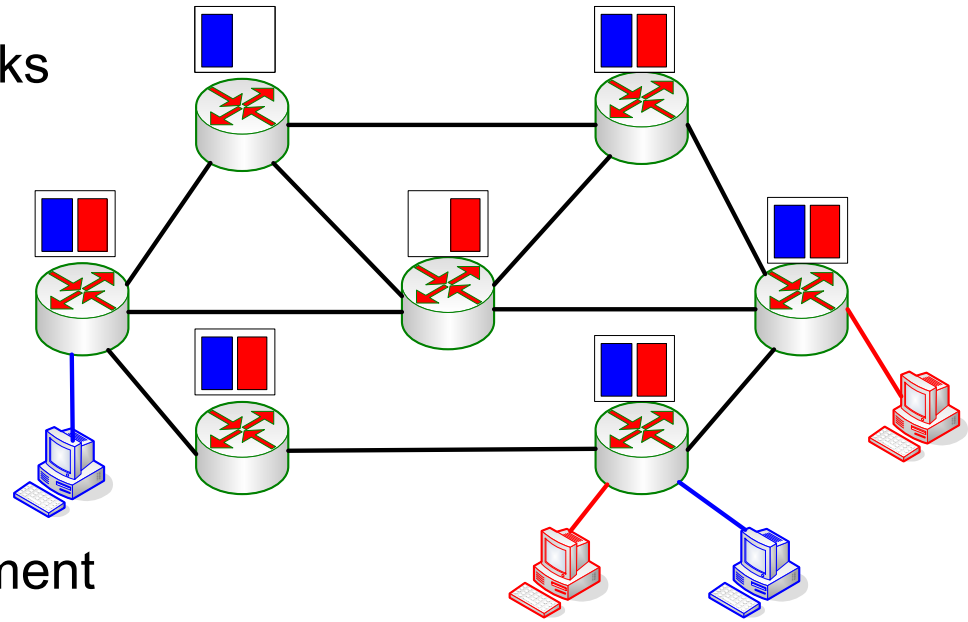
PdP: Parallelizing Data Plane in Virtual Network Substrate

Yong Liao, Dong Yin, Lixin Gao
University of Massachusetts at Amherst

Network Virtualization Platform

- Multiple heterogeneous concurrent virtual networks
  - Flexibility
    - Customizable virtual networks
  - High-performance
    - Good forwarding speed
  - Isolation
    - Minimal interference
  - Low-cost
    - Facilitate wide-area deployment



- Red virtualized network
- Blue virtualized network

Existing Network Virtualization Platforms

- VINI
  - User mode forwarding, slow, highly customizable
- Trellis
  - Kernel mode forwarding, faster, less customizable
- VRouter (Xen)
  - Close to native speed
  - Needs hardware support to scale
- Supercharging Planetlab
  - Special-purpose hardware, superior speed
  - Harder to program

