Agenda

- Dynamic Rebalancing
- Design and Implementation for OVS TC-Flower (Kernel) Offload
- Proposed changes for OVS DPDK (User mode) Offload
**Dynamic Rebalancing**

- **Problem Description**
  - HW flow offload resources: Limited by NIC capacity
  - Once offloaded, a flow stays offloaded until deleted by OVS
  - New offload requests may fail due to offload capacity
  - Leads to inefficient utilization of HW and CPU resources

Can the flows be offloaded and managed more intelligently?
Dynamic Rebalancing – TC-Flower Offload

• **Design Points**
  – Dynamic discovery of capacity exhaustion
  – Offload selection
    – Based on flow PPS rate
  – Rebalancing of offloaded flows
    – Higher PPS flows replace lower PPS flows
  – User config – “offload-rebalance” option

• **Layered Design**
  – OOR error reporting (driver)
  – OOR detection (dpif-provider - handler)
  – Offload selection (ofproto-dpif - revalidator)
  – Rebalancing (ofproto-dpif - revalidator)
• Leverage rebalancing design from OvS-TC flower (kernel)
  — Rebalancing in the offload layer (transparent to users and applications)
  — PPS estimation based on flow packet counts over a sample period

• OvS-DPDK Gaps/Challenges for rebalancing
  — Resource exhaustion reporting in PMD
  — Asynchronous processing of candidate flows by offload thread
  — Offloaded flows maintained in user data path and HW (different from OvS-TC flower)

• Proposal
  — Add OOR error reporting in PMD → ENOSPC from rte_flow_create()
  — OOR detection in OvS-DPDK offload thread
  — Rebalancing logic in revalidators (reuse)
  — Add a mechanism to sync between offload thread & revalidators during rebalancing
  — Rebalanced flows need to be inserted/removed from HW only