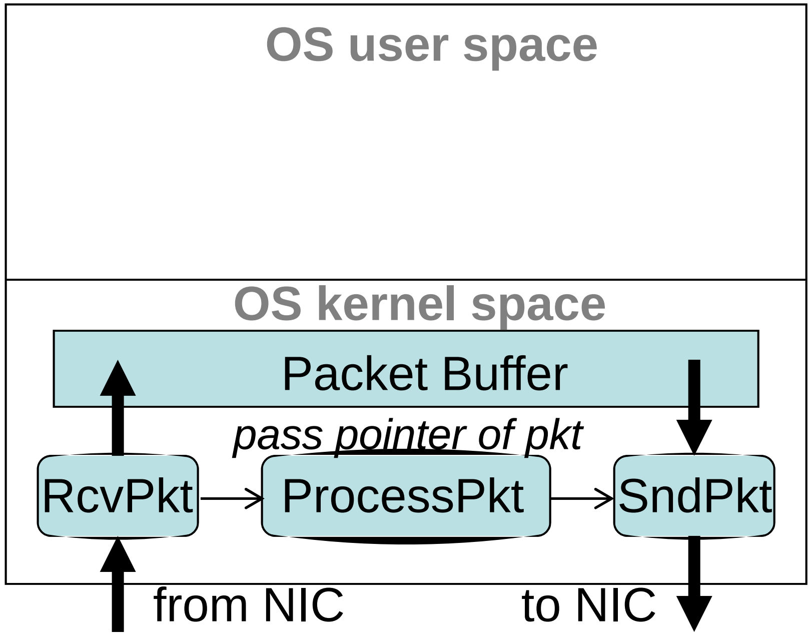
🟥 Red virtualized network

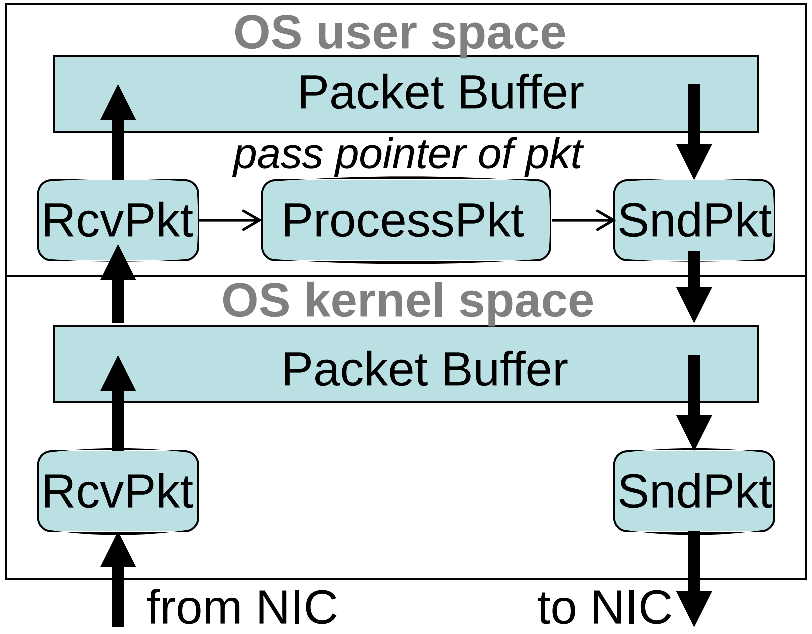
🟦 Blue virtualized network

Achieving the Design Goal

- Flexibility and isolation
  - User mode virtual router
- High-performance is challenging
  - Slow forwarding in VM
  - Overhead of running in user mode
- Solution
  - Efficient forwarding in user mode VM

Causes of Slow User Mode Forwarding





- Overhead
  - Copying packets between kernel and user space
  - Invoking system calls

Quantify the Overhead

- 2.66G CPU

System call	Send packet	Receive packet
CPU cycles	3,000	3,400

- Copying 64-byte packet

Memory copy	Copy to user	Copy from user
CPU cycles	160	140

- Overall overhead
  - 6,700 cycles per packet
  - No more than 400kpps in a 2.66G CPU