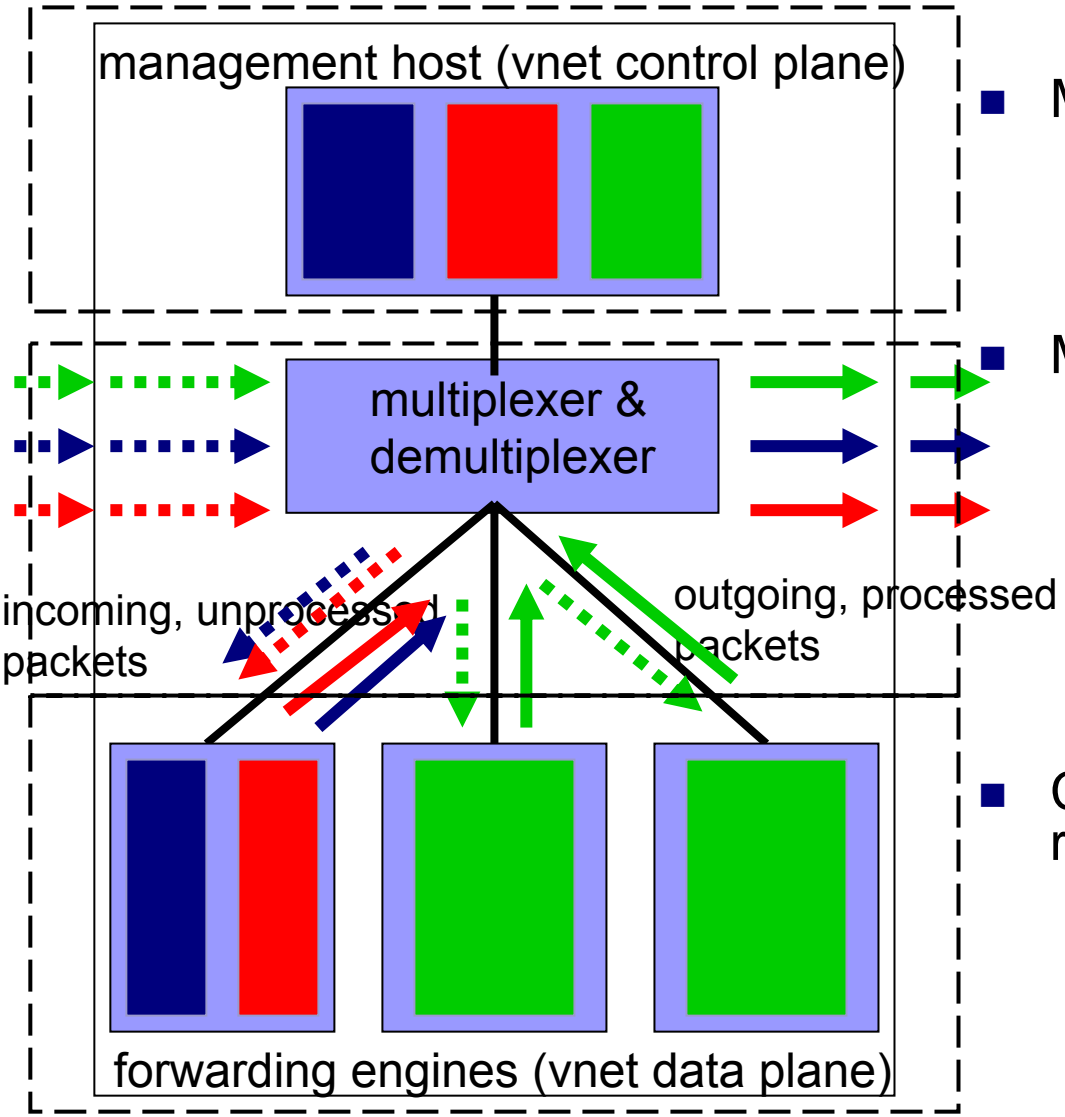
Existing Network Virtualization Platforms

	Flexibility	Performance	Isolation	Cost
VINI	Good	Slow forwarding	Good	Low cost
Trellis	Moderate	Close to native speed	Moderate	Low cost
VRouter	Good	Close to native speed	Good	High
SPP	Moderate	Superior speed	Moderate	High

