



Intel[®] Technology Journal

Communications Processing

Enterprise Edge Convergence: Packet Processing and Computing Silicon in the Data Center

Enterprise Edge Convergence: Packet Processing and Computing Silicon in the Data Center

Matthew Adiletta, Intel Communications Group, Intel Corporation
John Beck, Intel Communications Group, Intel Corporation
Doug Carrigan, Intel Communications Group, Intel Corporation
Mark Rosenbluth, Intel Communications Group, Intel Corporation
Bill Tiso, Intel Communications Group, Intel Corporation
Frank Hady, Technology and Manufacturing Group, Intel Corporation

Index words: enterprise, edge, communication processors, network processors, application processors, architecture, microprocessors, bladed servers, servers, load balancers, Layer 7, SSL, firewalls, security, storage, TCP, TOE, Xeon, IXP2xxx, virtual, convergence, media servers, VPN

ABSTRACT

This paper describes the need for enhancing data availability and integrity and reducing the cost of ownership of computing in the enterprise. The “front-edge” and “back-edge” are discussed and the ideal mid-sized enterprise highlighted. We introduce the usefulness of a packet-processing capability in the enterprise and of an architectural approach that converges the complementary capabilities of IXP network processors and the Intel Architecture (IA) in a coherent system.

INTRODUCTION

Enterprise Information Technology (IT) managers are seeking improved data availability and integrity and reduced cost of ownership. Data availability refers to the need for compute resources and network resources to be useable and functioning at all times. Data integrity addresses the issue of consumers of data having confidence that the data they are using has been unaltered by third parties. Cost of ownership is the inherent expense of operating a given enterprise system.

The *Enterprise Edge* represents the gateway between clients and servers—the point where decisions must be made as to who gets access to critical data center resources and under what conditions this access is given. This is the ideal point to implement management schemes that will determine how cost-effectively and how securely the enterprise IT infrastructure will operate. This infrastructure and the management policies

implemented here will dictate the productivity of virtually all workflow processes in the enterprise.

Reducing the cost of ownership is based on the belief in the inherent value of converging multiple functions into a single system. Full realization of the purported benefits of convergence will require that the capabilities that exist in individual platforms today (as measured by performance, functionality, and cost) be preserved during the transition to a single unified platform. The integration must also deliver a number of significant new benefits:

- Fewer individual systems must be deployed, requiring less overhead, floor space, power, and cooling, etc. There is also a direct correlation between the number of boxes that must be managed and the size of the required IT staff.
- Fewer different types of boxes are required. A single converged platform that delivers the functionality of multiple individual platforms should result in lower training costs, fewer IT “specialists” and should be easier to manage, maintain, etc.
- A more homogeneous infrastructure will be in place. Many of the difficulties and sources of error in network installation and management are a result of the interdependencies that exist between different systems deployed together.
- There needs to be a unified management domain. The key to realizing operational efficiencies, the ultimate goal, is to provide a single, unified view of

the network and compute environment via the management system.

In examining the enterprise system, there are components that may be optimized. The current definition of the enterprise generally has a network connectivity, going to a farm of web servers, connected through firewalls and load balancers to application servers. The application servers are then attached to file servers that provide requested data to the compute servers. The optimizations that can be created for the enterprise networking infrastructure can generally be categorized as front-edge and back-edge services.

The front-edge generally includes the connection to the Internet, and the subsequent web servers for external users, and Virtual Private Network (VPN) tunnels and load balancers to the Local Area Network (LAN) within the enterprise. The LAN includes application servers, mail servers, printers, etc. The LAN also communicates to the enterprise back-edge.

The back-edge generally includes the path from the LAN to storage. Today this path is generally to Direct Access Storage (DAS). Tomorrow it is expected that Network Attached Storage (NAS) with its virtualization capability will be widely deployed. Tomorrow's back-edge will also enable new capabilities that will enrich the enterprise. Specifically, it will enable high-performance Internet Protocol (IP) media services to the enterprise telephony infrastructure that provide voice, speech, and multimedia to augment and, in the future, replace current PBX solutions.

The eventual conclusion of this evolution is a network and computing infrastructure that is fully autonomic. This infrastructure will be one that can diagnose its own problems and respond with little or no operator intervention; one that automatically senses shifts in demand (or status of available resources) and adjusts its behavior to provide the optimal response; one that identifies critical resources and accelerates access to them; and one that is easy to install, provision, repair, upgrade, and afford. This vision will be built on extremely high-performance computing and packet-processing platforms, whose foundations are the silicon represented by converged Intel Architecture-Intel Exchange Architecture (IA-IXA) silicon.

FRONT-EDGE EVOLUTION STARTING TODAY

The enterprise edge represents the boundary between the pure computing world of the application and database servers found in larger enterprise and service provider data centers and the broad universe of internet or intranet

clients that they serve. As shown in [Figure 1](#), the enterprise edge represents the gateway between clients and servers—the point where decisions must be made as to who gets access to critical data center resources and under what conditions this access is given.