Avoiding the Overhead

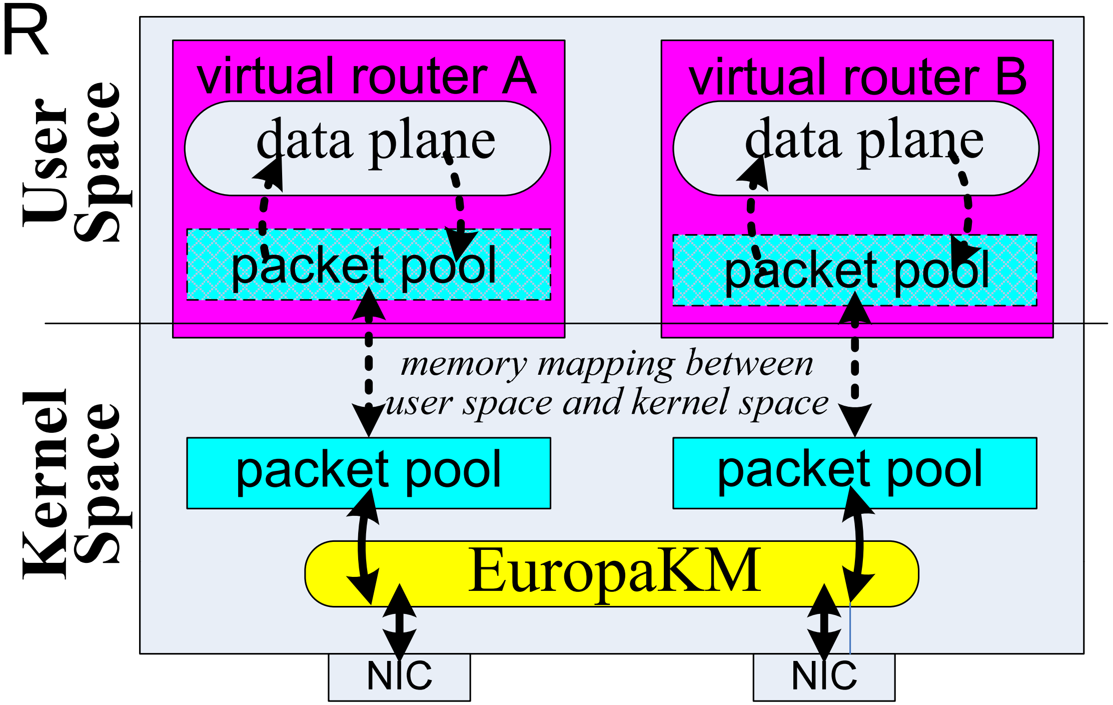
- Packet copying
  - Sharing buffer between kernel and user space virtual routers
- System calls
  - Kernel and user space virtual routers asynchronously access packet buffer
  - A “state” flag for each packet as mutex

Related Work in Improving I/O Performance

- Zero-copying schemes
  - Widely studied in OS community
  - Have not applied to virtual routers
- Polling for packets
  - Waiting for packets is expensive
  - Example: Click polling mode

