TECHNOLOGY TRENDS

Next Gen Switch/Router Design Issues

Bart Stuck and Michael Weingarten

Tomorrow's LANs, SANs and WANs will need more coprocessors, plus bigger buffers and programmable-on-the-fly, non-blocking switch/routers

Last month, we argued that fundamental changes in demand have substantially altered the requirements for next-generation enterprise switch/routers. Not only will they need to become faster, but they also will need to support substantially increased functionality and become more programmable (see *BCR*, September 2004, pp. 54–59). We also argued that the requisite hardware doesn't exist.

In this article, we wish to further the discussion by addressing the following questions:

- Why are current switch/router and port/line card designs unable to support the necessary speed and functionality?
- What are possible solutions?
- Are new designs achievable any time soon?

Before we can answer these questions, however, we need to establish some basic architectural groundwork on switch design (for an analogous discussion of line card design issues, see "What About Line Cards?", pp. 46–47).

Basic Network And Switch/Router Designs

Figure 1 is a useful starting point, illustrating how multiple endpoints in generic enterprise networks are connected to central switch/routers: PCs via LAN network interface cards (NICs) installed in the PCs; storage area network (SAN) switches via host bus adapters (HBAs); and wide area networks (WANs) via switch/router port/line cards.

The generic switch/router architecture shown in Figure 2 consists of port processors connected to a central switching fabric. Modern switch/routers have these switch architectures on their system-level backplanes as well as on their port/line cards.

Each port processor typically consists of data plane and control plane processors which perform the following functions:

- Receive each packet and verify that it is legitimate and uncorrupted.
- Parse each packet to identify different fields and classify the packet.
- Replace/encapsulate the incoming packet header and apply an appropriate outgoing header, according to the parsing classification rules. Encapsulation examples include Intel's Advanced Switching (AS) and the IETF's Multiprotocol Label Switching (MPLS).
- Perform additional actions based on classification rules, such as traffic shaping, compression and encryption

All packets travel through the data plane, which performs the basic receiving/parsing/header replacement functions. Packets that require

Executive Summary

The new demands that will be placed on switches and routers in next-generation networks will require new designs of port/line cards, port processors, switching interfaces and switch fabrics. These components of current products cannot support the higher bandwidth and functional requirements that will be needed in future networks.

What is required are big, non-blocking, programmable switches. Among the concepts that may be useful in creating such products are those being worked out by the Advanced Telecom Computing Architecture (ATCA), as well as the DARPA-funded concept of active networks. In addition, a set of standards called Forwarding and Control Element Separation (ForCES) may help designers create switch/routers that can be programmable-on-the-fly.

Among the challenges to creating such new switch/router designs are the need for more powerful and flexible network processors, and the requirement that such higher-function products still be affordable□

Bart Stuck (barts@signallake.com) and Michael Weingarten (mikew@signallake.com) are managing directors of Signal Lake, an early-stage telecom venture capital fund (Westport, CT and Boston).

exceptional processing—such as additional table lookups for classification and handling—are handled by the control plane.

In the past, switch designers counted on only about 20 percent of packets requiring exceptional processing (e.g., Layer 3 routing in addition to layer Layer 2 switching), so they designed control planes with only about 20 percent of the capacity of the data planes. Packets that required exceptional processing were described as traveling via the "slow path" through the control plane, while the rest took the "fast path" through the data plane.

More recently, vendors such as Cisco, Foundry and Juniper have developed switch/routers that process all packets through the control plane. In these types of switch/routers, port processing is a pipeline with a sequence of steps, in which each stage must take the same amount of time. If some stages take longer, packets can be delayed or dropped.

Once packets are handled by the port processor, they proceed to the *switching interface*, in which each packet is read for information regarding packet size and destination port. Packets that are too large to be handled efficiently through the switching fabric are split into multiple, smaller packets.

The switching interface is a Layer 2, dataplane function with minimal control plane activity. It determines which destination port the packet will be sent to, by:

- Maintaining constant awareness through each processor clock cycle regarding which destination ports are in use (for flow control); and
- Assigning an appropriate idle destination port for each packet.