This is the ideal point to implement management schemes that determine how cost-effectively and how securely the enterprise IT infrastructure will operate. The infrastructure and the management policies implemented here dictate the productivity of workflow processes within the enterprise. Whether to deliver enhanced manageability or enhanced security, the edge of the enterprise is the focus for much of the innovation in network equipment design. It is at the enterprise edge where much of the complexity of network management exists and a wide variety of platform types are deployed that provide an optimal point to deliver value-added services.

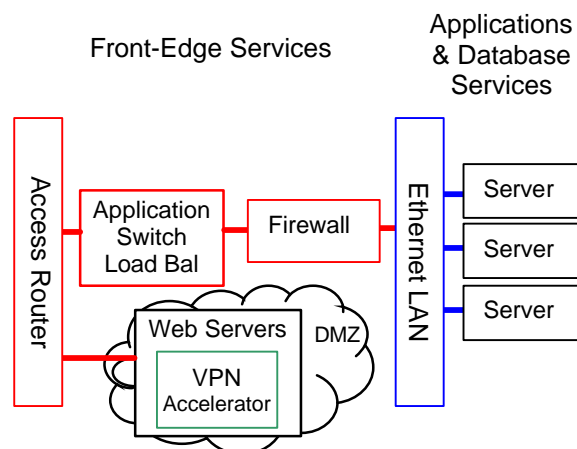


Figure 1: Today's front-edge to LAN and server

Today's Edge Configuration

As the gateway to the enterprise, the enterprise edge is the point at which customers, employees, business partners, and potential attackers first gain access to the information assets of the enterprise. Primary concerns of IT managers are to provide high bandwidth and reliable access to this information while protecting these assets from theft, inadvertent disclosure, and abuse. This is accomplished by carefully managing the traffic that traverses the edge. These gateways serve a number of functions: authenticating users who wish to access the network, protecting (encrypting) data in transit, and defining and enforcing service-level agreements with customers wishing to access the network. An additional goal of these gateways is to manage all of the various

access requests in such a way as to optimize utilization of the network and compute resources. Figure 1 is representative of typical current mid-range enterprise edge configurations.

The access router provides connectivity to the Internet, with the ability to manage on the order of 100,000 connections or routes. These routes are typically dynamically managed through route tables via either vector (route hops to destination) or link (connection state to local routers) protocols. Routing tables are dynamically updated and communicated to other routers as network links are added, removed, or become unavailable. A de facto standard protocol is Border Gateway Protocol (BGP), a vector-based protocol.

Multiple configurations of the application switch and firewall are possible depending on the relative needs for performance, security, and load balancing within the edge and web or application servers. The application switch and firewall perform a variety of functions, which are off-loaded from the web and application servers. The firewall supports services such as the following:

- *VPN/IPSec Gateway.* This requires line rate termination of IPSec protocols including full packet encryption, decryption, and authentication using 3DES, AES, SHA1, MD5, and RC4 algorithms. The Virtual Private Network (VPN) then provides the tunnel to the enterprise Local Area Network (LAN).
- *Filtering.* This characterizes Internet Protocol (IP) traffic to identify streams that have permission to access various information, and it can define access to demilitarized zones (DMZ) or intranet resources. Filtering can be done based on mechanisms such as L3-based Access Control Lists (ACLs) or L4-based TCP port blocking.
- *Network Address Translation (NAT)*

The application switch supports services including the following:

- *TCP termination.* Web servers are deployed in environments where thousands to millions of simultaneous clients submit HTTP requests via independent short-lived TCP connections. These connections must be set up, serviced, and torn down at very high connection rates and must support an aggregate throughput rate in excess of 1 Gigabit/second (eventually 10 Gigabits/second).
- *Intrusion/virus detection.* This requires the ability to analyze entire streams of network data traffic for arbitrary and constantly expanding sets of attack or virus signatures. The data stream is most often

segmented across many discontinuous packets. These signatures must be compared against a database of potentially tens of thousands of active rules. This also requires the ability to track protocol state machines for each active connection, looking for improper protocol usage.

- *Traffic manager/Layer 4-7 Load Balancer.* This requires the reassembly of Ethernet packets into TCP streams and subsequent classification of packets using Layer 4 through Layer 7 parameters, including, but not limited to, TCP port numbers, URLs, cookies, user names, application type, etc. Subsequent forwarding/drop priorities are based on these parameters.

Web servers are often established in a DMZ, such that external initiated Internet access requests are able to access data contained within the DMZ, but do not have visibility or access to the internal network or hosts. Both enterprise hosts and appropriate external agents have access to information and servers contained within the DMZ; however, protection protocols prevent external access to internal servers inside the firewall. Servers, such as web or mail servers, are considered to be in a semi-trusted DMZ area, based on optional security or firewall protection present before DMZ access and the level of DMZ isolation to the internal network. Servers contained in the DMZ may also have VPN termination capability.