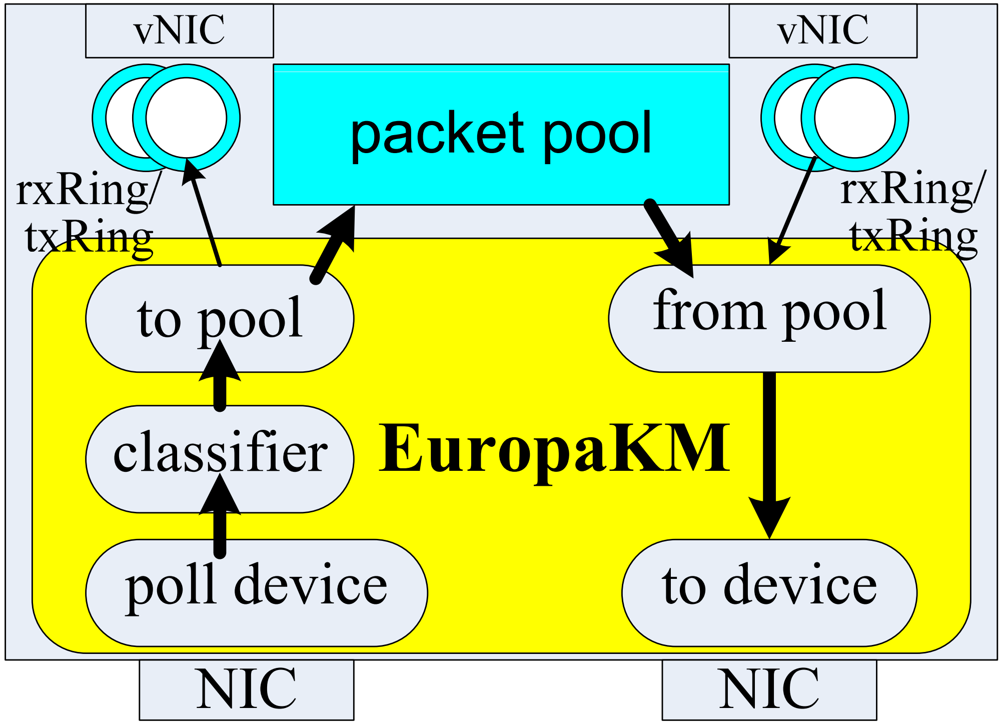
Europa Architecture

- Slicing server into VMs (OpenVZ)
  - VR data plane running in VM
- Shared packet pools (mmap mechanism)
  - One for each VR
- Europa kernel
  - One module serving for all VRs

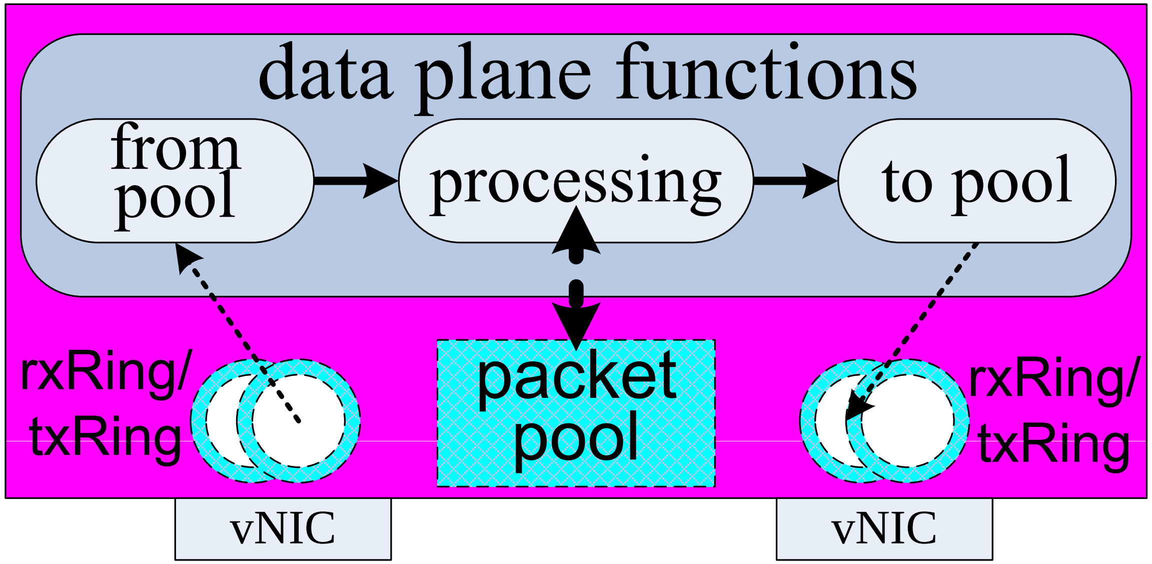


Europa Kernel Module

- Receiving packet
  - Classifying
  - Moving to VR’s pool
  - Writing index in rx ring
  - Changing packet state
- Sending packet
  - Reading index in VR’s tx ring
  - Polling packet state
  - Sending out via NIC