The *switch fabric* also sends flow control information back to the port processor. As is suggested by the term "fabric" (i.e., an interleaving grid laid out in an xy pattern) switching fabrics are grids of inputs and outputs, in which any input can be switched to any output. In a blocking switch fabric, two packets that are trying to go through the fabric to *two different ports* may get blocked. Even in a non-blocking switch fabric, two packets attempting to reach *the same port* could get blocked, unless the traffic is flow controlled or queued.

To the best of our knowledge, all enterprise Ethernet switch fabrics today use blocking designs, as part of tiered LAN architectures with concentration occurring at each tier. This has worked fine historically, since the enterprise systems could be designed with enough capacity at each tier to avoid significant blocking. Also, since LANs were transmitting data traffic, as opposed to time-critical voice, having some packets blocked was not that big an issue.

FIGURE 1 Enterprise Switching Block Diagram

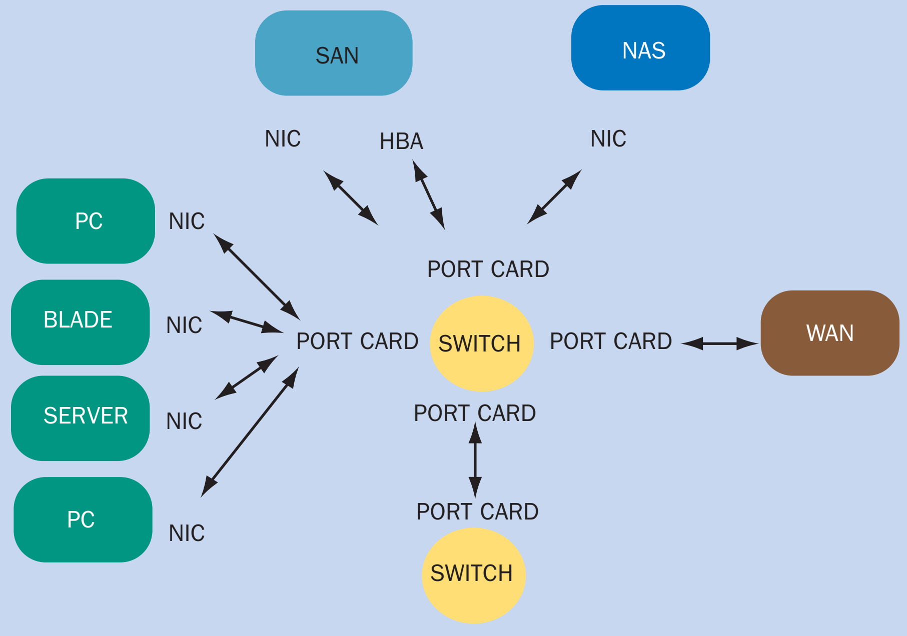
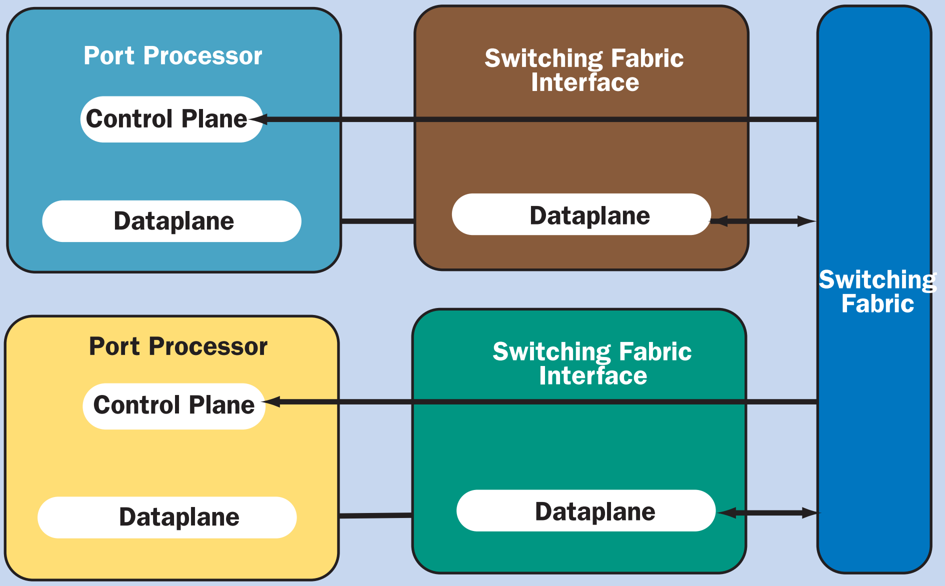


FIGURE 2 Switch Architecture



What About Line Cards?

