A New Era of Network Processing

By Bob Wheeler
Senior Analyst

October 2013

The Linley Group

www.linleygroup.com

This paper examines traditional network-processor (NPU) architectures, technology trends driving new requirements, limitations of NPUs and CPUs, and new architectures that overcome these challenges.

A Brief History of Network Processors

Network processors, as we define them, have been around for more than a decade. This category of chips grew out of the need for greater flexibility at the edge of service-provider networks. In edge routers, high-end NPUs displaced the traditional forwarding ASICs used in earlier designs. For example, Cisco's Toaster 2 network processor enabled that company's 10000 edge router introduced in April 2000. Alcatel-Lucent gained its FP1 NPU through the 2003 acquisition of TiMetra, resulting in the 7750 service router. Redback Networks, which was acquired by Ericsson in 2007, developed its PPA NPUs for the SmartEdge router line.

In parallel with these OEM-internal NPU developments, a bevy of merchant vendors were developing NPUs. Some vendors focused on high-end packet processing, essentially competing with OEM-internal designs. Others addressed multiservice designs, particularly for access networks, where NPUs replaced ATM chips (i.e., SARs) as well as lower-speed communications processors like Motorola's PowerQUICC. After years of consolidation, only a handful of merchant NPU vendors remain active. These include PMC-Sierra (Wintegra) for access NPUs as well as Broadcom (Sandburst), EZchip, Marvell (Xelerated), and Netronome (Intel IXP) for high-end NPUs.

In our product taxonomy, an NPU is a software-programmable device that is optimized for processing packets. Specifically, it is designed to handle the tasks such as header parsing, bit-field manipulation, table lookups, packet modification, data movement, and traffic management. To handle many independent packets simultaneously, NPUs use various parallel processing techniques. Programmability (the ability to execute a sequence of instructions) is an important characteristic of network processors, because it provides the flexibility needed to address new or evolving protocols and features.

Others sometimes use the term network processor to refer to embedded processors designed for networking applications. These processors typically include one or more high-performance CPUs that implement an ARM, MIPS, Power, or x86 instruction set. Some of these embedded processors include packet-processing accelerators to offload the general-purpose CPUs. In most networking designs, however, embedded processors implement the complete control plane in addition to data-plane functions. By contrast, NPUs usually implement the complete data plane but not the control plane.

Traditional NPU designs have focused on Layer 2 and Layer 3 processing, which is relatively simple. As a result, these NPUs have not included floating-point units (FPUs), memory-management units (MMUs), or hierarchical memory caches. By eliminating general-purpose features and focusing on just the requirements for packet processing, NPU designers have packed hundreds of tiny packet engines on a single chip. These

chips also include hardware coprocessors that offload specific tasks from the programmable engines. These coprocessors include header parsers, classification engines, lookup engines, counter/statistics engines, and traffic managers.

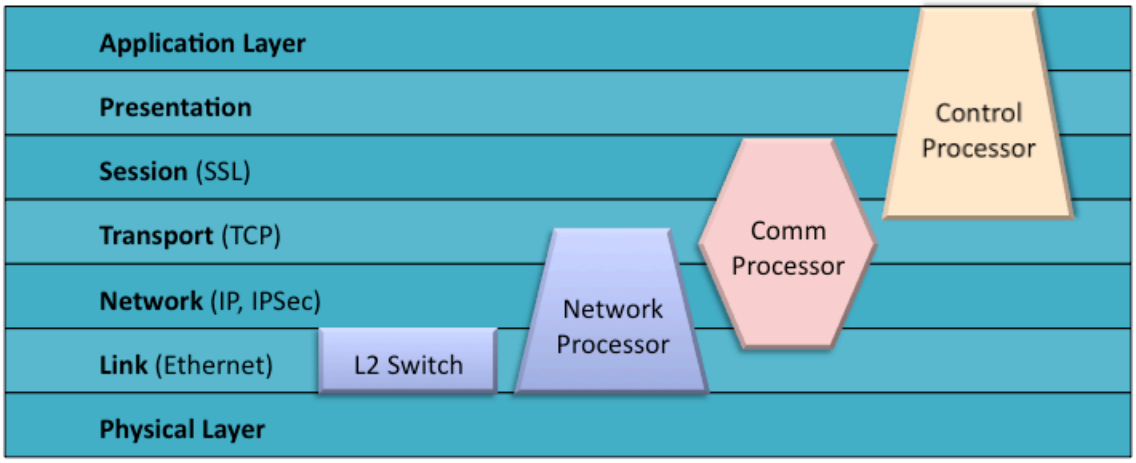

Figure 1. Networking processing taxonomy.