Main Ideas of PdP

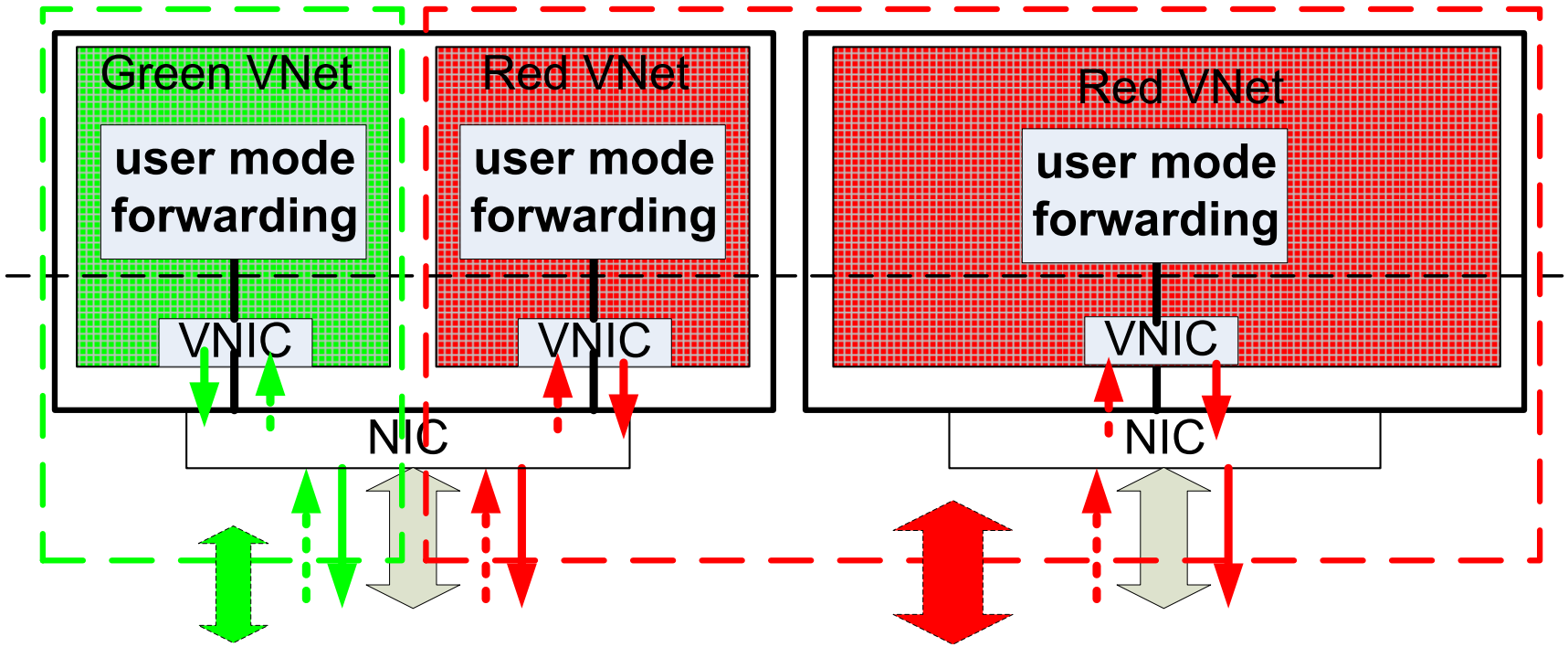
- Accelerate data forwarding with multiple forwarding engines
  - Faster aggregate forwarding speed
  - Commodity hardware is inexpensive
- Run virtual network data plane and control plane in VMs
  - Isolation among virtual networks
  - Better flexibility to customization