To a large extent, the changes needed in enterprise switch/routers will require analogous changes in line cards found in PCs (where they are called NICs), SANs (where they are called HBAs) and WANs (where they are called line cards).

Line cards in PCs and servers operate as communications interfaces, in which information generated by an application-level program (such as an email program or Web browser) can be transmitted to/from another end device. When an application program wants to transmit to another machine, it contacts the operating system kernel in the origin machine (a similar process happens at the receiving-end PC). The OS microprocessor then handles the necessary functions, which include chopping up large files into packets with appropriate header information, and requesting positive acknowledgement from the receiving end device.

Since the microprocessor is multitasking a variety of functions in addition to message transmission, (i.e., running the application itself), there necessarily will be frequent interrupts, with the information stored in memory buffers, until the microprocessor is ready to perform the necessary operations. Since there may be delays in receiving positive acknowledgements from end devices, line cards typically have additional memory buffers for packets waiting to be sent over the network.

Different generations of PC and server line cards use different architectures for switching transmissions to/from different ports. Most personal computers today have Peripheral Component Interconnect (PCI), a parallel bus that can sustain peak rates of 133 Mbps (a bus is a set of parallel links that permit any device to broadcast to all other connected devices). PCI Extended, or PCI-X, is a backward-compatible parallel bus extension of PCI that can operate at 1 Gbps.

TABLE A Processor Utilization @ 10-Gbps Transmission Speeds (Pentium 4, 1 GHz Clock Rate)

Task	Processor Utilization
TCP/IP	12%
Memory Mapping	5%
Interrupts	23%
Copy & Checksum	29%
Buffering	17%
NIC Driver	6%
Available for Applications	8%
Total	100%

Source: Intel

The Problem With Current Designs

One problem with the current PC and server line card topology is that at high speeds, multitasking PC microprocessors get swamped handling operating system communications tasks, and have little to no time left to execute application software. Intel provides the data in Table A on PC processor utilization when the processor is interfacing to a 10-Gbps link, showing that 92 percent of the processor time is consumed with communication functions, with only 8 percent available for running applications. As a result, the processor can run applications or handle communications, but not usually both.

A second problem is that the current PCI bus architecture only supports up to four 1-Gbps ports (depending upon the clock rate and bus width, it could do 5 Gbps aggregate), or substantially less than 10 Gbps, because the PCI bus is completely utilized at that point and cannot sustain higher rates. If we want to fully utilize 10-Gbps switches, we need 10-Gbps line cards for the network endpoints.

The 10-Gbps line card solutions also need to become much less expensive. Right now, specialized 10-Gbps NICs, for high-end blade servers like the Small Tree or S2IO sell for thousands of dollars. To gain market traction, they need to sell for less than $500.

Next -Gen Line Card Solutions

An obvious solution to the overtaxed microprocessor problem is to add coprocessors to

Current Design Limitations

The basic problem with current port processors, switching interfaces and switch fabrics is that they were not designed to sustain the additional functional requirements and the much higher speeds that will be needed in next generation switch/routers.

For example, today's port processors would simply run out of cycles if asked to perform tomorrow's functions—such as additional lookup functions for quality of service (QOS), access control lists (ACL), MPLS label swapping, XML tags, load balancing, etc.—all executed at wire speed with minimal input queuing and sub-microsecond packet latencies.

In addition, next-generation control planes will need to incorporate a non-trivial amount of new software that current switches don't support. For relieve the pressure by handling more of the transmission work. Current line cards have coprocessors, but very few operate at 10 Gbps, they only handle a small set of simple functions, and they cannot provide sub-microsecond delays. As the required level of line card functionality increases (for XML, SIP, etc.), these offerings will be unable to meet the necessary wirespeed and delay requirements.

This is starting to happen, facilitated by improvements in PC operating systems that will support offload coprocessors. As one example, Microsoft has been working with a number of NIC and HBA vendors on hardware assistance (on the processors and line cards) that will work best with server and PC system software. It plans to provide TCP/IP kernel support in the forthcoming Longhorn release of Windows to allow hardware to assist in NIC (or motherboard) buffer management and processor offload for device drivers (see http://www.microsoft.com/whdc/device/network/tcp_chimney.mspx).

