DPDK Quality of Service (QoS) APIs

Cristian Dumitrescu, Intel
Jasvinder Singh, Intel

DPDK Summit Userspace - Dublin - 2017
Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel, the Intel logo and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

© 2017 Intel Corporation.
Traffic Management API (rte_tm.h)
Traffic Metering and Policing API (rte_mtr.h)
Soft NIC: SW fall-back for ethdev APIs (drivers/net/softnic)
- Generic API for yet another standard feature of Ethernet devices (TX path)
- Implementation agnostic: HW-SW mix, device type (NIC, CPU, NPU, FPGA, ASIC), vendor
- Part of DPDK since release 17.08

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability</td>
<td>Per port, level, node</td>
</tr>
<tr>
<td>Scheduling</td>
<td>Strict Priority (SP), Weighted Fair Queueing (WFQ)</td>
</tr>
<tr>
<td>Traffic Shaping</td>
<td>Single/dual rate shapers&lt;br&gt;Private (per node) and shared (by many nodes) shapers</td>
</tr>
<tr>
<td>Congestion Mgmt</td>
<td>Tail Drop, Head Drop, WRED</td>
</tr>
<tr>
<td>Packet Marking</td>
<td>VLAN DEI (802.1Q), IPv4/IPv6 ECN (RFC 3168), IPv4/IPv6 DSCP (RFC 2597)</td>
</tr>
</tbody>
</table>
Steps to build the hierarchy

- Ethernet device (port)
- Hierarchy nodes
  - Leaf nodes: Predefined IDs
  - Non-leaf nodes: Application provided IDs
- Shapers: private, shared
- Shaper profiles
- WRED contexts: private, shared
- WRED profiles
Traffic Metering & Policing (rte_mtr.h)

- Generic API for yet another standard feature of Ethernet devices (RX path)
- Implementation agnostic: HW-SW mix, device type (NIC, CPU, NPU, FPGA, ASIC), vendor
- WIP (V2 pending)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability</td>
<td>Per port, per meter</td>
</tr>
<tr>
<td>Metering</td>
<td>srTCM (RFC 2697), trTCM (RFC 2698, RFC 4115), bypass</td>
</tr>
<tr>
<td>Policing</td>
<td>Actions: No-op, Recolor, Drop</td>
</tr>
<tr>
<td>Statistics</td>
<td>Per output color, configurable</td>
</tr>
<tr>
<td>Flow action</td>
<td>Meter action</td>
</tr>
</tbody>
</table>
NIC operation executed by the CPU

SW fall-back for specific ethdev APIs
  - Example: flow & policing (RX), traffic mgmt (TX)

Generic: work with any "hard" NIC

Configurable: user selects features to be enabled

NFV vision: App (almost) agnostic about the NIC under ethdev API

Status: WIP (V5 pending)
Questions?

Cristian Dumitrescu
cristian.dumitrescu@intel.com

Jasvinder Singh
jasvinder.singh@intel.com