Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: January 5, 2015

Z. Cao
Q. Fu
L. Deng
China Mobile

July 4, 2014

Data Plane Processing Acceleration Framework
draft-cao-dataplane-acceleration-framework-01

Abstract

It is getting popular to running data applications over general purpose hardware/chipsets, instead of customized and dedicated hardware/chipset. This way further decouples the software functions from the hardware. But moving data processing intensive applications to general purpose hardware is still challenging, although the industry has supplied some proprietary solutions. This document discusses the problems of data plane acceleration and proposes its framework.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Table of Contents

1. Introduction

The need of running network data processing functions over general purpose hardware/chipset (e.g., X86, PPC, etc) is multi-folded.

1. Decoupling software functions from hardware. Traditional network devices are built upon dedicated or deep customized hardware and chipsets. This way restricts the flexibility of both service providers and network operators.

2. Network Function Virtualization (NFV). NFV is an initiative of ETSI to virtualize the network functions to the overlay on top of the virtualization layer. It provides network elasticity in that the network functions can be scaled up/down according to the traffic load. NFV solutions often bundle with the virtual switches to provide VM-VM communications. Theses virtual switches are running on top of the servers that bear the network functions. Therefore, the need to accelerate the data processing efficiency is indispensable.

3. Service Time-to-Market . Via the software and hardware decoupling, the speed to provide new services (TTM) is greatly enhanced. Since more and more services would like to have the most convenient time to market, they would also like to move data processing functions on top of general purpose hardware/chipsets.

4. Capex and Opex pressure. Having the network functions running over general purpose device will help operators to cut down their Capex and Opex.

5. Cost-performance targets: software development, debug and integration is simplified; processor resource utilization is improved because the control plane and data plane can be distributed among cores with greater flexibility; development schedule risk is minimized and software maintenance is much easier with a common code base and a single development environment.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. DPA Framework

NF (Network Function): A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behaviour. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms, a Network Function is today often a network node or physical appliance. [Quoted from ETSI NFV]

3.1. Framework

The framework is depicted in Figure 1. Framework.

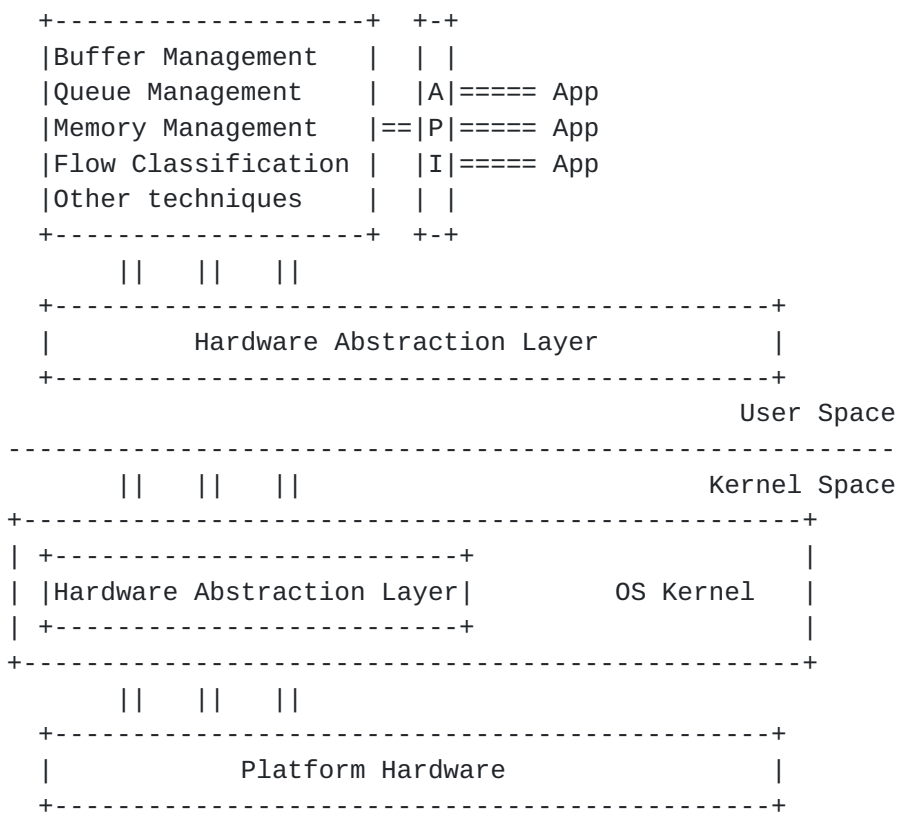

Figure 1

3.2. Components

The DPA may include the following components.

Memory/Buffer Manager. The Memory/Buffer Manager is responsible for allocating NUMA-aware pools of objects in memory and balancing memory bandwidth utilization across the channels. Such management can significantly reduces the amount of time the operating system must spend allocating and de-allocating buffers.

Queue Manager. The Queue Manager is responsible for queue scheduling. The ultimate goal of the Queue Manager is to allow different software components to process packets, while avoiding unnecessary wait times.