The choice of how to organize the many packet engines in an NPU impacts the chip's software model. Designers have used many different architectures, including pipelined designs, fully symmetric arrays, and hybrid approaches. The pipelined model passes a packet from engine to engine as it is processed. The symmetric multiprocessor (SMP) model allows each packet engine to execute its own set of software, which enables a single packet engine to completely process a packet. One merchant vendor uses a single-instruction multiple-data (SIMD) technique to organize its packet engines. Each NPU design is unique, which is unfavorable for software teams that must master these esoteric architectures.

Technology's Inescapable Advance

Over the past decade, many NPU designs have successfully scaled in performance while preserving their fundamental architectures. Reusing an architecture preserves the OEM's software investment and reduces silicon-development costs. Existing NPU designs, however, may be unable to meet new-feature requirements. Although carrier networks have traditionally evolved slowly, economic forces are now compelling service providers to deliver new revenue-generating services while also reducing costs. As a result, equipment vendors must address customers' feature requirements driven by emerging technologies.

Software-Defined Networking

One way to reduce the operational cost of a network is to make it easier to manage. Software-defined networking (SDN) can reduce management costs by centralizing

control and provisioning. Whereas traditional switches and routers implement both a control and data plane, SDN physically separates the control-plane function from the forwarding function. A centralized controller then manages a group of forwarding nodes, giving the network manager a global view of those nodes.

Another benefit of SDN is it can remove limitations imposed by existing protocols such as Ethernet or IP. The OpenFlow protocol for SDN enables network managers to define traffic flows and a set of actions for those flows. The central controller pushes flow-table entries to the forwarding nodes, which then take the specified action for a given flow. This abstraction of the forwarding function gives network architects greater flexibility in traffic engineering. Although much of the recent discussion around OpenFlow has centered on data-center networks, carriers actually provided funding for the protocol's early development.

Virtualization

The term virtualization can have many different meanings depending on its context. In enterprise data centers, server virtualization is well established and has delivered improved utilization of resources. In networks, existing technologies such as virtual routing and forwarding (VRF) instances and VLANs can be viewed as forms of virtualization. Multitenancy is another trend driving the development of new virtualization features and protocols. For example, VXLAN and NVGRE are new overlay or tunneling protocols that enable the logical separation of customer traffic in multitenant networks.

A more radical form of virtualization driven by carriers is Network Functions Virtualization (NFV), which is under development in ETSI. The goal of NFV is to reduce the number of proprietary platforms service providers must deploy by migrating network functions to common hardware platforms. Although some processing-intensive functions lend themselves to the use of commodity server hardware, forwarding functions require greater performance and power efficiency than these platforms can deliver. At the other end of the spectrum, high-volume Ethernet switches deliver performance and efficiency at the cost of flexibility. In our view, neither "commodity" servers nor switches can supplant NPU-based edge routers.

Advanced Services

Edge routers are now implementing services that previously required dedicated hardware or were handled by customer-premises equipment (CPE). To improve management and billing for tiered services, edge routers are implementing application visibility using deep-packet inspection (DPI). To implement LTE gateway functions, edge routers must terminate IPSec traffic from cellular base stations. Another function routers are handling is large-scale (or carrier-grade) network-address translation (NAT), which mitigates IPv4 address exhaustion. Some edge routers even implement firewall and intrusion-prevention services. Whereas traditional routers focused on Layer 3 packet forwarding, many of these advanced services require stateful processing using Layer 4-7 information.

Another requirement that also relates to virtualization is service chaining. Rather than simply forwarding traffic based on a destination address, packets must be classified and assigned to service flows. Traffic associated with a specific flow is then forwarded (or chained) through various service functions. Equipment that implements service chaining, such as an edge router, must classify and send traffic through a chain of internal virtualized services. Such an implementation represents a much higher level of network-traffic abstraction than traditional routing.

NPU Architectures Reach the Breaking Point

Because traditional NPU designs impose many limitations, these designs are unable to handle the some of the new requirements discussed above. The most fundamental problems relate to how the NPU's packet engines are organized and programmed. Designed for Layer 3 packet processing, some NPUs pass individual packets through a processing pipeline. Such a design cannot reassemble packets into flows for Layer 4 processing or Layer 7 inspection. Pipelined architectures also complicate programming, as they require dividing the packet-processing software into multiple stages. This approach can also lead to poor resource utilization, as one stage becomes the critical path while others are underutilized.

To hide hardware complexity from the programmer, most NPUs offer a single-image programming model. Although this model simplifies software development, it requires all packet engines and threads to run the same code. This model works well for parallel processing of packet forwarding, but it disallows assigning some packet engines to advanced services. As a result, many edge router designs must rely on services blades for these features, and this approach limits performance scaling for advanced services.

