NETRONOME

WHITE PAPER

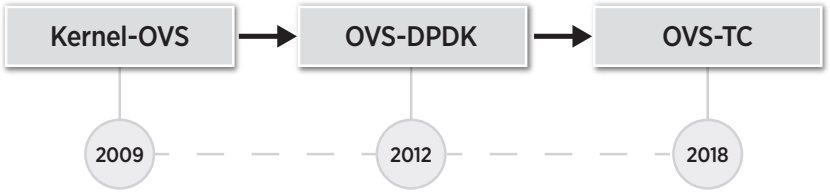
Virtual Switch Acceleration with OVS-TC and Agilio 40GbE SmartNICs

HARDWARE-ACCELERATED OVS-TC PROVIDES BETTER CPU EFFICIENCY, LOWER COMPLEXITY, ENHANCED SCALABILITY AND INCREASED NETWORK PERFORMANCE COMPARED TO KERNEL-OVS AND OVS-DPDK.

CONTENTS

EXECUTIVE SUMMARY

Modern virtualization technologies allow the development and deployment of advanced network services. Virtualized infrastructures take advantage of standardized commercial off-the-shelf (COTS) servers within the data center. This new business model offers versatility, cost savings, easier integrations, more attractive maintenance and management profiles, and an overall lower total cost of ownership (TCO). Virtual switching is an integral part of highly-virtualized environments. Software-defined networking (SDN), also known as server-based networking, has revolutionized the data center, enabling service providers to introduce new services with greater speed and agility. Open vSwitch (OVS) is a widely-deployed example of an SDN-controlled virtual switch for server-based networking. The benefits of OVS for server-based networking deployments have been well established: software-defined flexibility and control of datapath functions, fast feature rollouts, and the benefits of open source ecosystems.



The Open vSwitch kernel module (Kernel-OVS) is the most commonly used OVS datapath. Kernel-OVS is implemented as a match/action forwarding engine based on flows that are inserted, modified or removed by user space. In 2012 OVS was further enhanced with another user space datapath based on the data plane development kit (DPDK). The addition of OVS-DPDK improved performance but created some challenges. OVS-DPDK bypasses the Linux kernel networking stack, requires third party modules and defines its own security model for user space access to networking hardware. DPDK applications are more difficult to configure optimally and while OVS-DPDK management solutions do exist, debugging can become a challenge without access to the tools generally available for the Linux kernel networking stack. It has become clear that a better solution is needed.

OVS using traffic control (TC) is the newest kernel-based approach and improves upon Kernel-OVS and OVS-DPDK by providing a standard upstream interface for hardware acceleration. This paper will discuss how an offloaded OVS-TC solution performs against software-based OVS-DPDK.

OVS SOFTWARE-BASED SOLUTIONS: THE CPU BOTTLENECK

Kernel OVS

Following its initial release as an open source project in 2009, OVS has become the most ubiquitous virtual switch (vSwitch) in Linux deployments. The standard OVS architecture consists of user space and kernel space components. The switch daemon runs in user space and controls the switch while the kernel module implements the OVS packet datapath.

In a software-based solution, the kernel is not ideal for virtual switching because it grants short time quanta to the processes and treats them like any other process in the system. As a result, it imposes excessive contention, the latency of which is greater than the actual runtime of a service. On top of that, virtual switching requires frequent, per-packet, system calls that cause the vSwitch to yield the CPU to the operating system (OS) so that the OS can perform the necessary I/O operations. This leads to degraded network performance.

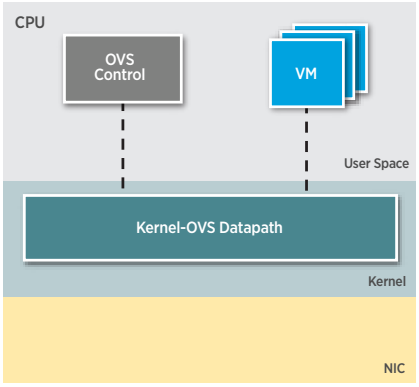


Figure 1: Kernel-OVS datapath

OVS-DPDK

OVS-DPDK is a noteworthy attempt to address the fundamental limitations of Kernel-OVS. By implementing the OVS datapath in user space, the DPDK poll mode driver delivers packets directly into the dedicated user space application, bypassing the Linux kernel stack altogether.