Architecture of PdP



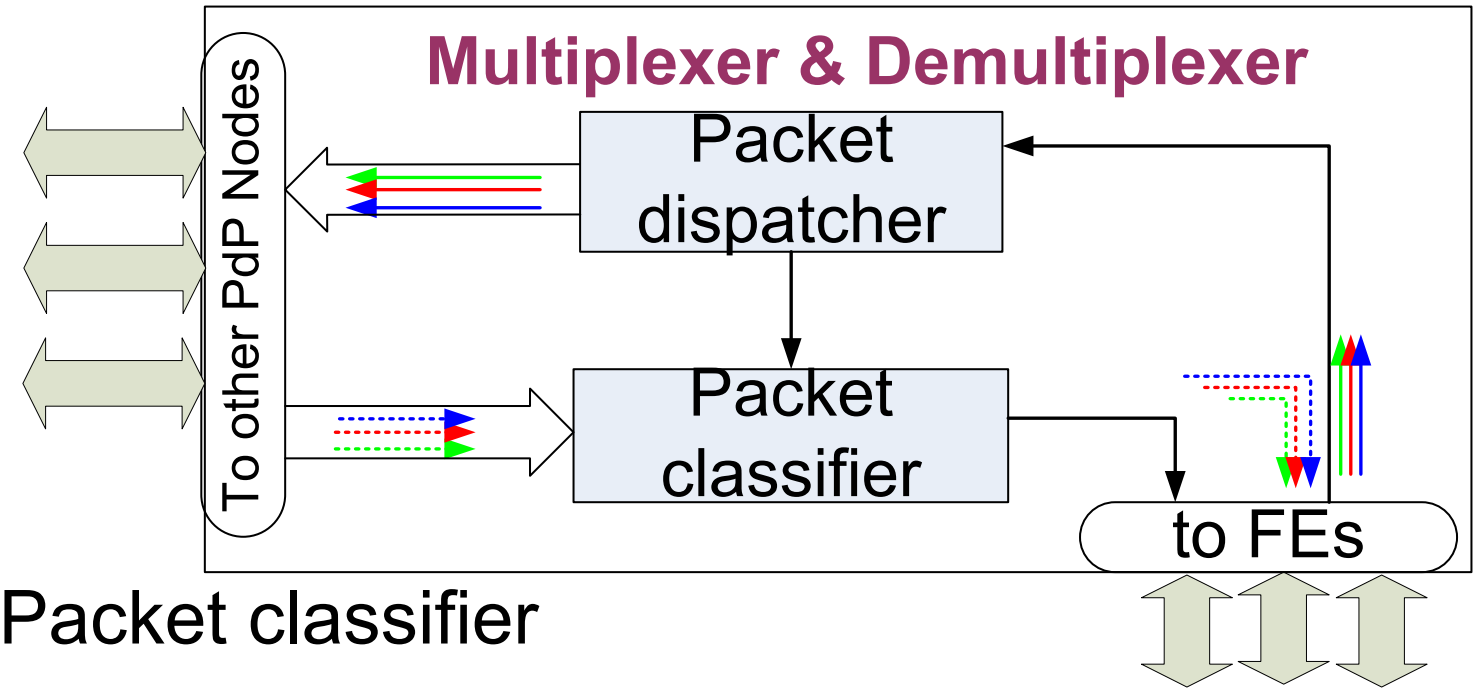
- Multiple forwarding engines (FEs)
  - Sliced into virtual nodes
  - Isolation
- Multiplexer & demultiplexer
  - Classify packets to data plane VMs
  - Send packets out to physical NICs
  - High-speed
- Control plane and data plane running in VMs
  - Customizable
  - Isolation and management

VNet Data Plane



- Mapping between VNet and Forwarding Engines
  - Multiple FEs for one VNet
  - How to allocate FEs to VNets
- Each virtual node performs (in user mode)
  - Lookup, encapsulation for virtual links