Traffic that passes through the application switch and firewall now has access to the intranet or enterprise LAN typically via an Ethernet switch, to be serviced by a variety of enterprise data and application servers. HTTP streams are directed to the appropriate web servers. HTTP streams are SSL terminated either via an SSL appliance or capability within the web server. SSL acceleration requires very highly sustained SSL record processing, including public/private key processing using up to 2048-bit keys. Communication intended for other enterprise clients or servers is also routed via the Ethernet switch.

TOMORROW'S EDGE CONFIGURATION

The access router, application switch, and firewall capabilities outlined in the previous section are supported via server-based software or accelerated via stand-alone fixed function appliance products installed at the edge of the network behind the edge router. As outlined, edge functional requirements have spawned a number of different solutions ranging from IPSec and SSL-based VPN gateways through firewalls, NAT proxies, virus scanners and intrusion-detection systems. Because the

nature of the problem addressed by these systems often requires very specific processing on all data that traverses the gateway (such as encryption and virus scanning), these applications are very compute intensive. Historically, developers have had to trade off performance (line rate) for functionality when designing these systems, particularly with software-only implementations. Alternatively, non-extendable platform-specific optimizations have been done to get the required level of performance at the expense of time-to-market and solution scalability.

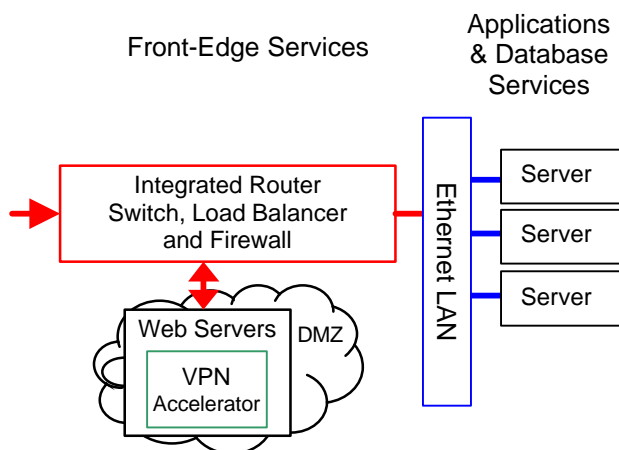


Figure 2: Tomorrow's front-edge configuration

The configuration of future enterprise edge solutions shown in [Figure 2](#) highlights three important enhancements to address the constraints presented by today's edge solutions:

- Multi-function converged hardware acceleration is available to support the application switch, VPN, firewall, load balancing, routing, and TCP termination to enable a full-featured enterprise edge.
- Merged general-purpose (i.e., architecture based on the Intel[®] Xeon[™] processor) compute capability with tightly coupled packet processing (i.e., Intel IXP architecture).
- Reconfigurability and flexibility to support evolving features, standards, and protocols via software programmability.

The consolidation of various appliance and “add-on” boxes into a single device provides inherent advantages

[®] Intel Xeon is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

to cost, power, footprint area, and on-going IT support. The merged edge solution will provide a platform upon which the various capabilities can be built in a clean consistent way with the ability to separate the “general-purpose” compute functions from “packet processing” compute cycles—with each operation allocated to the appropriate resource.

This configuration not only offloads edge software solutions from the enterprise web and application servers, but provides the necessary processing power to support full-featured capabilities at line rates up to and beyond 10 Gb/sec. Programmable general-purpose and packet processors provide the capability to adapt to support additional product differentiation via customer-specific value-added services, and expand product capability to new features or standards via a common hardware platform. This will allow for quicker system design and validation and the reuse of existing application code. It will also result in a much more scalable architecture that can easily take advantage of advances in CPU and packet-processing capability from one silicon generation to the next.

The extended programmable processing represented can also enable an evolution towards a network and computing infrastructure that is fully autonomic i.e., one that can diagnose its own problems and respond with little to no operator intervention; one that automatically senses shifts in demand (or status of available resources) and adjusts its behavior to provide the optimal response; one that identifies critical resources and accelerates access to them; and one that is easy to install, provision, repair, upgrade, and afford. Autonomic features could include the following:

- Load-driven provisioning and balancing of services.
- Self-healing through automatic discovery and alerting of faults.
- Intrusion prevention whereby instead of just simply logging the fact that an attack has occurred, the enterprise is protected from the malicious entity.
- Protected (authenticated) management of the platform.

Expense Reduction with the Converged Edge

The focus of IT managers on reducing operating expenses is supported with the inherent value of converging multiple Enterprise Edge functions into a single platform. This hardware integration, in conjunction with continued advances in operating system and management tools, will deliver a number of significant benefits. As outlined in the introduction, this

converged solution will support reduced capital and operating expenses by having fewer numbers and types of unique boxes deployed, by requiring less IT operations support due to the reduced box count, and by a migration to a unified network and computer management system.

THE BACK-EDGE: A DRAMATIC SHIFT TO VIRTUALIZED, NETWORKED STORAGE

Over the last several years, the storage market has been undergoing a significant and rapid transition away from a Direct Attached Storage (DAS) model (storage installed inside a particular server) to a Networked Storage model. Networked Storage includes both Fibre Channel-based Storage Area Networks (SANs) and Ethernet-based Network Attached Storage (NAS) appliances (see [Figure 3](#)).

