DPDK

DPDK Quality of Service (QoS) APIs

Cristian Dumitrescu, Intel

Jasvinder Singh, Intel

DPDK Summit Userspace - Dublin- 2017

Legal Notices and Disclaimers

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.

Agenda

- 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)

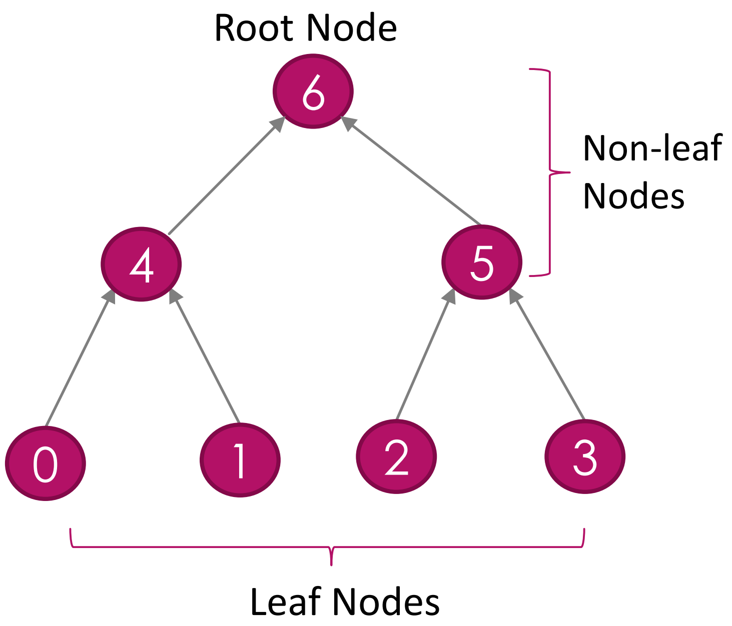
Traffic Management (rte_tm.h)

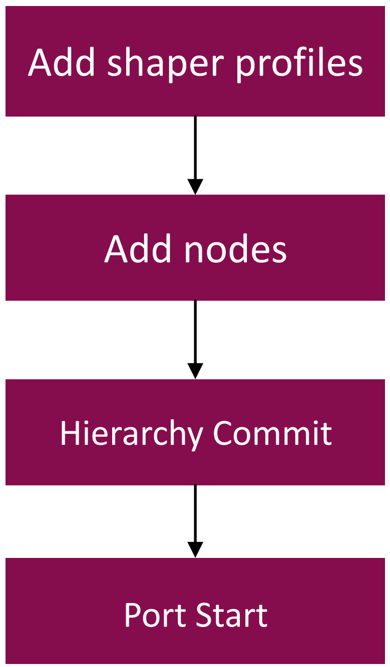
- Generic API for yet another standard feature of Ethernet devices (TX path)
- Implementation agnostic: HW-SW mix, device type (NIC, CPU, NPU, FPGA, ASIC), vendor
- Part of DPDK since release 17.08

Feature	Description
Capability	Per port, level, node
Scheduling	Strict Priority (SP), Weighted Fair Queueing (WFQ)
Traffic Shaping	Single/dual rate shapers Private (per node) and shared (by many nodes) shapers
Congestion Mgmt	Tail Drop, Head Drop, WRED
Packet Marking	VLAN DEI (802.1Q), IPv4/IPv6 ECN (RFC 3168), IPv4/IPv6 DSCP (RFC 2597)

