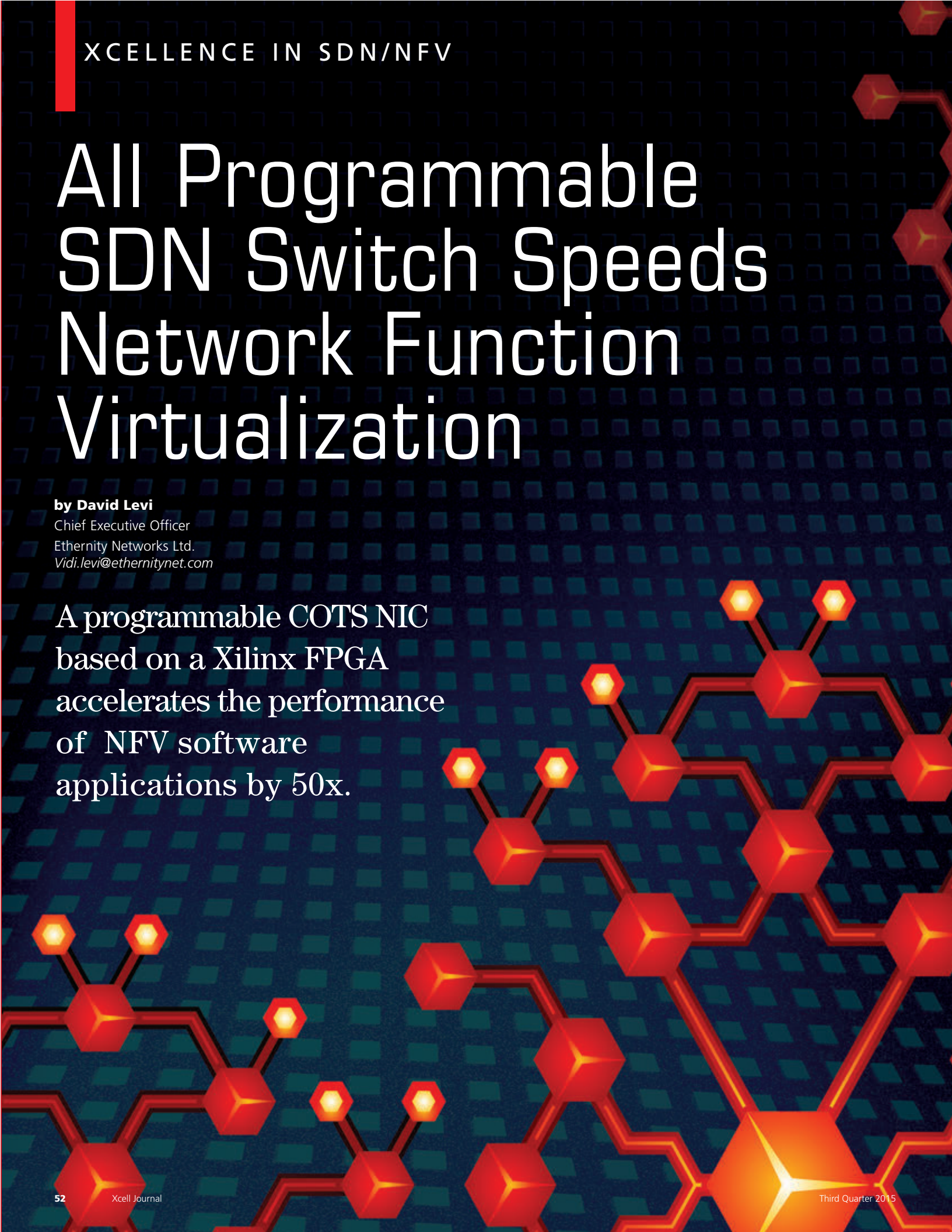
All Programmable SDN Switch Speeds Network Function Virtualization

by David Levi
Chief Executive Officer
Ethernity Networks Ltd.
Vidi.levi@ethernitynet.com

A programmable COTS NIC based on a Xilinx FPGA accelerates the performance of NFV software applications by 50x.