Because edge routers must deliver deterministic forwarding performance, traditional NPU designs use simple memory architectures that can create usage limitations. For example, many NPUs have only internal memories for storing instructions (control store) for the packet engines. Although this design maximizes performance, it severely limits code size. Whereas Layer 3 forwarding code can require only a few thousand instructions, advanced services may require more than ten thousand instructions. Another example is function-specific memories, which limit how flexibly external memories can be assigned to various functions such as lookup tables, statistics, and packet buffers.

Many NPUs use proprietary packet engines that implement an instruction set optimized for packet processing. Although these designs are power efficient and minimize silicon area, they prevent the use of well understood programming languages and tools. In fact, no shipping merchant NPU can be programmed in full ANSI C and some require assembly-code (microcode) programming. Because each NPU architecture is unique, these designs create a large learning curve for programmers. The lack of standard tools also complicates debugging. Most programmers are familiar with Linux environments and GNU tools, but typical NPUs support neither of these.

Another problem with NPU-software development is that there exists a small number of software engineers capable of programming these devices. This limits the resources available for new-feature implementation to a highly specialized team, which is difficult to staff. Separate from the difficulty of coding and debugging, the limited pool of programmers can further delay the introduction of new features.

Why Not Use CPUs?

In parallel with NPUs, general-purpose processors have also scaled in performance thanks to multicore designs. Since 2006, vendors have been shipping multicore embedded processors with 16 or more general-purpose CPUs. These system-on-a-chip (SoC) designs integrate all required I/O for typical networking applications. In 2013, networking SoCs with eight or more cores are available using ARM, MIPS, Power, or x86 instruction sets. Unlike typical NPUs, the CPUs in these chips employ a standard instruction set and store program code in external memory.

Admittedly, the distinction between an NPU and a multi-CPU processor with networking accelerators is becoming less obvious. Several high-end multicore SoCs include packet-processing hardware blocks, which allow these chips to handle throughputs of 40Gbps or more. Although these chips can perform the same functions as an NPU, however, they cannot match the performance per watt of an NPU in typical switching and routing applications. We estimate the newest multicore processors deliver about 1Gbps/W, whereas the newest NPUs provide nearly 5Gbps/W. In an edge-router line card, this difference translates directly to network-port density.

Meet the New NPU

Recognizing the limitations of Layer 3 oriented NPUs, two OEMs have developed processors that can be viewed as hybrids of an NPU and a multicore processor. One merchant vendor is also developing such a chip.

Cisco deserves credit for delivering the first such design in its QuantumFlow Processor (QFP) for the ASR 1000, which was introduced in March 2008. The QFP implements advanced services, including load balancing, firewall/NAT, and IPSec. The first-generation QFP integrates 40 multithreaded RISC CPUs that operate at up to 1.2GHz and share level-two caches connected to external memory. The QFP supports full ANSI C programming, while a separate configurable device handles traffic management. Designed for 20Gbps of system throughput, the original QFP handled a maximum of 23 million packets per second (Mpps).

In 2H12, Cisco shipped a second-generation QFP using 64 CPUs at 1.5GHz. Using a pair of these packet processors plus a pair of traffic managers, the ASR 1000 is now capable of 100Gbps throughputs. This configuration's maximum packet rate, however, is only 58Mpps, or slightly less than 40Gbps for 64-byte packets.

Cisco uses its most recent internal NPU design, the nPower X1, in the new CRS-X and NCS 6000 routers. The nPower X1's leading integration enables a 10x100G line card, which achieved first customer shipments in July 2013. Each NPU handles a pair of 100G Ethernet ports or 400Gbps of aggregate throughput. The nPower X1 is rated at only 300 million packets per second, however, meaning it cannot sustain line-rate forwarding for 64-byte packets. In addition, the line card handles only Layer 3 forwarding rather than the advanced features provided by the ASR 1000. Although Cisco withheld the nPower X1's architecture details, the only unique feature it touted was in-service software upgrades without packet loss or topology loss.

Ericsson is the first vendor to develop an NPU that handles advanced features without compromising performance or the scale of services. In March 2013, the company disclosed its SNP 4000 NPU, which integrates thousands of multithreaded 64-bit RISC cores, supports a GNU tool chain for full C/C++ programming, and runs SMP Linux. The new NPU combines 200Gbps of packet processing with a 100Gbps hierarchical traffic manager on a single chip. The SNP 4000 is designed for line-rate processing at these throughputs, meaning it is capable of processing 300Mpps, as Table 1 shows. The first product based on the new chip, which will be available in 4Q13, is a 20-port 10G Ethernet line card for the SSR 8000.

