OvS Offload: Too Many Models?

Hemal V. Shah, Distinguished Engineer and Architect, Broadcom Inc.
Agenda

- OvS Offload Terminology
- OvS-DPDK Offload Models
- OvS Kernel Offload Models
- Comparison
- Summary
OvS Offload

- OvS Offload: Offloading of match/action processing from OvS
- Match Offload: Offloading of flow classification (RX only)
- Action Offload: Partial offloading of match and actions (TX/RX)
- Full Offload: Offloading of match and actions (TX/RX), OvS is bypassed

Why Offload? Higher Throughput, Higher Efficiency, Lower Latency
VMs connect to OvS-DPDK using virtio
• OvS-DPDK provides VM-VM and VM-Net connectivity
• OvS-DPDK flow processing offloads are executed by NIC
• OvS-DPDK is always in the datapath
• Control Plane: rte_flow
• PF or trusted VF PMD per physical uplink port

• Match Offload (RX only) well integrated into OvS-DPDK
• Arch Challenges and Complexities with actions offload
• Benefits of actions offload are not characterized
OvS-DPDK Full Offload with SR-IOV

- VMs bypass OvS-DPDK
- NIC applies OvS policies
- VF-Reps represent VFs in OvS-DPDK
- Control plane: rte_flow & VF-Rep
- PF or trusted VF PMD per phy uplink port

- Control plane infrastructure is in place
- Full benefits of offload and OvS bypass
- Requires vendor specific VF driver in VM
- Doesn’t support live VM migration
- Two bridge design is challenging for Tunnel Decap offload
OvS-DPDK Full vhost Offload with SR-IOV

- VMs use virtio
- NIC applies OvS policies
- VF-Reps represent VFs in OvS-DPDK
- Control plane: rte_flow & VF-Rep
- PF or trusted VF PMD per phy uplink port

- Doesn’t require vendor specific driver in VM
- Live VM migration is supported
- Control plane infrastructure is in place
- SW forwarder overhead reduces offload benefits
- Two bridge design challenging for Tunnel Decap offload
VMs can bypass OvS
NIC applies OvS policies
VF-Reps represent VFs in OvS
Control plane: TC-Flower & VF-Rep
Management plane: switchdev

Control & Management planes are in place
Full benefits of offload and OvS bypass
Requires vendor specific VF driver in VM
 Doesn’t support live VM migration
Does not Support User Mode Appliances/Switch
# Comparison of OvS Offload Models

<table>
<thead>
<tr>
<th>Model</th>
<th>OvS Bypass</th>
<th>Supports VM Migration</th>
<th>VM Driver</th>
<th>Control Plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>OvS-DPDK Partial Action Offload</td>
<td>No</td>
<td>Yes</td>
<td>virtio</td>
<td>rte_flow</td>
</tr>
<tr>
<td>OvS-DPDK Full Offload w/ SR-IOV</td>
<td>Yes</td>
<td>No</td>
<td>Vendor Specific</td>
<td>rte_flow VF-Rep</td>
</tr>
<tr>
<td>OvS-DPDK VHOST Full Offload (w/ virtio &amp; SR-IOV)</td>
<td>Yes (FWD bridge)</td>
<td>Yes</td>
<td>virtio</td>
<td>rte_flow VF-Rep</td>
</tr>
<tr>
<td>OvS-TC-flower Full Offload w/ SR-IOV</td>
<td>Yes</td>
<td>No</td>
<td>Vendor Specific</td>
<td>TC Flower VF-Rep</td>
</tr>
</tbody>
</table>
Summary

- **OvS Data Path Offload:** Models exist for user and kernel modes
- **OvS Offload Layer (Control Plane):** Remains challenging
- **Action Offload:** Inherently Challenging with OvS architecture & Offload Layer
- **Full Offload:** SR-IOV infrastructure exists, Issues - VM migration & offload design

**Over choice of OvS Offload Models and Complexity of Offload Layer are Challenging for the Deployment and Adoption of OvS Offload**