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

<table>
<thead>
<tr>
<th></th>
<th>Flexibility</th>
<th>Performance</th>
<th>Isolation</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>VINI</td>
<td>Good</td>
<td>Slow forwarding</td>
<td>Good</td>
<td>Low cost</td>
</tr>
<tr>
<td>Trellis</td>
<td>Moderate</td>
<td>Close to native speed</td>
<td>Moderate</td>
<td>Low cost</td>
</tr>
<tr>
<td>VRouter</td>
<td>Good</td>
<td>Close to native speed</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>SPP</td>
<td>Moderate</td>
<td>Superior speed</td>
<td>Moderate</td>
<td>High</td>
</tr>
</tbody>
</table>
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 V Nets

- 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

Loss rate in UDP packet forwarding

<table>
<thead>
<tr>
<th>Two-hop RTT(ms)</th>
<th>User Click</th>
<th>PdP</th>
<th>Kernel Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.208</td>
<td>0.296</td>
<td>0.132</td>
<td></td>
</tr>
</tbody>
</table>
TCP Performance

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

<table>
<thead>
<tr>
<th>Throughput (Mbps)</th>
<th>User Click</th>
<th>One FE</th>
<th>Two FEs</th>
<th>Three Fes</th>
<th>Kernel Click</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>360</td>
<td>369</td>
<td>565</td>
<td>763</td>
<td>860</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Percentage of out-of-order pkts</th>
<th>one FE</th>
<th>two FEs</th>
<th>three FEs</th>
<th>round-robin</th>
<th>proportional</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of out-of-order pkts</td>
<td>0.31%</td>
<td>10.19%</td>
<td>13.02%</td>
<td>12.27%</td>
<td>10.02%</td>
</tr>
</tbody>
</table>
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)