Flow Classification. The Flow Classification component is an efficient mechanism for generating a hash used to quickly combine packets into flows, which enables faster processing and greater throughput.

Poll Mode Drivers. The Poll Mode Drivers is capable of speeding up the packet pipeline for 1 GbE and 10 GbE ethernet controllers by

receiving and transmitting packets without the use of asynchronous, interrupt- based signaling mechanisms, which have a lot of overhead.

Environment Abstraction Layer. The Environment Abstraction Layer provides an abstraction to platform-specific initialization code, which eases application porting effort. The EAL provides access to low-level resources (hardware, memory space, logical cores, etc.) through a generic interface that hides the environment specifics from the applications and libraries.

3.3. Protocol Portfolio

On one hand, for the data plane, DPA should provide an efficient stack for common protocols utilized by various internet applications, including but not limited to:

1. Link layer: Layer 2 switch, VLAN.

2. Network layer: IPv4 and IPv6 for packet routing; MPLS and GRE/GTP for tunneled routing; IPsec, TLS/DTLS, NAT and QoS support for security and management features.

3. Transport layer: SCTP/MPTCP as well as TCP and UDP, for multi-homing/stream traffic.

4. Application layer: SSL termination for remote administration of virtualized device.

On the other hand, for the control plane, DPA should provide an efficient stack for common protocols utilized by various network devices/ISPs for improved operation and Management, including: NetFlow, sFlow, IPFIX, SPAN, RSPAN for VM traffic monitory, LACP, STP and openflow for L2/L3 management.

4. Existing Work - Intel DPDK

This section introduces DPDK [DPDK].

Intel Data Plane Development Kit (DPDK) is a set of libraries and drivers for fast packet processing on x86 platforms. It runs mostly in Linux userland.The idea of DPDK has significantly advanced the concept of consolidation of data and control planes on a general purpose processor. Such idea greatly boosts packet processing performance and throughput by providing Intel architecture-optimized libraries to accelerate L3 forwarding, yielding performance that scales linearly with the number of cores, in contrast to native Linux.

The Intel DPDK contains a growing number of libraries, whose source code is available for developers to use and/or modify in a production network element. Likewise, there are various usecase examples, such as L3 forwarding, load balancing, and timers, that help reduce development time. The libraries can be used to build applications based on "run-to completion" or "pipeline" models, enabling the equipment provider's application to maintain complete control.

the Intel DPDK software is also available to aid in the development of I/O intensive applications running in a virtualized environment. This combination allows application developers to achieve near-native performance.

The Intel DPDK provides a simple framework for fast packet processing in data plane application. Developers may use the code to understand some of the techniques employed, to build upon for prototyping, or to add their own protocol stacks. SR-IOV features are also used for hadware-based I/O sharing in I/O virtualization (IOV) mode. Therefore, it is possible to partition intel 82599 10 Gb Ethernet controller NIC resources logically and expose them to a VM as a virtual function

Furthermore, 6WIND has developed a number of value-added enhancements to the Intel DPDK library that provide increased system functionality and performance compared to the baseline software. These value-added enhancements include the following aspects.

Hige-performance software crypto support, implemented via the Intel Advanced Encryption Standard New Instructions (Intel AES-NI) in the Intel Xeon processor E5600 series and E5-2600 v2 series.

Device monitoring and statistics functions,such as Linux Ethtool MTU support, full RX/TX queue statistics and CRC error statistics, which enable improved system-level profiling, analysis and debug.

Support for additional Network Interface Cards(NICs), such as the Intel 82571EB Gibabit Ethernet controller, beyond those supported in the baseline Intel DPDK library.

6WIND also provides a range of optional add-on extensions to the Intel DPDK designed to improve the cost/performance of both physical and virtual networking appliances while enabling the use of the intel DPDK in software-defined networks. These optional add-ons include:

IPsec acceleration, achieved through integration of the Intel Multi-buffer Crypto for IPSec library;

Crypto acceleration via support of an external accelerator, the Intel Communications Chipset 89xx series, which is part of Intel's next-generation communications platform,codenamed "Crystal Forest"

Virtualization-related enhancements that maximize system performance by removing key I/O and communication bottlenecks include:

1. I/O Virtualization(IOV), an industry-standard approach for increasing the performance of virtual network appliances by bypassing the virtual switch within the hypervisor, thus removing the I/O performance constraints imposed by the virtual switch.

2. A virtual NIC(vNIC) driver that leverages communication between virtual machines via the virtual switch, enabling the efficient development and provisioning of systems with multiple VMs and significant East-West network traffic.

3. For system that require the ultimate level of performance for East- West traffic between VMs, a VM-to-VM driver enables direct VM-to-VM communication, bypassing the virtual switch while remaining fully compatible with industry-standard hypervisors.