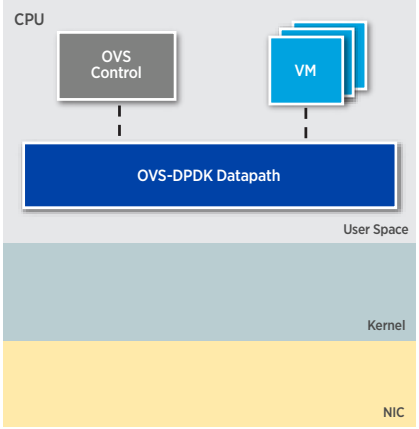


Figure 2: OVS-DPDK datapath

This eliminates unnecessary overhead in the stack and can enable additional optimizations for the vSwitch, such as loading packets directly into caches and batch processing. The DPDK community regularly provides further optimizations and tuning for OVS with upstreamed DPDK setup for network interface cards (NICs).

OVS-DPDK enables software acceleration and in a few cases the user can tune it to be an adequate "work around" for the major Kernel-OVS performance bottlenecks. DPDK generally uses dedicated logical cores to gain sufficient networking performance which places a limit on the scalability of DPDK as a software-accelerated virtual switching solution. Moreover, OVS-DPDK is not part of the Linux kernel and this renders the solution more cumbersome with higher operational overhead.

It is important to note that for OVS-DPDK to run closer to line-rate performance it has to consume more CPU cores. This is the true hidden cost of deploying OVS-DPDK. As shown in the benchmarks, even when sacrificing precious CPU cores, this solution is not capable of performing in a scaled-up data center environment.

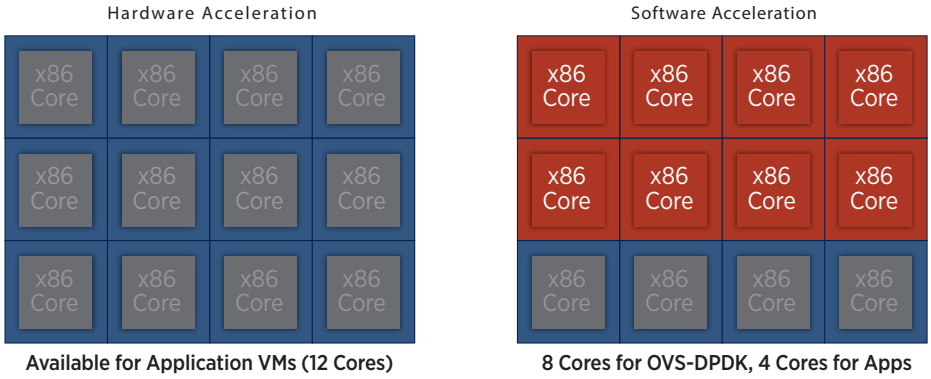


Figure 3: True performance cost of software acceleration on a single CPU socket

OVS-TC

TC Flower is a packet classifier in the Linux kernel and part of the kernel traffic classification subsystem. OVS-TC is the extension of OVS user space code to enable it to offload flows using TC Flower and TC actions. OVS-TC allows matching on a variety of predefined flow keys. The user can match on IP addresses, UDP/TCP ports, metadata and more. Like OVS, OVS-TC includes an action side which allows packets to be modified, forwarded or dropped. The user can influence what is offloaded or not down to a per flow basis.