Europa Virtual Router



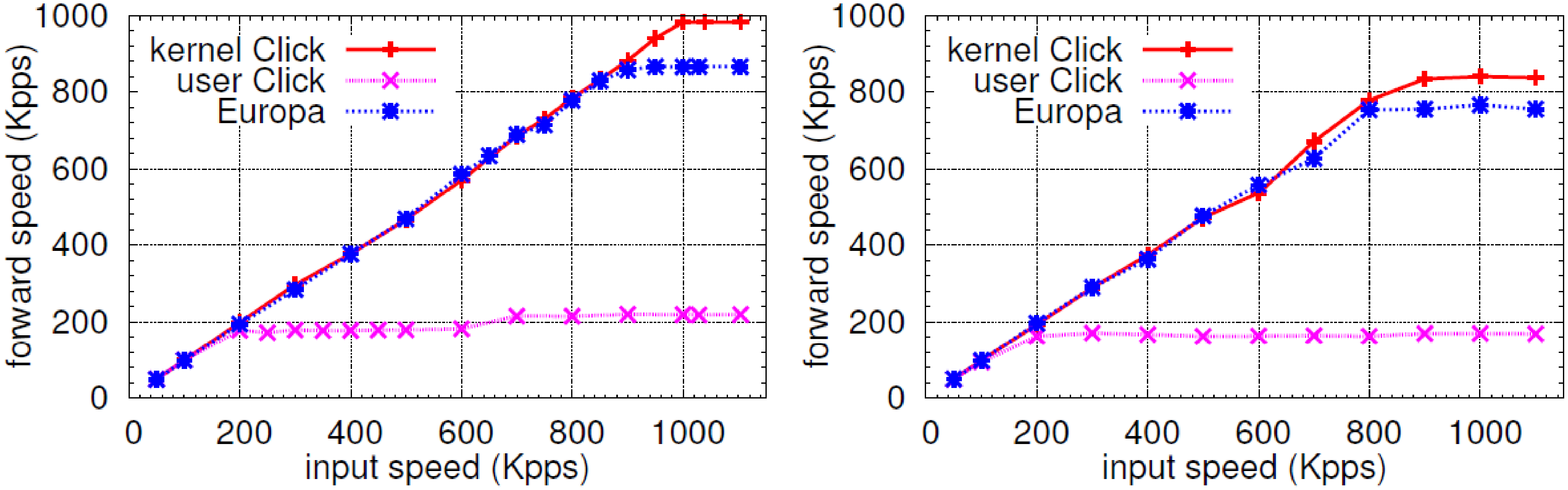
- Receiving packet
  - Reading index from VR's rx ring, polling packet state
- Processing packet
  - "In-place" processing
- Sending packet
  - Writing an index into VR's tx ring, changing packet state

Prototyping Implementation

- Hardware
  - Workstation PC with: 2.66G Core2 Duo CPU, 4G memory, Intel E1000 Gbit NIC
- Virtualization: OpenVZ
- Kernel module
  - Kernel Click: interact with NIC, classify packets, copy packets to VM’s shared buffer
  - Multi-threaded Click
- User module
  - User Click: process packet in an in-place manner