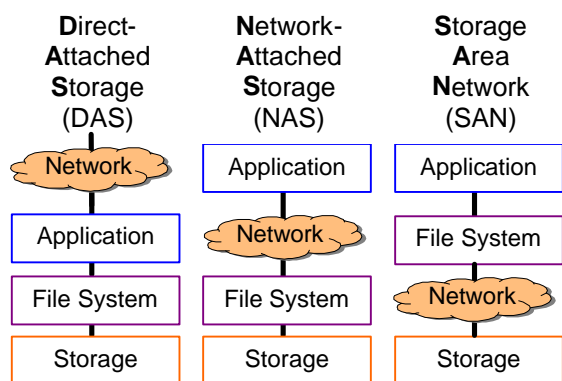


Figure 3: Simple view of DAS, NAS and SAN

The shift to networked storage has opened the possibility to view storage as an independent resource that can be managed and optimized more efficiently than can be done with DAS. The economics underlying the transition to networked storage are very compelling. When viewed from an IT manager’s perspective, the impact of a SAN implementation (vs. a DAS implementation) typically includes the following:

- 75% more storage capacity (GBytes) handled per IT manager.
- A 70% increase in disk utilization.
- A 50% reduction in costs for back-up hardware.
- Less data center floor space (due to increased density).
- Fewer general-purpose file servers.

While most network storage implementations today are Fibre Channel-based SANs, similar benefits can be

accrued by LAN (Ethernet)-attached NAS boxes. The costs associated with storage parallel those seen in the server world, with ongoing maintenance, backup and provisioning making up the majority of the total cost of ownership.

Network Storage Today

The transition from DAS to NAS has been underway for some time, though it is far from complete. The storage infrastructure will typically include the elements in [Figure 4](#)

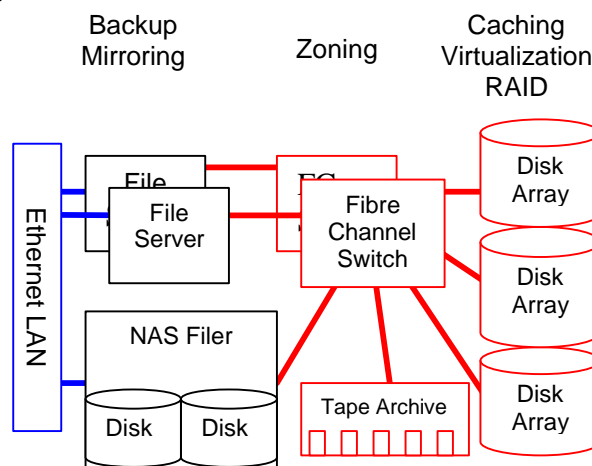


Figure 4: Today’s back-edge storage configuration

Fibre Channel Storage Area Networks

The SAN infrastructure is a totally separate network with all of the normal costs and complexities involved in any large network. A series of Fibre Channel switches provide the interconnect within this network, while the connections to the LAN are through either a dedicated file/database server or via a NAS appliance.

In this model, the actual Fibre Channel SAN is relatively “dumb,” providing network connectivity and some very basic services (zoning, etc.). The switches are roughly equivalent in functionality and complexity to managed Layer 2 Ethernet LAN switches that support Virtual LANS (VLANs). Fibre Channel directors (large, modular switching platforms) are found in large installations. These products provide larger port counts and greater reliability via redundancy, etc., but are still essentially “dumb” switches.

Fibre Channel SAN implementations span a range of performance levels with 1 Gb/sec and 2 Gb/sec links representing the majority of the installed base of switch ports. During 2003, the market has begun to transition to

4 Gb/s switch ports. This progression is expected to continue to include 10 Gb/sec Fibre Channel links in the 2005-2007 timeframe. 10 Gb/sec links will initially be relegated to aggregating traffic from multiple switches/directors or for switch to switch links, though native 10 Gb/sec server and array connections will begin to be deployed during this period.

Disk Arrays and Storage Subsystems

The actual data storage elements reside in the end-points of the Fibre Channel network: disk arrays and disk subsystems. The basic architecture of a storage subsystem includes one or more channel controllers (the Fibre Channel network interface), a number of disk controllers (providing the interface to a series of physical disks, typically via a Fibre Channel-arbitrated loop) and an array controller (a processor complex that manages all of the functions of the array and provides storage services).

The array and subsystem platforms provide many of the intelligent services delivered in the SAN including the following:

- *Redundant Array of Inexpensive Disks (RAID)*. To provide improved data integrity and immunity from disk failures.
- *Disk caching*. To provide improved response time for frequently accessed data.
- *Disk virtualization*. To allow “clients” to reference a logical disk where the actual data storage may exist anywhere on one or more physical disks, thus improving utilization.