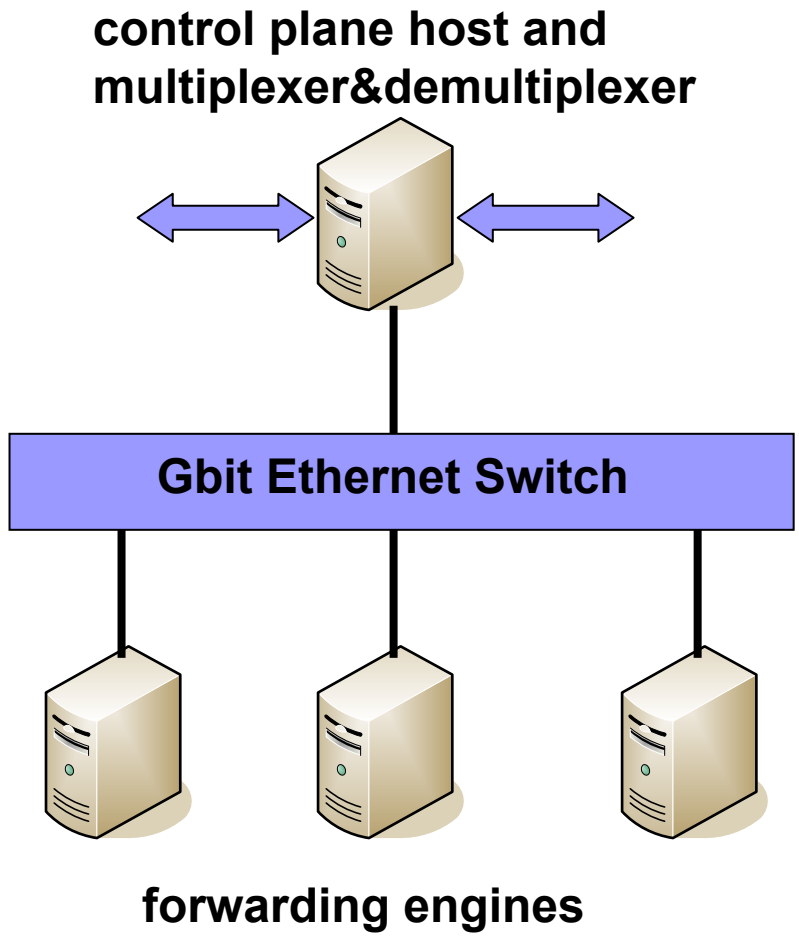
Multiplexer & Demultiplexer



- Packet classifier
  - Different ports for different VMs
- Packet dispatcher sends packets out
  - FE already marked the outgoing NIC
- Can potentially be bottleneck