Evaluation: UDP Experiments

- Forwarding 64-byte UDP packets

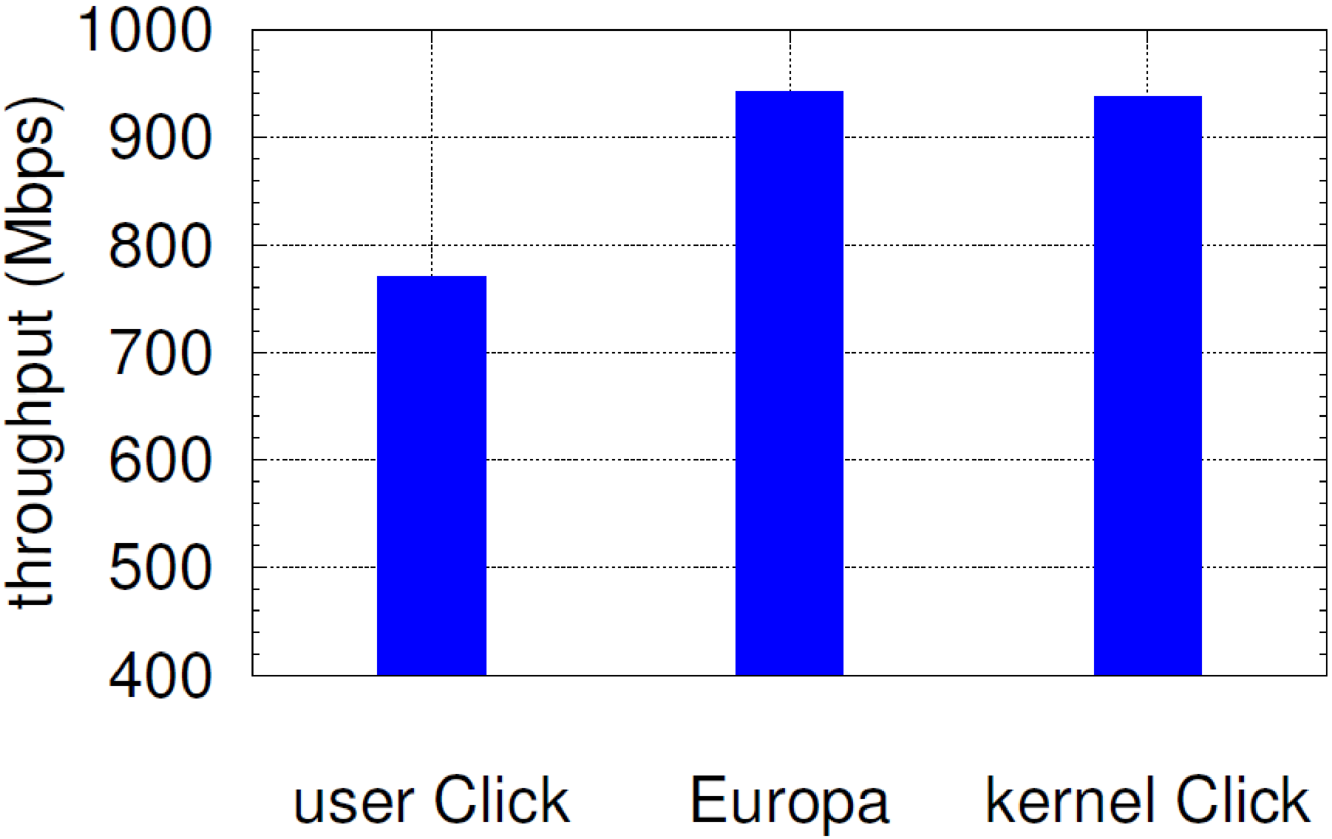


(a) small forwarding table (b) large forwarding table

- Europa matches kernel Click

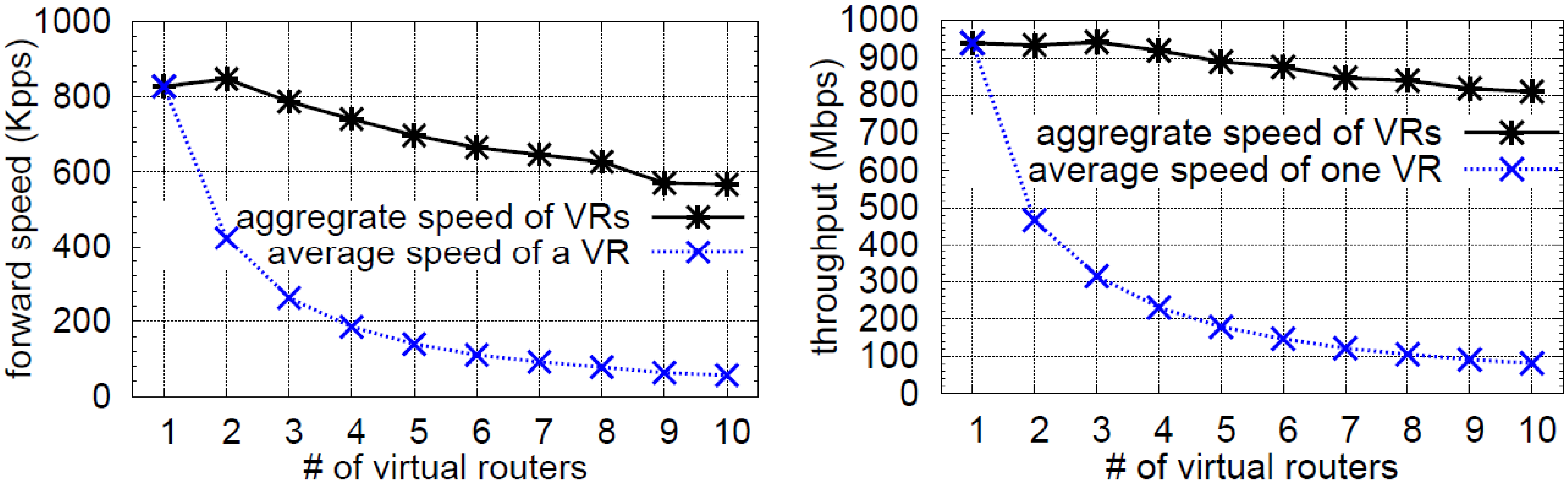
Evaluation: TCP Experiments

- TCP throughput



Scalability of Europa

- Multiple VRs concurrent forwarding



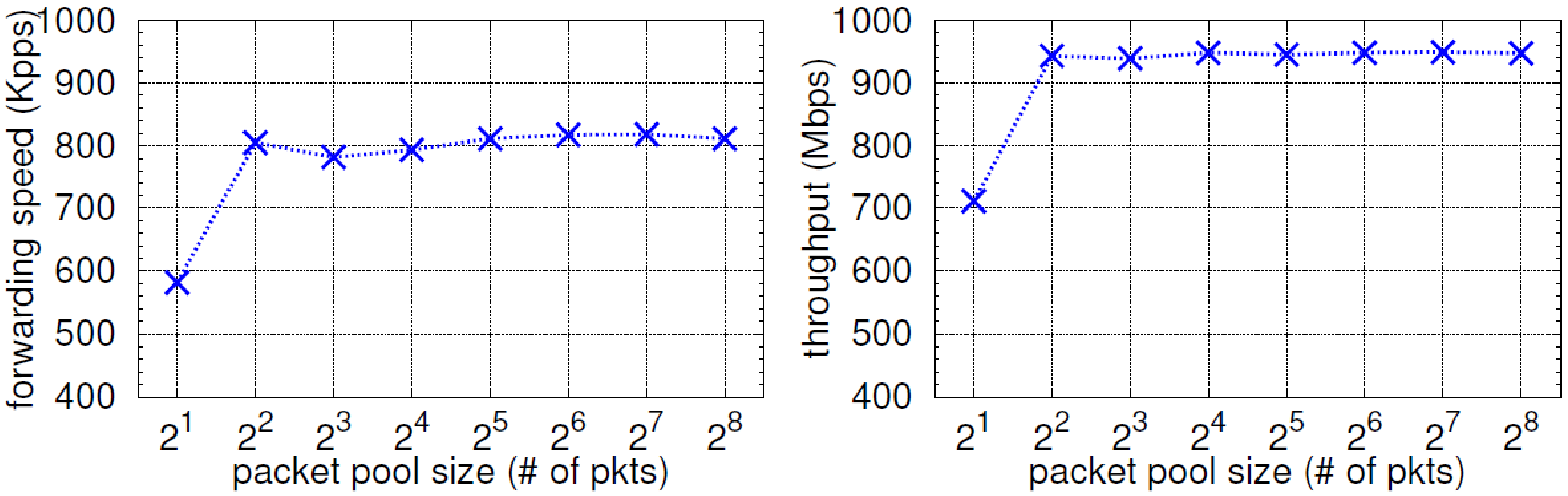