A related solution is the Remote Direct Memory Access (RDMA) standards initiative, which extends the three-decades-old DMA (direct memory access) to *all* devices attached to a network, without taking up processor time. From a networking perspective, DMA's utility has been limited because only devices physically located on the same backplane can use it. RDMA offloads interrupt handling from one or more processors, allowing bits to pass directly to and from processors and memories. The result is that one can:

- Offload "context switching" by processors at either end between user and kernel address space (i.e., switching between application and operating system programs, which involves suspending the context of one program, storing all of this in temporary memory called registers, and switching to another program)
- Handle TCP/IP address space mapping to server address space; and
- All this can be done without processor intervention at either end (see the RDMA website, http://www.rdmaconsortium.org for more information).

One other change that will be necessary for all future line cards—not just server and PC NICs, but also for SAN HBAs and the port/line cards on enterprise and WAN switch/routers—will be the movement from parallel bus architectures to switched serial architectures. The parallel bus architectures that have been used for decades will only support four to five 1-Gbps ports. To support multiple 10-Gbps ports we will need serial transmission, which can scale to higher clock rates without the skewing and synchronization problems seen in high-speed parallel buses.

Serial transmission does have an additional complication of needing to detect a clock from the bit stream and staying synchronized to it, but this can be handled with 30-year-old techniques, and will continue to scale with higher clock rates. Eventually, virtually all integrated circuits will use serial high-speed interfaces for communications activities.

PC manufacturers are moving toward a new generation of line card interconnects to replace the 10- year- old PCI bus, with the adoption of PCI Express (also called 3GIO) for third generation input/output). PCI Express is a serial full-duplex connection that can sustain 2.5 Gbps in a single lane, and 200 Gbps in aggregate. Advanced Switching Function (ASF), sponsored by the PCI Special Interest Group (www.pcisig.com) adds a transaction layer on top of PCI Express to create a high-performance switching fabric interface on the NIC. This permits wirespeed throughput with sub-microsecond delays and could be used on line cards for 1-Gbps and 10-Gbps LANs; (Gigabit and 10 Gigabit Ethernet) SANs (FCS at 1-Gbps, 2-Gbps, 4-Gbps and iSCSI at 1-Gbps/10-Gbps); and WANs (SONET/SDH and ATM at 622 Mbps, 2.5-Gbps, 10-Gbps).

In summary, line cards are going to go through architectural changes that will be at least as far-reaching as those that will be taking place on the switch side. The result will be substantially increasing functionality for enterprise users□

Today's switches and routers assume only 20 percent of packets need control plane processing

example, if you want to do native switching of multiple protocols, or convert one protocol into another, or encapsulate one protocol inside another, you are going to need additional processor capacity and software—all the more if you also want your switch to be programmable or (better yet) programmable on the fly.

As previously noted, Cisco, Foundry and Juniper have moved from control plane designs assuming 20 percent exception processing, to designs assuming 100 percent exception processing. Unfortunately, this has been done via frozen-in-silicon designs involving half a dozen large ASICs on large circuit boards with limited programmability. These kinds of designs can handle fast, basic routing, plus standard applications like DiffServ and MPLS, but they are not very programmable and they are incapable of handling

Offloading tasks to co-processors, like TOEs, is a step in the right direction

future demands for extreme heterogeneity (multiple packet types and protocols).

Even if you can re-program current switches, this is very difficult to do using current technology. If, for example, you want to modify a Cisco router, you need to understand Cisco's IOS software and then write custom scripts for your particular application need. Besides being extremely time consuming, the result typically won't be fault-tested under a variety of conditions—so as a result, won't be carrier-grade. While Cisco's new modular IOS will make greater degrees of programmability possible, the issues of ease-of-programming and carrier-grade fault tolerance remain.

Switching fabrics and switching interfaces will face similar speed and complexity challenges—particularly if your LANs are handling time-sensitive traffic like voice that cannot tolerate dropped packets.

