**LATENCY IN SDN**

Timely interaction between an SDN controller and switches is crucial to many applications like MicroTE, Fast Failover, Mobility, etc. These applications assume that the latency in interacting with the network switches is constant and negligible. However, our measurement studies show that this latency is significant. Moreover, it varies with the switch platforms, type of operations performed, table occupancy and concurrent operations on the switches.

Using grey-box probing, we narrow down the key factors for these latencies to be TCAM Organization, Low power switch CPU and software implementation inefficiencies. To overcome the latencies and achieve responsive control, we develop a systematic framework leveraging both the logically central view and global control in SDN, and the dissection of latencies from our measurement study.

---

**ELEMENTS OF LATENCY**

- **Inbound Latency**
  - Higher than insertion latency
  - Not affected by rule priority but affected by table occupancy

- **Outbound Latency - Insertion**
  - Affected by priority insertion patterns
  - Affected by the table occupancy

- **Outbound Latency - Modify/Delete**
  - Affected by priority insertion patterns
  - Affected by the table occupancy

---

**TAMING LATENCY**

- **Controllers**
  - Flow engineering
  - Rule offloading
  - Flow level optimization
  - Rule level optimization

- **Data Plane Hardware**
  - Switch agent
  - Unmatched packet

- **Four Modules**
  1. Proxy
  2. Flow Engineering
  3. Rule Offloading
  4. Optimal Rule Update

---

**INBOUND LATENCY**

- Increases with flow arrival rate
- Increases with interference from outbound msgs

<table>
<thead>
<tr>
<th>Flow Arrival Rate (packets/sec)</th>
<th>Mean Delay per packet (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>3.32</td>
</tr>
<tr>
<td>200</td>
<td>8.13</td>
</tr>
</tbody>
</table>

---

**OUTBOUND LATENCY**

- **Outbound Latency**
  - Inbound: Send to ASIC SDK
  - Outbound: Send to OF Agent
  - Send to Controller

---

**PERFORMANCE**

- **Inbound**
  - Prototyped on a commodity host(101 quad core, 2.66GHz, 8GB RAM)
  - Proxy almost completely eliminates the inbound delay

<table>
<thead>
<tr>
<th>Flow Arrival Rate (packets/sec)</th>
<th>Delay w/o Proxy (msec)</th>
<th>Delay with Proxy (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>8</td>
<td>0.01</td>
</tr>
<tr>
<td>2000</td>
<td>-</td>
<td>0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow Arrival Rate (packets/sec)</th>
<th>99th percentile delay w/o Proxy (msec)</th>
<th>99th percentile delay with Proxy (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>192</td>
<td>0.07</td>
</tr>
<tr>
<td>2000</td>
<td>30</td>
<td>3.5</td>
</tr>
</tbody>
</table>

- **Simulated failover scenario in a tunneled WAN Network**
- **Topology:** Full mesh with 25 nodes
- **Traffic matrix:** Assign a popularity index to each node
- **Table occupancy:** Assume switches have some pre-installed rules
- **Workloads:** 6 workloads which have different table occupancies and traffic volumes