DPDK-accelerated Partial Offload for Fine-grained HQoS

ROSEN XU, QINGSONG WANG
INTEL
Agenda

• A generic HQoS Use-case Overview
• Existing DPDK HQoS Capabilities
• Hardware Limitation for Fine-grained HQoS
• FPGA based Fine-grained HQoS reference design in Intel PAC N3000
• Fine-grained Partial Offload HQoS
A generic HQoS Use-case Overview

- Which flow a packet belongs to
- When a packet becomes eligible for scheduling
- What order to schedule packet flows
Existing DPDK HQoS Capabilities

```c
struct rte_tm_ops {
    /** Traffic manager node type get */
    rte_tm_node_type_get_t node_type_get;
    /** Traffic manager capabilities_get */
    rte_tm_capabilities_get_t capabilities_get;
    /** Traffic manager level capabilities_get */
    rte_tm_level_capabilities_get_t level_capabilities_get;
    /** Traffic manager node capabilities get */
    rte_tm_node_capabilities_get_t node_capabilities_get;
    /* Traffic manager WRED profile add */
    rte_tm_wred_profile_add_t wred_profile_add;
    /* Traffic manager WRED profile delete */
    rte_tm_wred_profile_delete_t wred_profile_delete;
    /* Traffic manager shared WRED context add/update */
    rte_tm_shared_wred_context_add_update_t
        shared_wred_context_add_update;
    /* Traffic manager shared WRED context delete */
    rte_tm_shared_wred_context_delete_t
        shared_wred_context_delete;
    /* Traffic manager shaper profile add */
    rte_tm_shaper_profile_add_t shaper_profile_add;
    /* Traffic manager shaper profile delete */
    rte_tm_shaper_profile_delete_t shaper_profile_delete;
    /* Traffic manager shared shaper add/update */
    rte_tm_shared_shaper_add_update_t
        shared_shaper_add_update;
    /* Traffic manager shared shaper delete */
    rte_tm_shared_shaper_delete_t
        shared_shaper_delete;
    ...
};
```

Meter

Congestion management

Shaping
HW Limitation for Fine-grained HQoS

- Multi-tenant cloud network requires tens of thousands of flows
- Limited and none-scalable Hardware Queue Number
- Lack of flexibility for packet scheduler than software
- Limitation of enough Board Memory to keep packets
Fine-grained HQoS reference design – Intel PAC N3000

► Intel® Arria® 10 GT1150
  ▪ 8x10GbE, 4x25GbE Network Interfaces
  ▪ Local DDR4 and QDR Memory

► Dual Intel® Ethernet Controller XL710
  ▪ Extensive OS support and Easier system integration
Fine-grained HQoS reference design - WRED

- Every traffic class has 7 states, different threshold for Inqueue and Dequeue
- WRED calculates average queue length when packets Inqueue and Dequeue
- Packet drop or not depends on its color and current queue status
Fine-grained HQoS reference design - Shaper & Sche

- Queuing
  - Queue ID from Metadata
  - Traffic enqueued into 64K queues in Board Memory

- Scheduling
  - 3-layer scheduler
  - SP+DWRR

- Shaping
  - Input queue shaper per scheduler layer and port shaper

- Statistics
  - Port packet + byte counters
Intel PAC N3000 HQoS API

- 2018'Q3 DPDK with FPGA TM Offloading POC, enable 64k flows
- 2019'Q1 DPDK with PAC N3000 FPGA TM Offloading upstream 64K flows
Summary of FPGA based HQoS

- More configurable and flexible than NIC
- Provide Fine-grained HQoS to Multi-tenant cloud network
- Need lots of Board Memory to keep all packets
- Can software take up some parts of HQoS?
Fine-grained Partial Offload HQoS

- [RX Thread] Packet receive and parse
- [Classifier Thread] Packet classifier, divides input packets into different flow, load balance and dispatch flow to different queue
- [Enqueue Thread] Meter, mark color and calculate WRED for different queue and enqueue
- [TX Thread] Generate queue info for FPGA Shaping, packets are sent from NIC DMA