The shift toward network function virtualization (NFV) and software-defined networks (SDN) represents the most transformative architectural network trend in nearly 20 years. With their promise of open systems and network neutrality, NFV and SDN stand poised to make a far-reaching impact in shaping the communications networks and businesses of tomorrow.

We at Ethernity Networks are leveraging Xilinx® devices to bring truly open and highly programmable SDN and NFV solutions to the market sooner. Let's take a look at how Ethernity crafted its solution, after first exploring the promise and requirements of NFV and SDN.

UBIQUITOUS HARDWARE AND THE NFV/SDN REVOLUTION

The network infrastructure business of the last few decades has largely been a continuance of the mainframe business model in which a handful of large corporations offer proprietary infrastructure equipment that runs proprietary software, all purposefully built not to communicate with systems from competitors. In most cases, infrastructure vendors create custom hardware for each node of their customer's network, and they build each node to be minimally programmable and upgradable, ensuring that customers wanting to expand or upgrade their networks would need to buy next-generation equipment from the same vendor or make the futile choice of buying an entirely new network from another company but running into the same consequences.

In the last five years, operators, academics and vendor upstarts have been calling for a move to ubiquitous hardware, network neutrality, open systems and software compatibility by maximizing hardware and software programmability. NFV and SDN are at the vanguard of this trend, carrying the banner for this growing and sure-to-succeed revolution.

With NFV, companies run a variety of network functions in software on commodity, general-purpose hardware platforms, as opposed to running each specialized network task on customized and expensive proprietary hardware. Maximizing programmability on these open, ubiquitous platforms enables companies to run in data centers, or even in smaller networked nodes, many tasks heretofore performed by specialized hardware devices. NFV further aims to reduce the time it takes to establish a new network service by allowing operators to simply upload new network software of a given service to the commodity hardware resources as needed. This allows operators to scale networks easily and choose best-in-class functionality for their businesses instead of being forced to purchase and operate new proprietary hardware that affords limited software flexibility.

NFV can be effective because many nodes in a network share common functionality requirements. Those nodes with common requirements include switching and routing, classification of millions of flows, access control lists (ACL), stateful flow awareness, deep packet inspection (DPI), tunneling gateway, traffic analysis, performance monitoring, fragmentation, security, virtual routing and switching. NFV has its challenges, however. With exponential growth expected in Internet and data center traffic in the coming years, network infrastructure equipment must be able to handle vast increases in traffic. Software programmability alone won't be enough to enable generic hardware to scale easily with growing bandwidth demands. The ubiquitous hardware will need to be reprogrammable to optimize overall system performance. This allows vendors and operators to leverage NFV and SDN in a "work smarter, not harder" manner to meet growing network demands of the operators' end customers—consumers. A truly hardware- and software-programmable infrastructure is the only way to truly realize the vision of NFV and SDN.

SDN is a modern approach to networking that eliminates the complex and static nature of legacy distributed network architectures through the use of a standards-based software abstraction between the network control plane and underlying data-forwarding plane in both physical and virtual devices. Over the last five years, the industry has developed a standards-based data plane abstraction called OpenFlow that provides a novel and practical approach to dynamically provisioning the network fabric from a centralized software-based controller.

An open SDN platform with centralized software provisioning delivers dramatic improvements in the network agility via programmability and automation, while substantially reducing the cost of the network operations. An industry-standard data plane abstraction protocol like OpenFlow gives providers the freedom to use any type and brand of data plane device, since all the underlying network hardware is addressable through a common abstraction protocol. Importantly, OpenFlow facilitates the use of "bare-metal switches" and eliminates traditional vendor lock-in, giving operators the same freedom of choice in networking as can be found today in other areas of IT infrastructure, such as servers.

Because SDN is in its infancy, standards are still in flux. This means that equipment vendors and operators need to hedge their bets and design and operate SDN equipment with maximum flexibility, leveraging the hardware and software programmability of FPGAs. FPGA-based SDN equipment entering the market today

is quite affordable even for mass deployment. It offers the highest degree of hardware and software flexibility and maximizes compliance with OpenFlow.