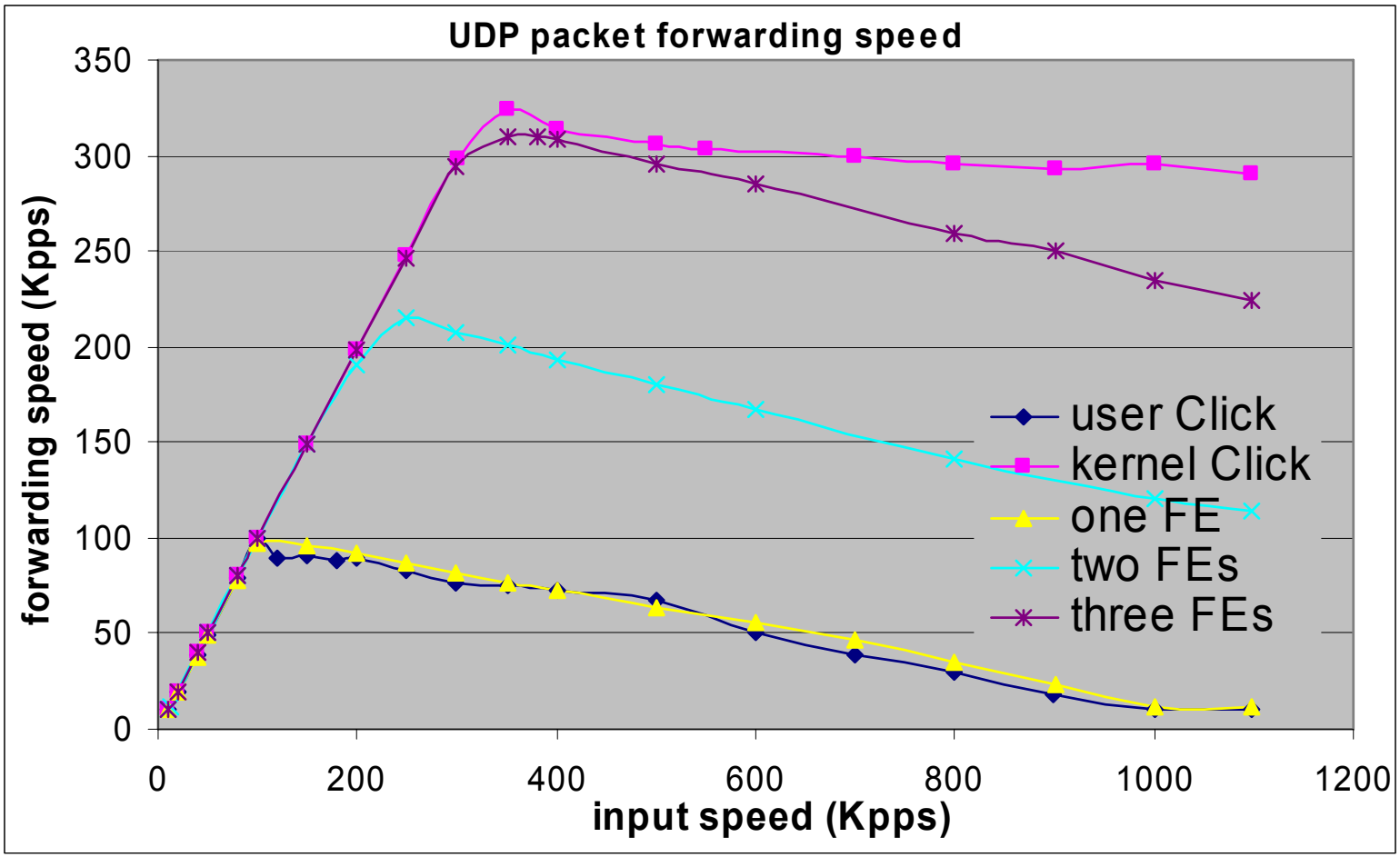
Prototype Implementation

- Commodity PCs
  - P4 2.6GHz CPU, 1G mem, Gbit NIC
- Multiplexer & demultiplexer
  - Kernel mode Click
- VNet Data plane
  - User mode Click in VM
- VNet Control plane
  - XORP in VM
- Interaction between VNet control plane and data plane
  - Updating forwarding table by multicast