The services provided by the disk subsystems represent only a fraction of the services demanded of an enterprise storage network. Many additional services must be supplied by the computing elements in the network (the general-purpose server platforms acting as file servers). File servers typically provide the following services:

- *File system services*. These implement the actual file system and maintain the mapping of files to actual blocks of data on a disk. Typical file system support would include CIFS, NFS, NTFS, etc.
- *SCSI processing*. This implements the SCSI protocol for accessing block data from the disks/subsystems.
- *Backup, mirroring, etc.* The file servers are often responsible for directly managing the data movement required to mirror databases and perform backups for both archiving data and remote backup that provides for disaster recovery.

- *Storage management and monitoring services.*

Network Storage Tomorrow

The back-edge of tomorrow will be based on platforms that deliver the same services deployed today in a much more consolidated, easy to manage system. This will include the ability to consolidate around a single data center network, seamlessly support both **block and file services** (using multiple file systems) with the same infrastructure and the ability to integrate cleanly with legacy file servers, Fibre Channel switches, and storage subsystems. These platforms will also serve as the deployment point for a range of services that are either totally new or are deployed at a different point in the network.

Network Attached Storage

NAS products (often referred to as “filers”) represent a step in the evolution toward a more consolidated back-edge of the enterprise data center. They provide the means to deliver many of the benefits of SAN-attached network storage without the necessity to install and maintain a separate Fibre Channel network.

The initial NAS products were storage appliances (often based on a standard PC/server platform) that could be attached to a standard LAN and act as a dedicated file server. NAS products represented a low-cost way to deploy networked storage, and they took advantage of the file system code running on the PC platform.

NAS appliances could include direct attached storage or could act as a front-end to a Fibre Channel SAN (referred to as a NAS head). In either case, the primary function of the NAS box is to implement the file system, taking in file requests and generating the appropriate SCSI commands to read/write the actual disk blocks. As the popularity of NAS products has grown, the trend has been to develop more hardware platforms specifically designed for NAS applications, thereby improving cost, performance, and scalability. The back-edge of tomorrow will extend the NAS concept to support multiple protocols and offer multiple services.

The initial deployment of NAS products in the enterprise provided incremental storage for a department or workgroup. The current usage model now includes NAS as a critical part of the total enterprise storage infrastructure. The overall trend is toward the convergence of both NAS and SAN deployments into a single, unified storage infrastructure.

Multi-Service Storage Platforms

The Multi-Service Storage Platform (MSSP) is a product concept for the next generation of storage infrastructure. The MSSP (see [Figure 5](#)) will allow an enterprise to extend the data center LAN to deploy an all IP-based storage network that offers the performance, reliability, and scalability that has come to be expected in a dedicated Fibre Channel SAN. The MSSP will be compatible with the legacy SAN infrastructure, allowing the IT manager to migrate to the IP infrastructure at a pace that makes sense for the business. The MSSP will also allow the deployment of multiple storage services at one central point in the network, thus reducing management complexity and expense and expanding the scope of a given service to include the entire infrastructure.

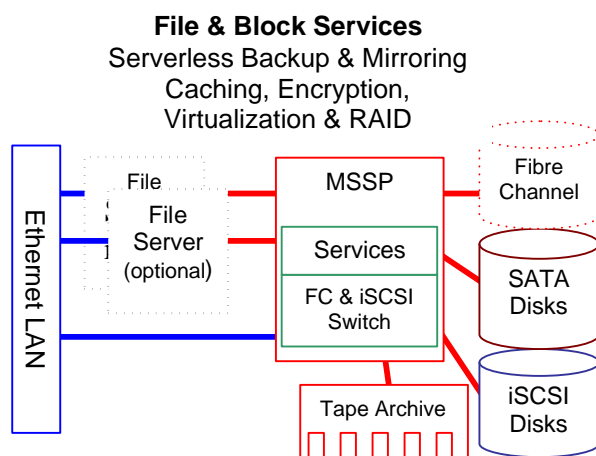


Figure 5: Tomorrow's Multi-Service Storage Platform

The MSSP supports the deployment of a number of new or improved protocols and services in the storage infrastructure. Some of the key services and protocols are discussed below.

iSCSI

Storage networks have traditionally been physically separate networks that provide logically different services: Ethernet LANs have hosted file-based storage traffic and Fibre Channel SANs have hosted block-based SCSI storage traffic. The iSCSI (IP SCSI) protocol allows the convergence of these logically and physically disparate networks. iSCSI provides the means of delivering SCSI commands (typically carried via Fibre Channel in a SAN) via an IP network.

The IP network may be an Ethernet LAN or any IP-based Wide Area Network/Metropolitan Area Network (WAN/MAN). iSCSI is seen as a way of extending the reach of SANs across large geographic distances (beyond the reach of Fibre Channel) to support remote storage.

iSCSI deployments are expected to be minimal over 1 Gb/s Ethernet (suffering from lower performance relative to 2 Gb/sec and 4 Gb/sec Fibre Channel links). As 10 Gb/sec Ethernet links are rolled out in the data center, iSCSI will likely begin to be viewed as a mainstream alternative to Fibre Channel.