THE NEED FOR PERFORMANCE ACCELERATION

Perhaps the most critical requirement for both NFV and SDN above and beyond openness is high performance. While NFV hardware will seemingly be less expensive than proprietary systems, NFV architectures will need to sustain competitively high data volumes, meet complex processing requirements for next-generation networking and be increasingly energy efficient.

In fact, the NFV Infrastructure Group Specification includes a special section describing the need for acceleration to increase network performance. The specification describes how a processor component can offload certain functions to a network interface card (NIC) to support certain acceleration functions, including TCP segmentation, Internet Protocol (IP) fragmentation, DPI, filtering of millions of entries, encryption, performance monitoring/counting, protocol interworking and OAM, among other acceleration capabilities.

The main engine driving this acceleration is the NIC, which is equipped with physical Ethernet interfaces to enable server connectivity to the network. As described in Figure 1, when a packet arrives the NIC, over 10GE, 40GE or 100GE ports, it is placed in a virtual port (VP) or queue that represents a specific virtual machine (VM) based on tag information such as IP, MAC or VLAN. The packet is then DMA'ed directly to the right VM located at the server for processing. Each virtual networking function (VNF) runs on a different VM, and certain networking functions require the use of multiple or even tens of VMs.

OpenFlow controls the hardware acceleration functions, such that these hardware acceleration functions located at the NIC can be viewed as an extension of the SDN switch.

NFV performance can be addressed for many functions by deploying multiple VNFs on multiple VMs and/or cores. This raises two main performance challenges for NFV. The first challenge is at the "vSwitch," which is typically a piece of software that processes network traffic between the Ethernet NIC and the virtual machines. The second performance challenge is balancing incoming 40/100GE data among multiple VMs. When adding IP fragmentation, TCP segmentation, encryption or other dedicated hardware functions, NFV software requires assistance to meet the performance needs and reduce the power. Ideally, it should be compact to reduce the amount of real estate required to house the network equipment.

To address NFV challenges and the variety of networking functions, NIC cards for NFV and SDN must be extremely high performance but also as flexible as possible.

In an attempt to be first to market with NFV hardware, several chip vendors have proposed platforms for NIC cards, each with some degree of programmability. Intel today is the main NIC component provider, equipped with its DPDK package for packet-processing acceleration. EZchip offers its NPS multithreaded CPU running Linux and programmed in C. Marvell offers two all-inclusive data plane software suites for its Xelerated processor for both metro Ethernet and the Unified Fiber Access Application, which consist of an application package running on the NPU and a control-plane API running