These Intel DPDK enhancements and optional add-ons are maintained by 6WIND as private branch, regularly synchronized with Intel's on-going releases of the baseline library. They are delivered to customers either as a stand alone library or, for applications that also require high- performance packet processing software, and integrated within the 6WINDGate software solution.

The 6WINDGate packet processing software is designed to solve the problem of exploiting the potential packet processing performance of multicore processor through a fast pth-based architecture, while incorporating a comprehensive set of high performance networking protocols fully optimized for intel Xeon processor-based platforms.

5. Fast Path across (Virtual) Network Functions

Previous sections basically talk about the data path acceleration on one device with multiple threads/VMs sharing the physical resource. This section will talk about the data plane acceleration across multiple (virtual) network functions.

In NFV, layer 4-7 network functions are virtualized on top of the computing nodes. But sometimes, these vNFs are only used for session estabalishment, after which the packets can be handled by the L2/3 devices. Given that he higher layer the packet is being processed, the more challenge to its performance. So in some scenarios, it is desirable to offload the packet processing to the L2/3 fabrics,

eliminating the burden on the higher layer NFs. The scenario is depicted in Figure 2.

One vivid example is the ACL or Parental Control services. The ACL Network Function will determine the forwarding rules configured by its user, say, IP 5 tuples. After the session has been established, the ACL NF can inform the L2/3 devices about the forwarding rule in a control message. And the followed packets will be handled according to the logics.

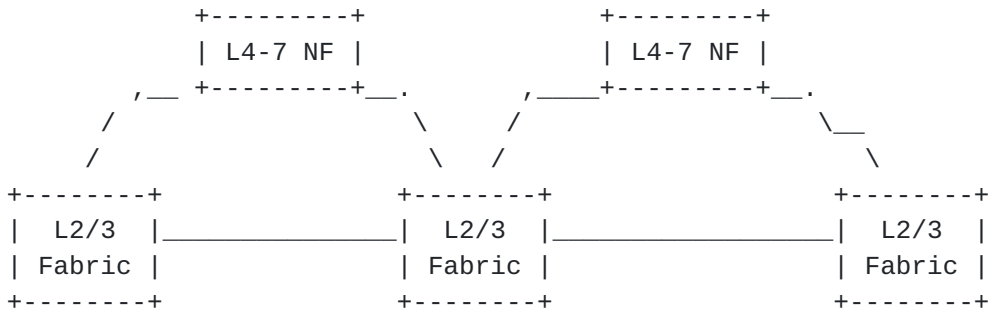

Figure 2: Fast Path across devices

5.1. ForCES

Forwarding and Control Element Separation (ForCES) [RFC5810][I-D.ietf-forces-protoextension] defines an architectural framework and associated protocols to standardize information exchange between the control plane and the forwarding plane .

In the Fast path offload senario described above, the ForCES protocols could be used or extended to serve as the communicaton protocols between the NF and L2/3 fabrics.

6. Open Questions to IETF

IETF has been design Layer 2&3 protocols, and most of them are dedicated to data plane processing. The efficient implementation of protocol and tailoring them for specific hardware/chipsets have not been considered as main-stream IETF work (there are indeed some thread anyway, e.g. tailor for M2M). But to make IETF protocols as efficient as possible is definitely within the scope of IETF. Below are some discussion of open questions to IETF w.r.t. the data plane process acceleration topic.

1. Importance. The game changing initiatives already started. NFV and further virtualization and decoupling practices are happening. Before the questions have been ported to specialized

hardware, but now the industry is changing the game. Do it need the standardization collaboration?

2. Relevance. As we authors believe it, to make IETF protocols as efficient as possible is definitely within the scope of IETF. Although implementation techniques are mostly software engineering practice and have no business with any SDOs, the abstract API design and exposure of lower layer capability will definitely benefit the data plane processing efficiency.

3. Necessity. Now that DPDK is already open source. But the experience in DPDK can feedback to IETF on how to improve the protocol design in promoting data plane acceleration effectiveness.

7. Acknowledgement

This work was inspired by the DPDK open source project.

Thank you for the discussion with Hui Deng, Dapeng Liu, and Lingli Deng on how to improve and promote this document.

8. IANA Considerations

To be specified.

9. Security Considerations

TBD.

10. Informative References

[DPDK] "Packet Processing - Intel DPDK, https://01.org/packet-processing/overview/dpdk-detail", .

[I-D.ietf-forces-protoextension] Salim, J., "ForCES Protocol Extensions", draft-ietf-forces-protoextension-02 (work in progress), June 2014.

[NFVE2E] "Network Functions Virtualisation: End to End Architecture, http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip", .

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5810] Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, March 2010.

Authors' Addresses

Zhen Cao
China Mobile
Xuanwumenxi Ave. No. 32
Beijing 100053
China

Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Qiao Fu
China Mobile
Xuanwumenxi Ave. No. 32
Beijing 100053
China

Email: fuqiao@chinamobile.com

Lingli Deng
China Mobile
Xuanwumenxi Ave. No. 32
Beijing 100053
China

Email: denglingli@chinamobile.com