The MSSP will support iSCSI at multi-Gigabit rates as a means of delivering block-based storage access via the IP network. This is key to enabling a converged LAN/SAN environment.

Virtualization

Virtualization of storage resources has been proven to have a strong positive impact on provisioning, availability, utilization, and maintenance (operating expenses associated with storage in the data center).

Virtualization encompasses a number of different technologies, applying to both block- and file-level virtualization. In either case, the concept is to decouple the logical representation of a piece of data (either block or file) from the physical instantiation of the data. This requires a mapping of the logical reference to one or many different physical structures (on the same or different storage subsystems).

The MSSP will support both block and file-level virtualization services at a central point in the network. This will allow the benefits of virtualization, transparently to the application, to span multiple hardware platforms from multiple vendors. In addition, management of the virtualization services will be focused on a single, central platform rather than across multiple disparate vendor platforms.

Disk Caching

Disk caching is a very powerful tool for improving the performance of storage subsystems. Caching is often closely coupled to the virtualization system (since it is the logical elements that an application cares about and will want to cache). Disk caching requires sophisticated algorithms to predict the data most often accessed and to store it in a local DRAM so it can be accessed without incurring the latency of reading from a physical disk. Disk caches also function as write buffers so that applications can "write and forget" without incurring disk-write latencies. Data integrity is a primary concern for disk caches. Vendors go to great lengths to ensure

that data in flight (cached or a buffered write) is immune to bit errors.

Both virtualization and disk caching are very compute-intensive applications that deal with very large active data sets (a disk cache may use 16-32 GBytes of DRAM for its cache).

Both caching and virtualization services in the MSSP will require very high compute performance and the ability to manage very large databases. For this reason, a large percentage of these applications are hosted on Intel Architecture microprocessors.

Secure Storage

Security technology is not widely deployed within storage networks. Historically, Fibre Channel SANs have benefited from their physical separation from the bulk of the enterprise (accessible only via dedicated file servers with access control capabilities). With the introduction of LAN-based technologies, such as NAS and iSCSI, the isolation of the Fibre Channel SAN is lost, particularly when IP networks are extended outside the enterprise. Recognition of this has led to the requirement for IPsec (the IP Security protocol) as part of the iSCSI protocol.

In addition to providing security for data in transit (e.g., when using iSCSI for Internet-based remote backup), a number of new environmental factors are driving a move to secure data at rest. The primary drivers for this move are based on an increasing need to protect private information—including legislative requirements.

The Health Information Portability and Accountability Act (HIPAA) prescribes that health care providers and insurers take steps to maintain confidentiality of patient records and that they retain records for extended periods. Many have interpreted this act to mean that medical records stored on disk or archived in electronic format must be encrypted to ensure confidentiality.

In addition to health care requirements, new standards of corporate accountability require that information regarding financial performance and reporting not only be retained but that the integrity of the data (freedom from tampering) be certifiable. This is most readily achieved by using encryption and authentication techniques (such as those used in IPsec).

In order to provide the level of security demanded, and not interfere with the expected performance and availability of the data storage encryption/decryption, processing must be handled on the fly (at full line rate). Further, the crypto processing must be compatible with the other required services, such as caching, virtualization, backup. This requires very high-

performance packet processing to complement the applications processing.

Application Servers: High-Performance Host Media Processing

Much of the focus of this paper has been on applications with a very high network bandwidth component. These offer clear examples of cases where the application of high-performance packet processing has a direct impact on the overall system performance. There are also applications, however, where the network bandwidth requirements are low that can still benefit from a coupling of the capabilities of the IXA and IA product families. Host-based Voice Band Media Processing (HMP) is such an application.

An IP media server is a solution using host media processing running on high-performance computing platforms using the Internet Protocol (IP) to provide media services to the enterprise telephony infrastructure. These services include voice, fax, speech, and audio conferencing services to augment and in the future replace PBX solutions, as a back-edge solution.

An IP media server could be used to deploy an Interactive Voice Response (IVR) system. IVR works like this. A media server answers a call, plays and streams a voice file, and detects digits (via a DTMF mechanism) for interactive selections. IVR is currently being deployed in such applications as financial services, (telebanking) and airline travel reservation call center call flow management.

Today's IP media servers operate on standard server platforms without acceleration hardware, supporting a maximum of ~400 IVR channels. Tomorrow's high-performance enterprise applications require 480 IVR channels, while telecommunication applications will require a minimum of 672 channel density. One approach to achieving these rates in tomorrow's IP media server is to increase supported channel density by offloading packet processing from the IA32 server to a network processor.

The technical challenge is in handling G.711 traffic with 10 msec frames, short RTP packets (80 byte payloads with 54 bytes of overhead) and 1 frame/packet. Each IP IVR channel has 200 packets per second to be processed (100 each direction). While the aggregate data rate is relatively low (<55 Mb/sec total for 500 channels), the overhead of managing a constant stream of small packets taxes even very high-performance server CPUs.