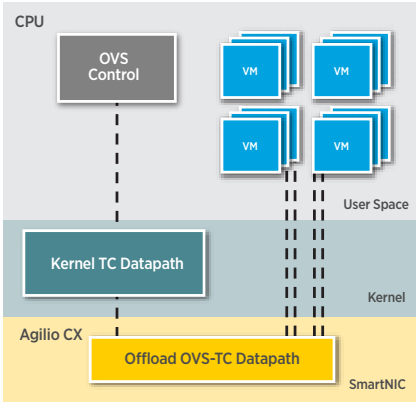


Figure 4: OVS-TC datapath

NETRONOME AGILIO CX SMARTNICS PROVIDE A FRAMEWORK FOR TRANSPARENT OFFLOAD OF THE TC DATAPATH.

The TC command line provides a common set of tools for configuring queuing disciplines, classifiers and actions. The TC Flower classifier, combined with actions, may be used to provide match/action behavior similar to Kernel-OVS and OVS-DPDK. OVS leverages the TC datapath to gain hardware acceleration.

Service providers need a scalable vSwitch, and now there is an open source, upstreamed and kernel-compliant solution with OVS-TC which maintains all the benefits of Kernel-OVS and OVS-DPDK. In addition, hardware-accelerated OVS-TC provides better CPU efficiency, lower complexity, enhanced scalability and increased network performance.

	Kernel-OVS	OVS-DPDK	OVS-TC Offload
CPU Efficiency	✗	✗	✔
Low Optimization Complexity	✔	✗	✔
Scalability	✗	✔	✔
High Performance	✗	✗	✔

UPSTREAMED OVS-TC OFFLOAD: THE SMARTNIC REVOLUTION

NETRONOME AGILIO CX SMARTNICS HAVE NATIVE, IN-BOX SUPPORT IN RED HAT ENTERPRISE LINUX 7.5 AND UBUNTU 18.04.

Netronome Agilio CX SmartNICs enable transparent offload of the TC datapath. While OVS software still runs on the server, the OVS-TC datapath match/action modules are synchronized down to the Agilio SmartNIC via hooks provided in the Linux kernel. The 60 cores (480 threads) on the Agilio CX SmartNIC provide industry-leading hardware acceleration, consume less than 25W and deliver groundbreaking ROI.

OVS-TC hardware acceleration on the Agilio SmartNIC is supported with a wide range of features. As the OVS community adds more features to OVS-TC, Netronome enables acceleration of those features with firmware upgrades. In the appendix of this whitepaper, there is a comprehensive list of the match/action supported features for OVS-TC.

Open source and upstreamed solutions like OVS-TC represent the future of our industry because they are easier to implement and non-proprietary. The traffic classification subsystem contained on TC makes it possible to use other kinds of classifiers to implement the matching of packets. An example of this would be BPF, which uses a bpfilter to match packets.

AGILIO TRANSPARENT OVS OFFLOAD ARCHITECTURE

By running the OVS-TC data functions on the Agilio SmartNIC the TC datapath is dramatically accelerated while leaving higher-level functionality and features under software control. Hardware acceleration achieves significant performance improvements while retaining the leverage and benefits derived from a sizeable open source development community.

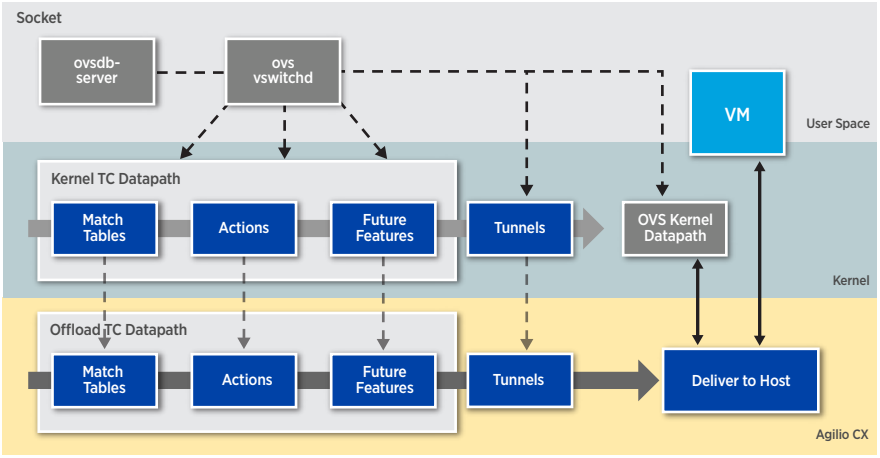


Figure 5: The Agilio transparent OVS-TC offload architecture

When using OVS-TC there are three datapaths present:

1. SmartNIC datapath (as per diagram)
2. TC kernel datapath (typically not populated)
3. OVS kernel datapath

OVS contains a user space-based agent and a kernel-based datapath. The user space agent is responsible for switch configuration and flow table population. As observed in Figure 5, it is broken up into two fundamental components: ovs-vswitchd and ovsdb-server. The user space agent can accept configuration via a local command line (e.g. ovs-vsctl) and from a remote controller. When using a remote controller it is common to use ovsdb to interact with ovsdb-server for vSwitch configuration. The kernel TC datapath is where the Agilio offload hooks are inserted. With this solution, the OVS software still runs on the server, but the OVS-TC datapath match/action modules are synchronized down to the Agilio SmartNIC via hooks provided in the Linux kernel.

BENCHMARK METHODOLOGY AND TEST SETUP (PHY-OVS-PHY)

The Netronome performance team has put together a set of tests that provide performance data comparing Netronome Agilio CX 2x40GbE SmartNICs offloading OVS-TC from the server CPU, to DPDK-based user space OVS (OVS-DPDK). For the latter approach, traditional 2x40GbE NICs are used. A PHY-OVS-PHY topology was tested on both OVS-DPDK and OVS-TC. The key goal of this benchmarking was to quantify the performance scalability of the OVS-TC offload datapath in comparison to OVS-DPDK.

The following hardware and software were used for the test:

Server:
Dell PowerEdge R730

CPU:
2x10 Core Intel Xeon CPU E5-2650 v4

Software:
Kernel Version: 4.15.0-041500-generic
Open vSwitch: v2.8.1
DPDK Release 17.05.2

NICs:
a) Netronome Agilio CX 2x40GbE SmartNIC
b) Intel XL710 2x40GbE Fortville NIC

Test Generator:
Ixia running RFC2544, 4X TX/RX (phy,vhost)

The benchmarking setups used for OVS-DPDK on the XL710 and accelerated OVS-TC on the Agilio CX SmartNIC are shown in the diagrams below:

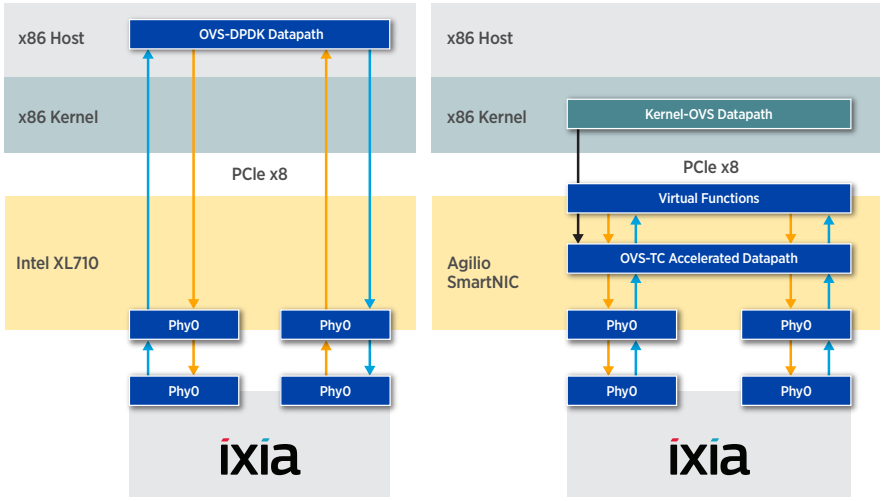


Figure 6: OVS-DPDK and OVS-TC benchmark setups connected to Ixia

Ixia was used as an external hardware traffic generator (IxOS/IxNetwork 8.01) which can transmit and receive traffic.