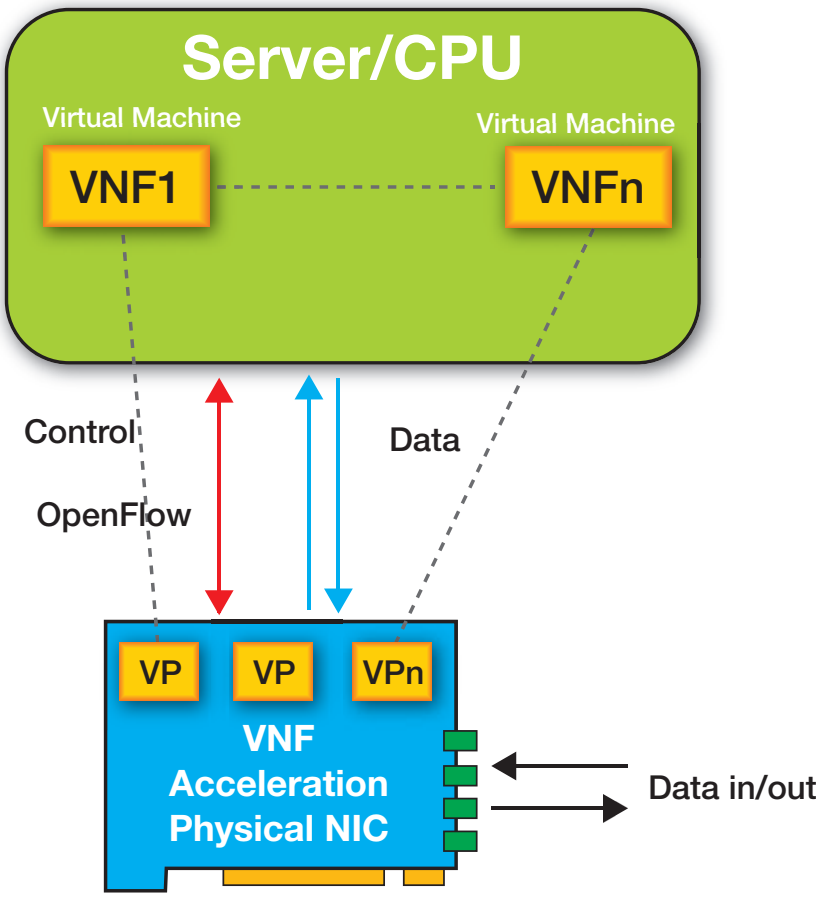


Figure 1 – When a packet arrives, the NIC enters a virtual port (VP) that represents a specific virtual machine. The packet is then sent through DMA to the right virtual machine at the server for processing.

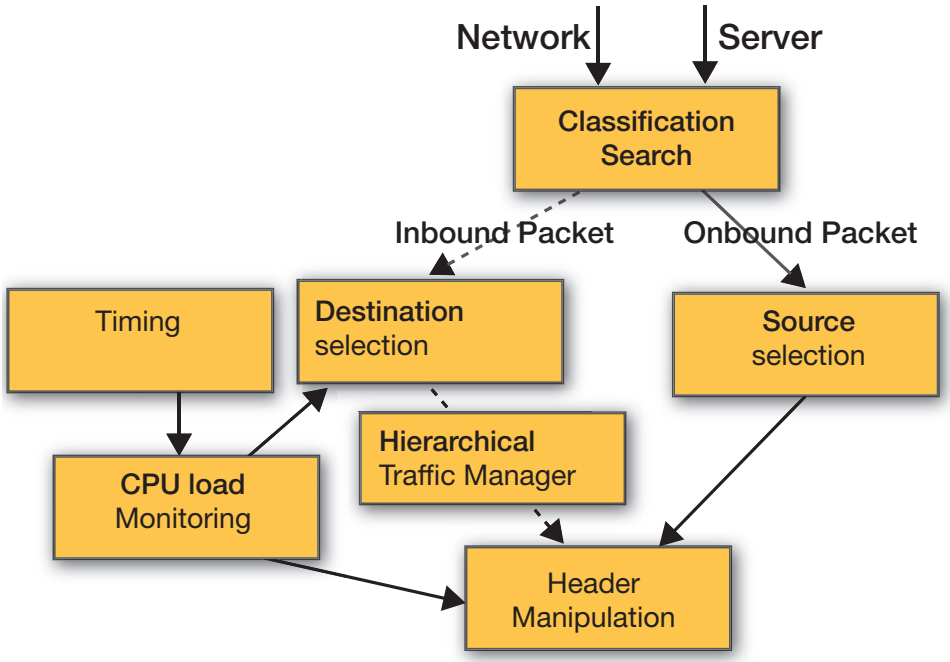


Figure 2 – This high-level block diagram shows the virtual machine's load balancing and switch.

on a host CPU. Cavium has opted for a more generic software development kit for its Octeon family. Broadcom, Intel and Marvel L2/L3 switches are mainly used for search and vSwitch offload. Meanwhile, Netronome's new Flow-NIC is equipped with software that runs on that company's specialized network processor hardware.

While all of these offerings are claiming to be open NFV approaches, they really aren't. All of the approaches involve rigid and, arguably, too restrictive hardware implementations that are only programmable in software and rely once again on rigid, proprietary hardware implementations in SoCs or standard processors.