A number of scenarios were analyzed to determine the optimal partitioning of tasks between the IA processor and the network processor. The scenarios ranged from

no packet processing offload using standard 100BASE-T and 1 GigE NICs through solutions with RTP (real-time transport protocol) and IP media offload by using IXP technology on PCI as well as a solution similar to the TWIN Cities Experimental prototype where the IXP and IA communicate through a shared DRAM architecture (see Appendix A).

The real benefit of the IP Media offload is reducing the IA32 media processing footprint (MHz) per channel, permitting more channels to be serviced per IA32 CPU. The improved compute density can also result in reducing the solution's physical size and power requirements. The number of IP IVR channels supported per server will vary depending on the type of offload solution. A shared memory inter-processor communication mechanism, similar to that used in the Twin Cities Experimental Prototype, is expected to yield the best results.

For an application of an IP media server, a typical IA server platform would be used, which includes the following:

- Dual 3.2 GHz Xeon CPU-based platform, with hyper threading (4-way simultaneous multithreading)
- 512 MB of SDRAM (minimum)
- A suitable chip set such as Intel's E7501

With this system configuration, the number of supported IVR channels will range from a low of approximately 400 with no packet processing offload to an estimate of over 700 using a solution similar to the Twin Cities shared memory approach.

The greatest gains at the system level can be realized by tightly coupling the packet processor with the IA compute engine, thereby minimizing the amount of overhead necessary to communicate between the two systems.

In summary, when tomorrow's high-performance IP media server combines edge router functions with media processing, the benefits to both the IXP family of network processors and IA promises to impact system performance even when the actual network bandwidth needed is less than 100 Mb/sec.

The Common Threads—Silicon Requirements at the Converged Edge

When analyzing opportunities for convergence at the front and back-edge of the enterprise data center, a number of common themes become apparent.

The most obvious is that there is a strong and growing need for improved packet-processing capabilities in the data center. The demands placed on edge equipment at Gigabit Ethernet rates and beyond require specifically targeted packet-processing silicon solutions.

A second important theme is that the demand for high-performance application processors is virtually insatiable. This is particularly true in applications where content processing is required on entire network data streams. It is also clear that the mix of packet processing and application processing will vary across different applications—aligning well with a scalable building-block solution.

A more subtle observation is the opportunity for improving the performance and time-to-market of edge systems by more tightly coupling the packet and application-processing silicon.

Intel's current portfolio includes a range of products targeted at addressing the needs of the converged enterprise edge. A brief summary of some of the characteristics of the relevant products follows.

EDGE PLATFORM BUILDING BLOCKS

Intel Corporation produces both a high-performance application processor, the Intel Xeon processor and a high-performance packet processor, the IXP2850 network processor. These processors are well suited for edge applications. When used together they deliver formidable edge performance across the range of edge applications described in the previous sections.

This section describes the two processors and their suitability for various edge application tasks. Then, we describe platform-level communication between the two processors. Finally, possible future enhancements to this communication are explored.

Intel Xeon Processor in the Edge

Intel Xeon processors are commonly found in front-edge devices such as intrusion-detection systems, SSL accelerators, traffic managers, Virtual Private Network (VPN) gateways and in back-edge devices focused on providing high-performance storage. The Intel Xeon processor is compelling for these applications because it offers both a high-performance/low-cost platform and a high-productivity development environment.

The Intel Xeon processor features a high-performance IA core with a large cache (as of 10/7/03 a 3.2 GHz core

with a 1 MB cache¹). The platform surrounding the processor features low latency access to high-bandwidth memory (up to 4.3 GB/sec). Dual-processor configurations are available and prevalent in the edge. In these configurations, two Intel Xeon processors are presented with a shared, coherent view of main memory, enabling easy and rapid sharing of packet data and state. The platform allows for a high-throughput network connection through Network Interface Cards (NICs) placed in its Peripheral Component Interconnect (PCI) or PCI-X busses.

Edge application code can be developed with standard IA development tools such as the GNU C compiler (GCC)² and Visual C++³ already familiar to developers. Moreover, developers can often utilize open source code like Linux Netfilter⁴ Firewalls and OpenSSL⁵ to speed code development.

Platforms based on the Intel Xeon processor excel at complex processing over the large state required by edge applications. Traffic management is a good example. For these applications, IP packets must be combined into Layer 7 requests. Next, packet data are searched for strings that identify the server that can best service the request. For example, cookies within HTTP *Get* requests identify the server that has already performed a public key exchange with a client and so can most efficiently service the request. Intrusion Detection is more complex, requiring not only searching within a given packet stream, but also searching through state held on all flows to identify attempted multi-flow attacks on a site (such as port scanning). Systems based on the Intel Xeon processor offer an excellent platform on which to write, debug, and execute the large body of code needed to run these applications quickly.