Traffic Profiles

The following traffic profiles were used:

- Frame sizes (bytes): 64, 128, 256, 512, 768, 1024, 1280, 1518
- 1-Port FWD, ethipv4_1p-fwd | PHY-OVS-PHY
- Flows: 1,000, 8,000, 16,000, 32,000 64,000, 128,000, 256,000
- Rules: 1,000, 8,000, 16,000, 32,000 64,000, 128,000, 256,000

Performance Tuning:

- IRQ pinning of management data
- BIOS tuning
- Host boot kernel tuning

BENCHMARK RESULTS

For each of the test cases, the applied load is 40Gb/s for packet sizes ranging from 64Byte to 1518Byte with 1,000 up to 256,000 flows and 1,000 up to 256,000 rules (1:1). Packets are injected from the traffic generator at the network interface and into the datapath for both cases. The following graphs display the datapath performance for OVS-DPDK and hardware accelerated OVS-TC in a scale-up test using a single port.

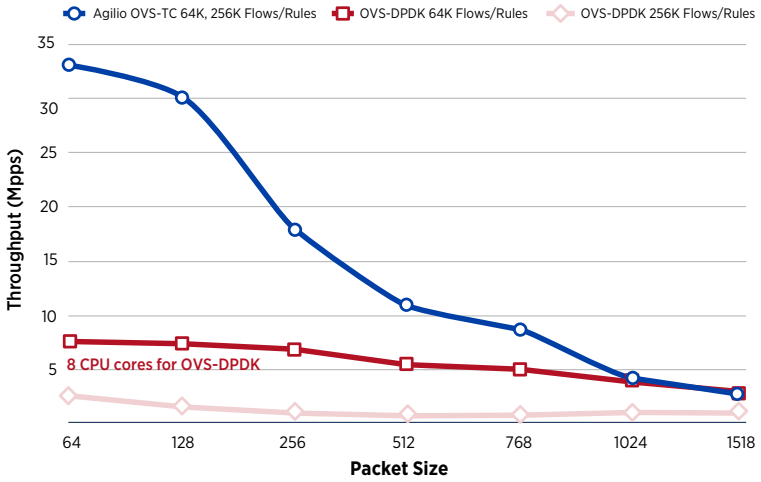


Figure 7: OVS-DPDK performance drops as we scale to higher flow count

Compared to OVS-DPDK, OVS-TC has a very different performance behavior with increased flows and rules. At 64B packet size (*Figure 7*), OVS-TC delivers 33 Mpps for 64K flows, and as we scale to 256K flows or more, the performance does not drop. When compared to OVS-DPDK performance for the same test, OVS-TC running on the Agilio SmartNIC performs 3X to 8X better than the software-accelerated solution.

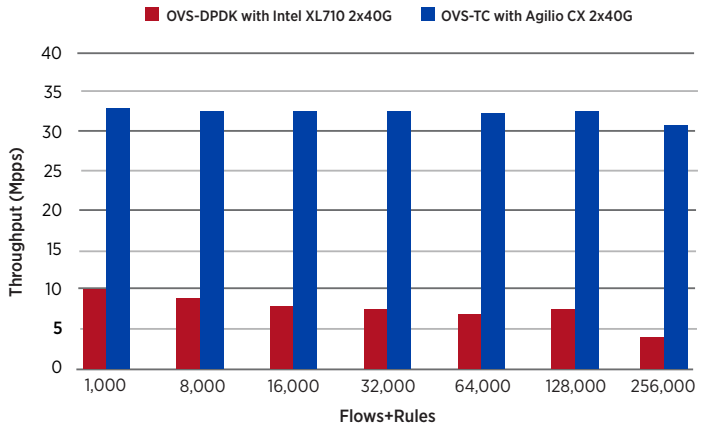


Figure 8: Flow scalability 64B packet size test results for 1:1 rules/flows

The results of the standard OVS-DPDK test highlight that increased number of flows has a negative impact on the throughput. As the flows increase, the frame rate drops. At 64B packet size, OVS-DPDK delivers 7 Mpps on 64K flows/rules but when scaled to 256K flows/rules the performance drops by 1.75X to approximately 4 Mpps (*Figure 8*). As the amount of flows and rules increase, the OVS-DPDK performance degrades dramatically. At a 64B packet size, OVS-TC delivers 19Gb/s for 64K flows+rules (*Figure 9*) . When compared to OVS-DPDK performance for this test, OVS-TC running on the Agilio SmartNIC performs 5X better than the software-accelerated solution.