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





Add shaper profiles

Add nodes

Hierarchy Commit

Port Start

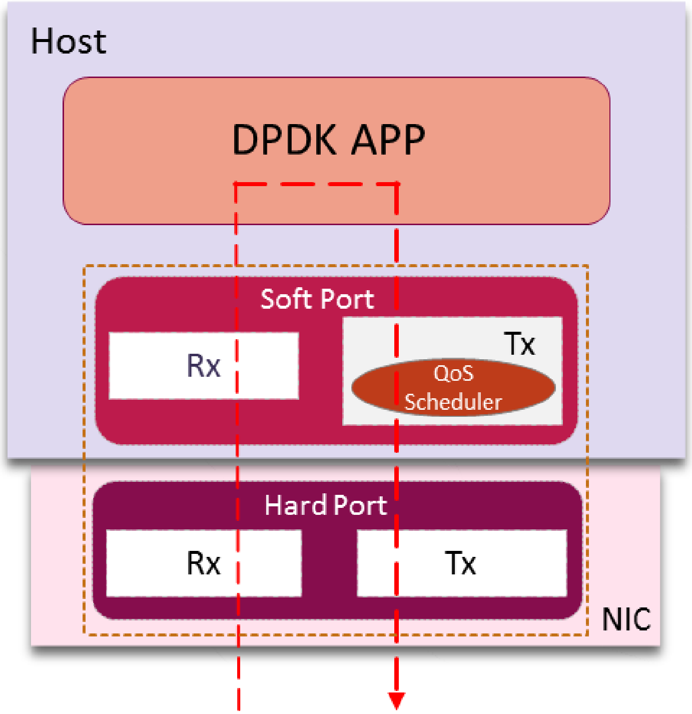
Traffic Metering & Policing (rte_mtr.h)

- 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)

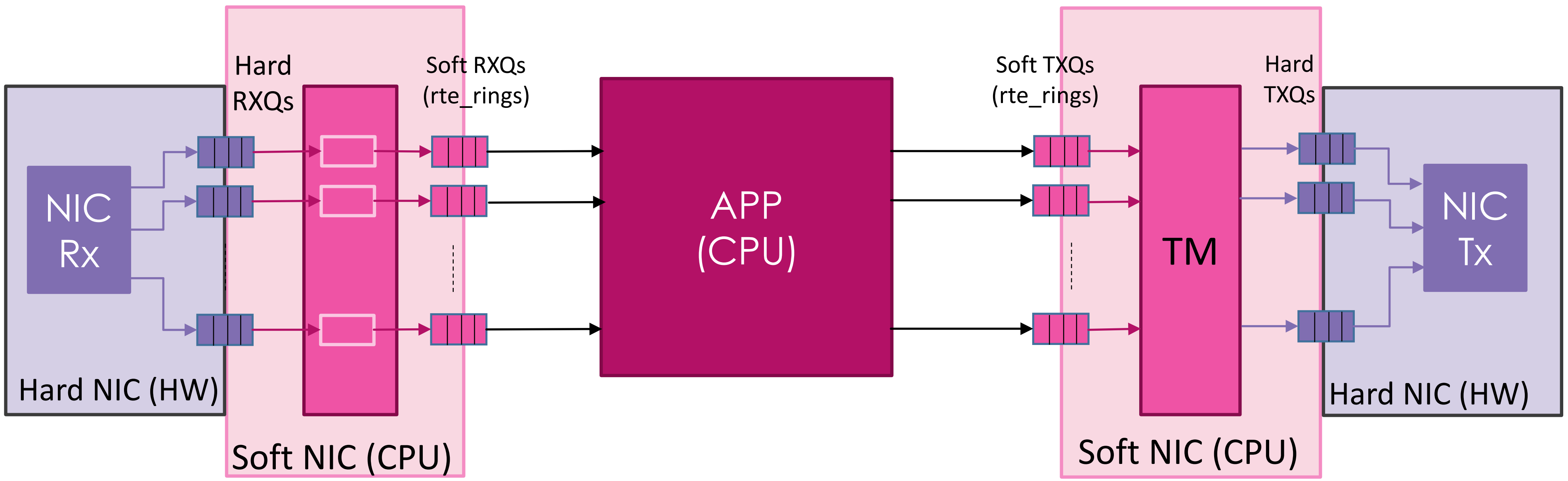
Feature	Description
Capability	Per port, per meter
Metering	srTCM (RFC 2697), trTCM (RFC 2698, RFC 4115), bypass
Policing	Actions: No-op, Recolor, Drop
Statistics	Per output color, configurable
Flow action	Meter action

SoftNIC

- 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)



SoftNIC



Hard RXQs
Soft RXQs (rte_rings)
NIC Rx
Hard NIC (HW)
Soft NIC (CPU)
APP (CPU)
Soft TXQs (rte_rings)
Hard TXQs
TM
NIC Tx
Soft NIC (CPU)
Hard NIC (HW)

Questions?

Cristian Dumitrescu

cristian.dumitrescu@intel.com

Jasvinder Singh

jasvinder.singh@intel.com