Here, the basic problem is that, as we operate at higher speeds, a single switch fabric that isn't non-blocking becomes a bottleneck. A converged switch—one that can switch LAN, SAN and WAN traffic—compounds the bottleneck problem. This is due to:

- Higher speeds for each stream.
- Each stream having different processing requirements that might take widely different amounts of time.
- The need to rationalize the different packet lengths.
- Overall higher throughput that demands sub-microsecond latencies.
- Intrinsically higher processing time in total.

Without a drastic redesign of switching fabrics to make them non-blocking, the result will be unacceptable throughputs, cross-switch delays and increased latency.

Possible Switch Solutions: Offload And Buffer

One logical inference from the above discussion is that just as networking engineers originally created control planes to handle exception processing, we are going to need a further offloading of functionality away from the control plane to handle the increasing flexibility needed by next generation switches. This isn't a new idea—what's new is doing it for many more functions and doing it at 10 Gbps with sub-microsecond delays.

An example of this, touted by Intel and others, is the TCP offload engine (TOE), in which a separate coprocessor handles the processor interrupts, checksum calculations, and buffer management associated with TCP/IP. We believe offloading concepts like TOE are going to be a necessary part of next-generation switch/router designs.

With offloading, switch/routers will be following the lead of the personal computer. For years, PCs have had standard physical packages for motherboards, for add-on cards, and for interfaces to the myriad different hardware and software components. This has led to high volumes, high quality and low prices.

In the networking space, a similar modular approach is being advocated by the Advanced Telecom Computing Architecture (ATCA) standards group. Virtually all chip, component and system makers support ATCA. Its approach will permit hundreds of vendors to build cards and boards for switch/routers and allow these cards and boards to be used in multiple applications and in multiple vendor product lines. This will improve product reliability and lower cost—thanks to greater economies of scale in manufacturing and less time spent on details standardized by

Decoding The Acronyms

Acronyms	Definition
ASF	Advanced Switching Function
ATCA	Advanced Telecom Computing Architecture
AMC	Advanced Mezzanine Card
DAS	Direct Attached Storage
DiffServ	Differentiated Services
DMA	Direct Memory Access
FCS	Fibre Channel System
GFP	Generic Framing Protocol
HBA	Host Bus Adapter
IP	Internet Protocol
iSCSI	Small Computer System Interface over IP
LAN	Local Area Network
MPLS	Multiprotocol Label Switching
NAS	Network Attached Storage
NGN	Next Generation Networks
NPU	Network Processor Unit
PCI	Personal Computer Interconnect
RDMA	Remote Direct Memory Access
SAN	Storage Area Network
TCP	Transport Control Protocol
TOE	Transport Control Protocol (TCP) Offload Engine
VOIP	Voice Over Internet Protocol
WAN	Wide Area Network
XML	Extensible Markup Language

ATCA (e.g., power, cooling, mechanical spacing and connectors issues).

One of the most important ATCA specifications is serial transmission at up to 10 Gbps per line, so a single ATCA card might have four or eight 10-Gbps ports on it. Other ATCA specs call for:

- Scalable shelf capacity to 2.5Tbps.
- Scalable system availability to 99.999 percent.
- Multiprotocol interfaces up to 40 Gbps.
- Robust electric power infrastructure and large cooling capacity.
- High levels of modularity and configurability.
- Convergence of telecom access, network core, optical network and datacenter functions.
- High security and regulatory conformance.

Next-gen switch port processors also will require larger memory buffers, particularly if these switch port processors will be handling large amounts of XML traffic. (There will be a similar need for increased buffering within NIC cards at each PC.)

XML is becoming the programming language of choice for the Internet, because of its ease of use and ability to handle rich content (voice, data, graphics, video). As users increase their file download and file streaming activity, the amount of XML control information is expected to mushroom. Moreover, anecdotal observations have shown that XML programs and streams are 10X-20X the size (measured in bytes) of comparable files written in Java, HTML, C, C++, or C# languages. Finally, XML tags need to be decoded by switch port processors; this takes more time than other types of processing, so port processors (and NICs) will need to buffer these streams.

Needed: Big, Non-Blocking, Programmable Switches

It's all well and good to offload, componentize and buffer to support heterogeneous traffic flows through the port processors, but next-generation switch fabrics and switch interfaces will also face additional demands. An enterprise switch/router that can handle a large number of 10-Gbps PCs, plus 10-Gbps SAN links and/or 10-Gbps WAN links, will necessarily require a (non-blocking) switching fabric much larger than an Internet core router needs for its substantially fewer high-capacity ports.