ALL PROGRAMMABLE ETHERNITY NIC FOR NFV PERFORMANCE ACCELERATION

To add programmability while also improving performance dramatically, many companies are examining a hybrid approach that pairs an off-the-shelf CPU with an FPGA. Over the last two years, a number of data center operators—most notably, Microsoft—have published papers describing the dramatic performance increases they've gained with a hybrid architecture. In the case of Microsoft, a white paper titled "The Catapult Project" cites a 95 percent performance increase at a cost of only a 10 percent power increase. Intel cited the potency of this combination of FPGA-plus-CPU in data center NICs as the primary reason for its $16.7 billion acquisition of the No. 2 player in FPGAs, Altera Corp.

The same hybrid CPU-FPGA approach is applicable for NFV, which runs virtual networking functions on virtual machines. In this approach, the FPGA serves as a complete programmable NIC that can be extended to accelerate the virtual-network functions that run on the server's CPUs/VMs.

But a NIC card based entirely on FPGAs is the ideal COTS hardware architecture for NFV. Multiple FPGA firmware vendors can provide firmware for NFV performance acceleration running on the FPGA NIC. And FPGA companies have recently developed C compiler technologies such as Xilinx's SDAccel™ and SDSoC™ development environments to enable OpenCL™ and C++ design entry and program acceleration, further opening up NFV equipment design to a greater number of users.

To accelerate NFV performance, NFV solution providers increase the number of VMs with a goal of distributing the VNFs on multiple VMs. When operating with multiple VMs, new challenges arise related to balancing the traffic load between the virtual machines while supporting IP fragments. In addition, challenges also exist in supporting switching between VMs and between VMs and the NIC. A pure software-based vSwitch element simply doesn't provide adequate performance to address these challenges. The integrity of the VMs must also be maintained so that the VMs store specific bursts adequately and do not deliver packets out of order.

Focusing on solving the performance issues for NFV, Ethernity's ENET FPGA firmware is equipped with a virtual switch/router implementation that enables a system to accelerate vSwitch functionality to switch data based on L2, L3 and L4 tags, while maintaining a dedicated virtual port for each VM. If a specific VM is not available, the ENET can store up to 100 milliseconds of traffic; then, upon availability, it will release the data through DMA to the VM. Equipped with delay measurement capabilities through an implementation of a standard CFM packet generator and a packet analyzer, our ENET can measure the availability and health of a VM and instruct the ENET's stateful load balancer regarding the availability of each VM for load distribution. The packet reorder engine can maintain the order of the frame if, for example, a packet moves out of order, which can result in the use of multiple VMs for one function.

Figure 2 depicts a block diagram of the the VM load-balancing ENET solution.

In Figure 2, the classification block performs hierarchical classification for L2, L3 and L4 fields to maintain a route for connection and flow supporting the long-lived TCP (telnet, FTP and more)

that doesn't immediately close. The load balancer must ensure that multiple data packets carried across that connection do not get load-balanced to other available service hosts. ENET includes an aging-mechanism feature to delete nonactive flows.

In the classification block, we configured the balancing hash algorithm based on L2, L3 and L4 fields. The algorithm includes fragmentation such that the load balancer is able to perform balancing based on inner tunnel information (such as VXLAN or NVGRE), while an IP fragment connection can be handled by a specific connection/CPU. For VM-to-VM connection, the classifier and search engine will forward the session to the destination VM instead of the vSwitch software. Meanwhile, a classifier feature assigns a header manipulation rule for each incoming flow based on its outgoing route, with eyes on modifying the IP address or offloading protocols.

For each new flow, the destination selection block's load balancer assigns a destination address from the available VM according to the Weighted Round Robin (WRR) technique. The WRR is configured based on the information derived from the VM load-monitoring block.

The hierarchical traffic manager block implements hierarchical WRR between an available VM and maintains an output virtual port for each VM to include three scheduling hierarchies based on priority, VM and physical port. The CPU hierarchy represents a certain VM, and the priority hierarchy may assign weights between different services/flows that are under the service of a specific VM. Operating with external DDR3, the ENET can support buffering of 100 ms to overcome the momentary load of a specific VM.