From a programmer's perspective, the SNP 4000 looks more like a massive multicore processor than a traditional NPU. It supports true C-language programming, standard GNU tools, and an unmodified Linux kernel. The chip supports virtual memory and offers a unified memory model, which enables scaling of various resources such as lookup tables and packet queues. For example, the NPU can classify and police 20 million flows, which provide the scale to guarantee quality of service (QoS) across multiple applications. Given the relative ease of programming the SNP 4000, Ericsson's claim of order-of-magnitude improvements in feature velocity over traditional NPUs appears credible.

A merchant NPU vendor, EZchip is developing a new architecture somewhat similar to Ericsson's design. The company's first chip based on this design is the NPS-400, which targets 400Gbps (or 600Mpps) of packet processing and integrates a pair of 240Gbps traffic managers. It uses 256 multithreaded 32-bit RISC cores that include special-function units carried over from the company's shipping line of NPUs. Like the SNP 4000, the NPS-400 promises ANSI C programming using GNU tools as well as SMP Linux support. Unlike Ericsson's design, however, the NPS-400 requires Linux kernel modifications due to its use of fixed memory mapping for some internal memories. The NPS-400 is due to sample in 2Q14, which we estimate will put it about two years behind the SNP 4000 in reaching production.

By supporting a GNU tool chain and SMP Linux, these new Ericsson and EZchip NPUs enable flexibility similar to that of multicore processors. This means the chips can handle virtually any new protocol or feature and adapt more quickly to carriers' requirements. For example, the NPUs can implement OpenFlow for SDN and adapt to new versions; handle multiple applications for service chaining; provide stateful processing for features

such as NAT and firewalls; deliver application visibility for billing, SLA enforcement, or bandwidth management.

Programmers should find these new NPUs no more difficult to program than high-end embedded multicore processors. The Ericsson and EZchip NPUs offers virtually unlimited code space as well as support for existing Linux code. These attributes expand the pool of software engineers capable of data-plane programming. In fact, OEMs could even open NPU programming to their service-provider customers.

	Cisco 2nd-gen QFP	Ericsson SNP 4000	EZchip NPS-400
PP Throughput, Rate	100Gbps, 58Mpps	200Gbps, 300Mpps	400Gbps, 600Mpps
PP+TM Chips	4 chips	1 chip	1 chip
Services	L2-L7	L2-L7	L2-L7
ANSI C	Yes	Yes	Yes
GNU Tool Chain	Not disclosed	Yes	Yes
SMP Linux	No	Yes	Yes
1st Silicon	2H11*	2Q12	2Q14
Production	2H12	4Q13 (est)	4Q15*

Table 1. Comparison of OEM-internal and merchant network processors. (Source: vendors, except *The Linley Group estimate)

There are always tradeoffs in processor designs and these new chips are no exception. Although the new breed of NPUs should deliver throughputs competitive with those of traditional designs, performance will surely decrease as advanced services are added. So long as these performance tradeoffs are predictable, however, carriers should welcome the platform unification and simplification of routers based on these new NPUs.

Revolution over Evolution

Following some initial growing pains and consolidation of the vendor base, network processors have become the de facto standard for edge-router data planes. Traditional NPUs successfully balanced the need for programmability with requirements for power efficiency. They have enabled routers to handle many new features and protocols to keep up with evolving carrier networks. To differentiate these strategic platforms, leading OEMs continue to invest in internal NPU designs. Many of these designs, however, are reaching a dead end.

An upheaval in service-provider networks is increasing the pace of change. Simple packet forwarding is giving way to flow processing. Services once implemented in dedicated equipment are being consolidated into router platforms. Wired and wireless

networks are converging. Traditional NPUs designed for Layer 3 routing cannot keep up. Suddenly, evolutionary NPU designs are no longer viable.

To meet carriers' emerging demands, OEMs and merchant vendors alike must develop bold new network-processing architectures. These designs must blend the favorable attributes of traditional network processors—high performance and power efficiency—with the greater flexibility of multicore processors based on general-purpose CPUs. Ericsson is leading this charge with the industry's first true C-programmable processor capable of running Linux and delivering 200Gbps throughputs. EZchip should be the first merchant NPU vendor to offer similar capabilities. Although equipment based on these new designs must still be proven in the field, we see nothing less than a new era of network processing emerging. ♦

About the Author

Bob Wheeler is a senior analyst at The Linley Group and co-author of A Guide to Network Processors. *The Linley Group offers the most comprehensive analysis of the networking-silicon industry. We analyze not only the business strategy but also the technology inside all the announced products. Our weekly publications and in-depth reports cover topics including Ethernet chips, network processors, multicore embedded processors, and wireless base-station processors. For more information, see our web site at www.linleygroup.com.*

This paper is sponsored by Ericsson, but all opinions and analysis are those of the author.