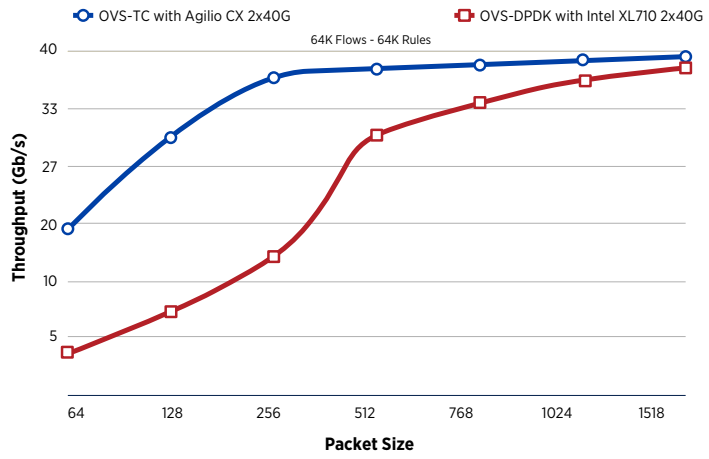


Figure 9: Agilio SmartNIC with OVS-TC thoughtput versus Intel XL710 with OVS-DPDK

For real-world data center conditions and larger number of flows and rules, OVS-TC with Agilio CX SmartNIC delivers 2.5X lower latency than OVS-DPDK with Intel NICs.

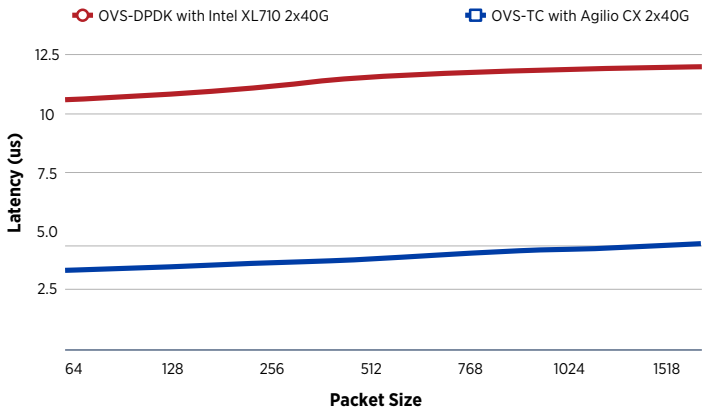


Figure 10: 2.5X lower latency with Agilio CX

CONCLUSION

Agilio CX SmartNICs with OVS-TC significantly outperform OVS-DPDK with traditional NICs. The packet-per-second throughput of hardware-accelerated OVS-TC is 3X to 8X higher than OVS-DPDK with eight CPU cores allocated. For flow/rule counts above 256,000 this delta becomes even larger, approaching 10X greater performance. Normalized per CPU core, Agilio OVS-TC performs more than 20X better than OVS-DPDK.

OVS-TC leads the way in virtual switching at scale by accelerating the Linux networking stack. The results shown in this paper confirm that Agilio SmartNICs with OVS-TC can deliver more data to more applications than OVS-DPDK. This translates directly to improved server efficiency and a dramatic reduction in TCO, as fewer servers and less data center infrastructure (such as switches, racks, and cabling) are needed to perform a given application workload.