The VM load monitoring uses the ENET Programmable Packet Generator and Packet Analyzer for carrier Ethernet service monitoring, which complies with Y.1731 and 802.1ag. The VM load-monitoring block maintains information on the availability of each CPU/VM, using metrics such as Ethernet CFM Delay Measurements Message (DMM) protocol generation toward a VM. By time-stamping each sent packet and measuring the delta time between send and receive, the block can determine the availability of each VM and, based on that, instruct the destination selection block on the available VMs.

The Source Selection block determines what outbound traffic sent from the host to the user will be classified and determines the source of that packet.

The Header Manipulation block in ENET will perform network address translation (NAT) to replace the incoming address with the right VM IP address to enable the NIC to forward the flow, packet or service to the right VM. For outbound traffic, the NAT will perform the reverse action and will send out the packet to the user with its original IP address. The Header Manipulation

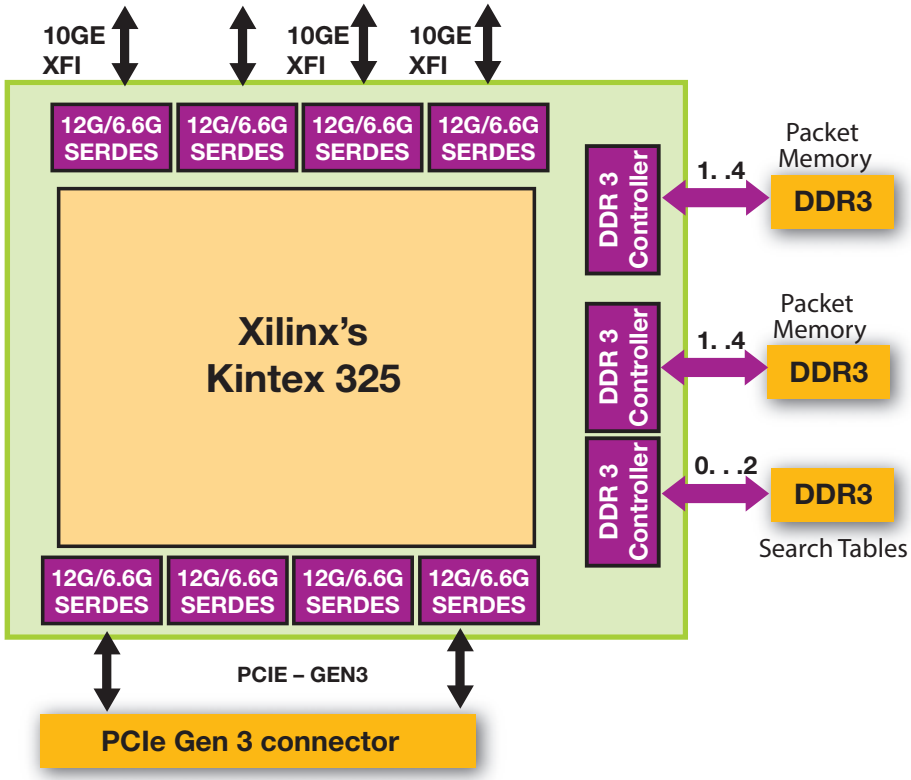


Figure 3 – A Xilinx Kintex FPGA is at the heart of Ethernity's NFV network interface card.

block also performs tunnel encapsulation. Here, the Header Manipulation block will execute the action rules assigned by the classifier from the classification, and will strip out tunnel headers or other headers to or from the CPU operation. In the reverse direction, it will append the original tunnel toward the outgoing user port.

As the number of subscribers to an operator's network increases, the size of the flow tables can quickly grow to exceed the cache capacity of standard servers. This is especially the case with current OpenFlow systems, which have table entries that require 40 different fields, IPv6 addresses, Multiprotocol Label Switching (MPLS) and Provider Backbone Bridges (PBB). The ENET search engine and parser can support classification on multiple fields and serve millions of flows, thus offloading the classification and search function from the software appliance.

And finally, with the ENET packet header manipulation engine, the ENET can offload any protocol handling and provide raw data info to the VM together with TCP segmentation, or interworking between various protocols, including 3GPP protocols for virtual EPC (vEPC) implementation, XVLAN, MPLS, PBB, NAT/PAT and so on.