(a) UDP

(b) TCP

- More VRs, slower aggregate speed
  – Multi-core CPU

Pool Size vs. Forwarding Performance

- Two concurrent VRs



(a) UDP

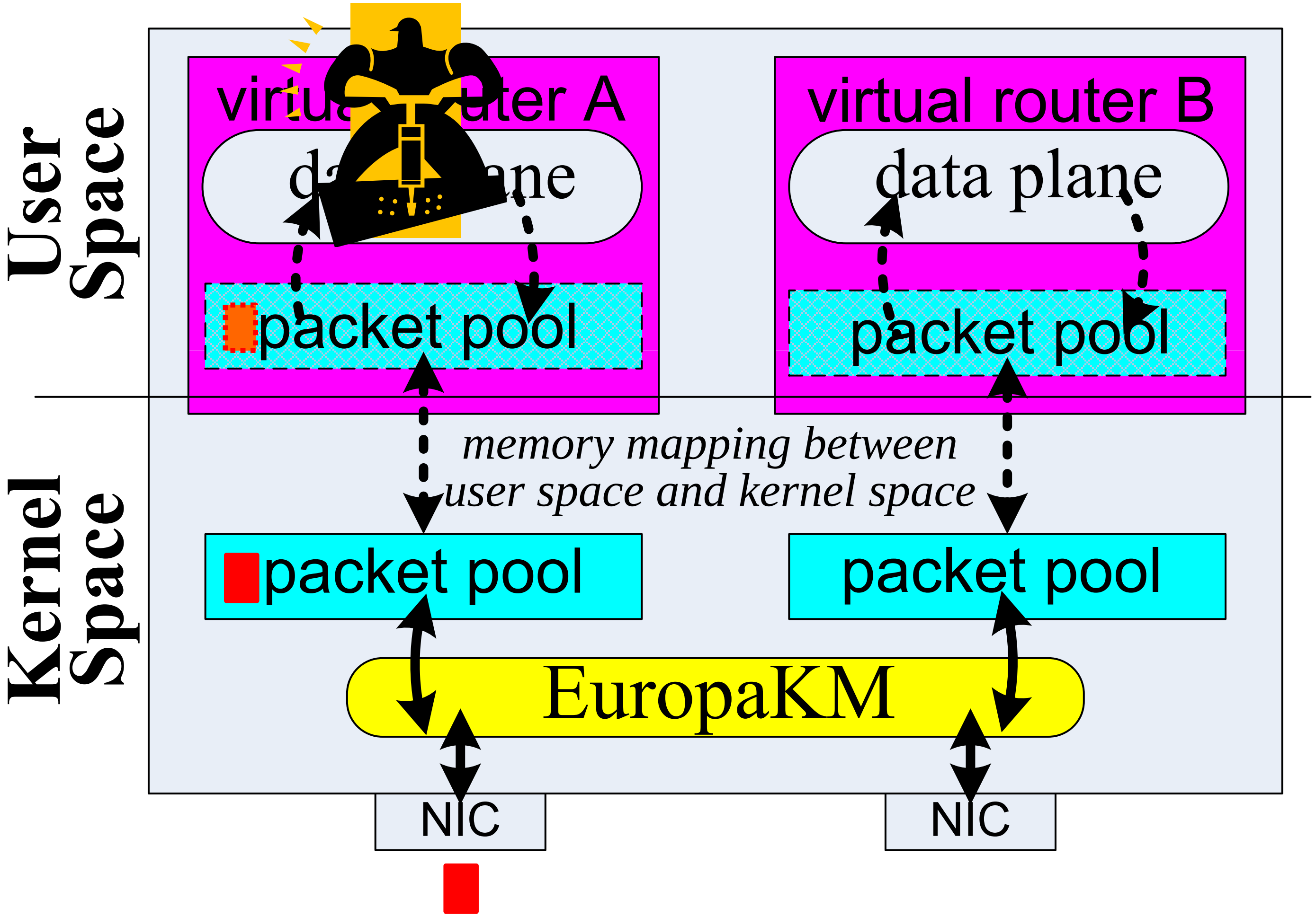
(b) TCP

- Speed is not sensitive to buffer size

Conclusion and Future Work

- Poor performance of conventional user mode packet processing
- Flexibility benefits of processing packet in user mode
- Europa achieves both flexibility and performance goal in virtual network
- Adaptive polling packets as future work

Packet Forwarding in Europa


User Space
Kernel Space
virtual router A
data plane
packet pool
virtual router B
data plane
packet pool
memory mapping between
user space and kernel space
packet pool
packet pool
EuropaKM
NIC
NIC