This suggests new approaches to scaling switching fabrics, such as:

a.) Adopting Clos or Batcher-Banyan non-blocking switch approaches (in which careful interactions between switching interface control plane and switching fabric data plane eliminate the possibility of contention); and/or

b.) Creating switching fabrics that have more than one stage. Although these approaches may increase the control plane processor time spent in specifying all the paths through the switch fabric, they would offer the potential for greater concurrency (i.e., the ability to handle more simultaneous transfers from input ports to output ports).

Another important element of next-generation switches is that they will need to be programmable—or even better, programmable-on-the-fly (particularly to defend against new types of security attacks). Of course. the results also will need to be carrier-grade.

One interesting approach that will provide both programming and programming on the fly is a DARPA-funded project begun in the late 1990s that is now called *active networks*. Active networks research developed an encapsulation protocol that could be wrapped around objects such as files, and these files could be transported across a network, and executed by enabled machines in network nodes. Since these programs would be executed entirely in multiple control planes and not dataplanes, they would not require reboots; they would be truly "on-the-fly programmable."

This flexibility has yet to be exploited in commercial products, but in 2002–2004, the Internet Engineering Task Force (IETF) captured the active network concepts in a set of standards called Forwarding and Control Element Separation (ForCES). ForCES describes how a switch can be programmed on the fly, allowing each packet to be handled differently, according to software that is securely downloaded over the network to the appropriate node. While ForCES brings with it new administrative and security challenges, it also offers the potential for a new approach that might be effective in handling spam and other forms of security attacks. If the libraries of standardized solutions are fully fault-tested, the results will be carrier-grade.

Conclusion

Developing the products described here will be difficult. It will only be possible with cost-effective state-of-the-art integrated circuitry supporting communications and operating systems tasks.

One obvious problem will be to achieve speed, robust scalability and flexibility simultaneously, for end-to-end applications. In the past, network processors (NPUs) touted as high-speed have achieved much lower throughput and higher delays than advertised when their programmability was actually used in the real world. And these NPUs have not been scalable to the millions of flows needed for voice, data and video world-class networking. Moreover, the level of flexibility needed for next generation boxes will be substantially higher with XML usage for Web applications mushrooming, and content-aware routing becoming mandatory.

Another difficult problem will be to execute the new designs affordably. Extremely robust control planes potentially could be much more expensive than control planes designed for single point-solution boxes. Dropping the price per port to well below $1,000 also will be critical for gen-

Virtually all the chip, component and system makers support the ATCA specs

erating market traction. Next–generation integrated circuitry built with 90-nanometer photolithography permits 500 million gates on a single chip: this is more than adequate to provide cost effective single-chip solutions on PC motherboards and commodity switches for workgroups.

Solving these and other problems will require a fundamental integration of:

Network thinking, oriented toward sending a stream of packets through a pipe without interruption but with limited flexibility; and

Computer thinking, oriented towards maximum flexibility but with lots of interrupts that come at the expense of smooth pipelined dataflows.

To achieve the necessary integration, network architects will need much more sophisticated designs using the best elements of each approach. We are not talking about adaptations of current architectures that can manage some types of flows differentially, but 100 percent packet-by-packet routing at 10 Gbps.

We think that successful next generation products will require fundamental redesigns, involving such things as offload engines, remote direct memory access (RDMA), ATCA, ForCES, larger memory buffers and multilayer switching fabrics. They also could use new approaches that emerge via the IETF (Internet Engineering Task Force) approach of trying out many different concepts, with iteration after iteration occurring rapidly around the globe, and with the best of these concepts being proven and codified in standards.

Existing players who want to participate will have to be willing to scrap their legacy design approaches and start with a clean sheet of paper. Startups, of course, don't have that problem, and we're already hearing about startups working on these approaches (both chip and network equipment companies). Net-net, we wouldn't be surprised to see ATCA-based, ForCES-compliant switching equipment (and equivalent next–generation line cards), at affordable prices, sooner rather than later. Stay tuned!□

Companies Mentioned In This Article

Cisco (www.cisco.com)
Foundry (www.foundrynet.com)
Juniper (www.juniper.net)
Microsoft (www.microsoft.com)