In addition to the firmware, Ethernity has also developed an NFV NIC that we call the ACE-NIC (Figure 3). To create the NIC, we integrated our ENET SoC firmware (already deployed in hundreds of thousands of systems in carrier Ethernet networks) into a single Xilinx Kintex®-7 FPGA. We also integrated into the same FPGA the functionality of five discrete components: NIC and SR-IOV support; network processing (including classification, load balancing, packet modification, switching, routing and OAM); 100-ms buffering; frame fragmentation; and encryption.

The ACE-NIC is an OpenFlow-enabled hardware acceleration NIC, operated in COTS servers. The ACE-NIC accelerates performance of vEPC and vCPE NFV platforms by 50 times, dramatically reducing the end-to-end latency associated with NFV platforms. The new ACE-NIC is equipped with four 10GE ports, along with software and hardware designed for an FPGA SoC based on Ethernity's ENET flow processor, supporting PCIe® Gen3. The ACE-NIC is further equipped with onboard DDR3 connected to the FPGA SoC, to support 100-ms buffering and search for a million entries.

The Ethernity ENET Flow Processor SoC platform uses a patented, unique flow-based processing engine to process any data unit in variable sizes, offering multiprotocol interworking, traffic management, switching, routing, fragmentation, time-stamping and network processing. The platform supports up to 80 Gbps on a Xilinx 28-nanometer Kintex-7XC7K325T FPGA, or higher throughput on larger FPGAs.

The ACE-NIC comes with basic features such as per-frame time-stamping that's accurate within nanoseconds, a packet generator, a packet analyzer, 100-ms buffering, frame filtering and load balancing between VMs. To serve multiple cloud appliances, it also has the ability to assign a virtual port per virtual machine.

Furthermore, the ACE-NIC comes with dedicated acceleration functions for NFV vEPC. They include frame header manipulation and offloading, 16K virtual-ports switch implementation, programmable frame fragmentation, QoS, counters and billing information, which can be controlled by OpenFlow for the vEPC. With its unique hardware and software design, the ACE-NIC accelerates software performance by 50x.

THE ALL PROGRAMMABLE ETHERNITY SDN SWITCH

Similarly, Ethernity integrated the ENET SoC firmware in an FPGA to create an All Programmable SDN switch, with support for OpenFlow version 1.4 and complete carrier Ethernet switch functionality, accelerating time-to-market for white-box SDN switch deployment.

The ENET SoC Carrier Ethernet Switch is an MEF-compliant L2, L3 and L4 switch/router that can switch and route frames with five levels of packet headers, between 16,000 internal virtual ports distributed over 128 physical channels. It includes support for FE, GbE and 10GbE Ethernet ports, and four levels of traffic-management scheduling hierarchy. With its inherent architecture to support fragment frames, the ENET can perform IP fragmentation and reordering of functions with technology of zero copy, such that segmentation-and-reassembly does not require dedicated store and forward. Furthermore, ENET has an integrated programmable packet generator and packet analyzer to ease CFM/OAM operation. Finally, the ENET can operate in 3GPP, LTE, mobile backhaul and broadband access. It supports interworking between multiple protocols, all with zero-copy operation and without a need to reroute frames for header manipulation.

Clearly, the communications industry is at the beginning of a new era. We are sure to see many new innovations in NFV and SDN. Any emerging solution for NFV performance acceleration or an SDN switch must have the ability to accommodate support for new versions of SDN. With Intel's acquisition of Altera and the increasing number of hardware architectures seeking greater degrees of programmability, we will certainly see a growing number of hybrid processor-plus-FPGA architectures along with new, innovative ways to implement NFV performance acceleration.

FPGA-based NFV NIC acceleration can provide the flexibility of NFV based on general-purpose processors while at the same time supplying the necessary throughput that the GPP cannot sustain, while performing certain network function acceleration that GPP can't support. By efficiently combining the SDN and the NFV in the FPGA platform, we can achieve the design of All Programmable network devices fostering the innovation to a new ecosystem for IP vendors in network applications.