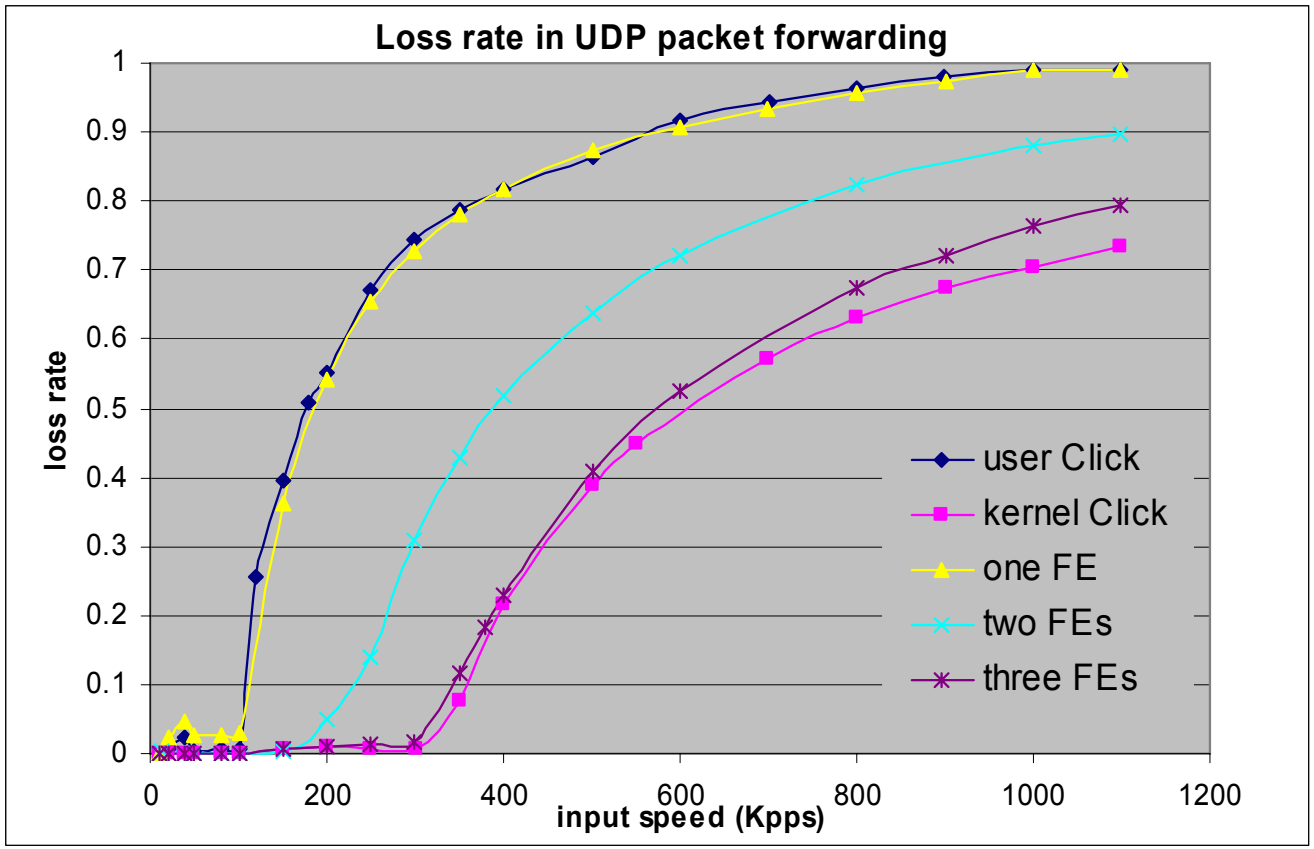


Raw UDP packet forwarding Speed

Small table: two entries, Similar for large table



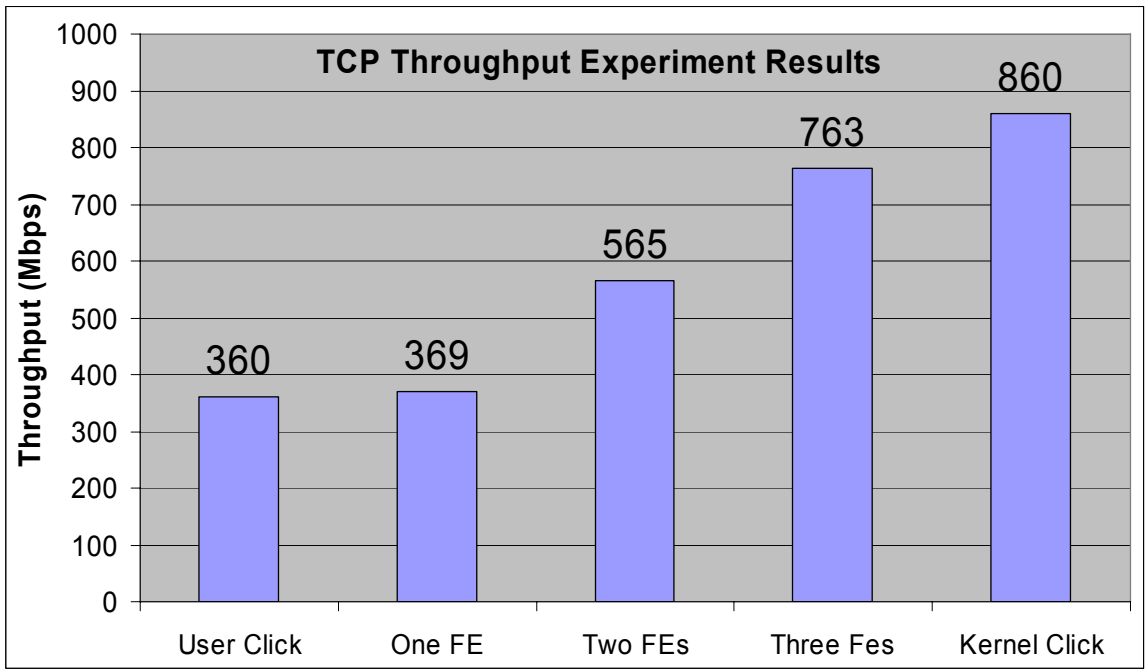
Raw UDP Packet Loss Rate and RTT



	User Click	PdP	Kernel Click
Two-hop RTT(ms)	0.208	0.296	0.132

TCP Performance

- Aggregate throughput is close to kernel Click
- Out-of-order packets



	one FE	two FEs	three FEs
% of out-of-order pkts	0.31%	10.19%	13.02%

	round-robin	proportional
% of out-of-order pkts	12.27%	10.02%

Conclusion and Future Work

- PdP provides the maximal flexibility to customize VNets
- Forwarding speed of PdP scales with the number of FEs

- Hardware multiplexer/demultiplexer
- Flow based classification (out-of-order packet problem)