APPENDIX A

OVS-TC Supported Hardware-Accelerated Match/Action Features on Agilio CX SmartNICs

Kernel Datapath Match: **Accelerated Match	OVS-DPDK	OVS-TC
OVS_KEY_ATTR_PRIORITY	✗	✗
OVS_KEY_ATTR_IN_PORT	✔	✔
OVS_KEY_ATTR_ETHERNET	✔	✔
OVS_KEY_ATTR_VLAN	✔	✔
OVS_KEY_ATTR_ETHERTYPE	✔	✔
OVS_KEY_ATTR_IPV4	✔	✔
OVS_KEY_ATTR_IPV6	✔	✔
OVS_KEY_ATTR_TCP	✔	✔
OVS_KEY_ATTR_UDP	✔	✔
OVS_KEY_ATTR_ICMP	✔	✔
OVS_KEY_ATTR_ICMPV6	✔	✔
OVS_KEY_ATTR_ARP	✗	✗
OVS_KEY_ATTR_ND	✗	✗
OVS_KEY_ATTR_SKB_MARK	✗	✗
OVS_KEY_ATTR_TUNNEL	✔	✔
OVS_KEY_ATTR_SCTP	✔	✔
OVS_KEY_ATTR_TCP_FLAGS	✗	✗
OVS_KEY_ATTR_DP_HASH	✗	✗
OVS_KEY_ATTR_RECIRC_ID	✗	✗
OVS_KEY_ATTR_MPLS	✔	✔
OVS_KEY_ATTR_CT_STATE	✔	✗
OVS_KEY_ATTR_CT_ZONE	✔	✗
OVS_KEY_ATTR_CT_MARK	✔	✗
OVS_KEY_ATTR_CT_LABELS	✔	✗
Additional Actions (Outside Vanilla OVS)		
OVS_KEY_ATTR_NSH	✗	✗

Kernel Datapath Actions: **Accelerated Actions	OVS-DPDK	OVS-TC
OVS_ACTION_ATTR_OUTPUT	✔	✔
OVS_ACTION_ATTR_SET_PRIORITY	✔	✗
OVS_ACTION_ATTR_SET_SKB_MARK	✗	✗
OVS_ACTION_ATTR_SET_TUNNEL_ INFO	✔	✔
OVS_ACTION_ATTR_SET_ETHERNET	✔	✔
OVS_ACTION_ATTR_SET_IPV4	✔	✔
OVS_ACTION_ATTR_SET_IPV6	✔	✔
OVS_ACTION_ATTR_SET_TCP	✔	✔
OVS_ACTION_ATTR_SET_UDP	✔	✔
OVS_ACTION_ATTR_SET_MPLS	✔	✗
OVS_ACTION_ATTR_SET_CT_STATE	✔	✗
OVS_ACTION_ATTR_SET_CT_ZONE	✔	✗
OVS_ACTION_ATTR_SET_CT_MARK	✔	✗
OVS_ACTION_ATTR_SET_CT_LABELS	✔	✗
OVS_ACTION_ATTR_PUSH_VLAN	✔	✔
OVS_ACTION_ATTR_POP_VLAN	✔	✔
OVS_ACTION_ATTR_RECIRC	✗	✗
OVS_ACTION_ATTR_HASH	✗	✗
OVS_ACTION_ATTR_PUSH_MPLS	✔	✗
OVS_ACTION_ATTR_POP_MPLS	✔	✗
OVS_ACTION_ATTR_SET_MASKED	✔	✔
OVS_ACTION_ATTR_SAMPLE	✗	✗
OVS_ACTION_ATTR_CT	✔	✗
OVS_ACTION_ATTR_DROP	✔	✔

Features	OVS-DPDK	OVS-TC
Breakout cable support	✔	✔
Balance SLB	✔	✔
VXLAN	✔	✔
VLAN	✔	✔
VXLAN+VLAN+LAG	✔	✗

NETRONOME

Netronome Systems, Inc.
2903 Bunker Hill Lane, Suite 150 Santa Clara, CA 95054
Tel: 408.496.0022 | Fax: 408.586.0002
www.netronome.com