¹ “Intel® Xeon™ Processor.” Intel Corporation. http://www.intel.com/products/server/processors/server/xeon/index.htm?iid=Homepage+STT_xeonproc_03ww41b&

² “Welcome to the GCC Home Page.” <http://gcc.gnu.org/>

³ “Microsoft Visual C++*.” Microsoft Corporation. <http://msdn.microsoft.com/visualc/>

⁴ “Netfilter, firewalling, NAT and Packet Mangling for Linux 2.4.” <http://www.netfilter.org/>

⁵ “Welcome to the OpenSSL Project.” <http://www.openssl.org/>.

IXP Network Processors in the Edge

IXP network processors are architected specifically for the task of packet processing. Traditionally, packet processing at high line rates has been done by special-purpose Application Specific Integrated Circuits (ASICs). The drawbacks to ASICs are well known: high development costs, long development cycles, and lack of flexibility (e.g., they are designed for a specific, single function and can not evolve to meet the need for new functions and features). There are two specific problems associated with packet processing on a general-purpose microprocessor.

Packets can arrive at a faster rate than a general-purpose processor can handle. For example, for Gigabit Ethernet, the frame arrival rate can be as many as 1.5 million frames per second (one frame every 672 nsec). For a 2 GHz processor, that equates to 1300 CPU instructions. For some edge applications this may not be enough for the required processing and packet IO. Also, cycles spent here subtract from the time available for application processing.

To compound the problem, the general-purpose processor often doesn't get to use all of the cycles for productive work. Memory latency causes lost processing time. Processing packets requires accessing and updating state stored in memory, for example information defining a given TCP connection to which a packet belongs. Main memory has a long latency in terms of processor cycles (for example a 100 nsec DRAM read is equivalent to 200 cycles on a 2 GHz processor). The large number of flows in many edge applications drives an increase in cache miss rates as the cumulative state and packet data for all the active flows becomes too large for the CPU's cache. Moreover, a large number of small packets increase the miss rate by requiring many small network IO operations, from the CPU to the NIC or to main memory. Each cache miss results in many wasted CPU cycles. Increasing CPU clock rates exacerbate this problem.

Network processors are designed specifically to minimize the performance impact of these problems using the following techniques:

- *Multiple RISC processors.* Intel's IXP processors contain specialized RISC processors called microengines, which are tailored specifically for packet processing. microengines are designed for parallel processing via multiple instantiations in arrays. They have an instruction set optimized for packet-oriented processing and are much smaller in silicon area than a general-purpose microprocessor. For example, Intel's IXP2800 contains 16 such microengines, running at 1.4 GHz for a total peak

processing capability of 22.4 billion instructions per second.

- *Multiple threads per processor.* The issue of long memory latency applies to network processors just as it does to general-purpose microprocessors. One approach to mitigate the impact of memory latency is to process many packets in parallel, using multiple threads per processor (Intel's IXP microengine contains 8 threads). Each thread handles a different packet. While one or more threads waits for the return of a memory read, another thread processes its packet. Specialized hardware in the network processor enables the parallel threads to maintain packet ordering and synchronization.
- *Multiple memory types.* During packet processing, two types of information are accessed, packet payload information, and control information. Packet payload tends to be accessed in large chunks (e.g., 64 bytes) which maps well to the IXPs directly connected DRAM. Control information often consists of small data structures (e.g., 4 or 8 bytes). Accessing a 32 byte cache line to modify 4 bytes results in very poor memory bandwidth utilization. Therefore, the IXP network processors also connect directly to SRAM, which performs well for small accesses.

As explained earlier, edge applications include significant amounts of packet processing. IP routing and Network Address Translation (NAT) are at the core of most packet-processing applications. Edge applications are no exception. The architecture of the IXP allows it to efficiently route and NAT packets, even streams of exclusively minimum-sized packets. The IXP2850 includes hardware acceleration for bulk encryption/decryption required by IPsec and SSL VPNs. The IXP2800 is also capable of terminating TCP connections over 100s of thousands of flows at multi-gigabit rates with minimum-sized packets.

Network processors are designed specifically for the lower-level packet processing described here, and generally not for higher-level application processing. General-purpose CPUs also benefit more directly from 30 years of compiler, debugger, language, and algorithm development and use. While software development tools, such as compilers, are available, a network processor's hardware may not be as fully abstracted from the programmer as current general-purpose CPU hardware. Network processors, for example, may place limits on program size and may require additional low-level coding. Network processors enable high-performance IO

and hide memory latency delays, but also require some additional programmer management of resources.

The strengths of the IXP complement the strengths of the Intel Xeon processor. A platform that includes both the IXP processor for packet movement/packet processing and the Intel Xeon processor for higher-level packet processing/application processing promises to deliver leading performance for edge applications.

IA/IXP Communication—This Generation

The converged applications described in this paper require the NP and CPU to share payload data and state information. In today's generation of IXP and Intel Xeon processors, information is transferred via the PCI bus. Each processor has its own private memory space. Buffers for passing packet data are set aside in each of the memories. A packet is passed by first allocating a buffer, writing the packet data into the buffer, and then posting a pointer onto a message queue.

The message queue is often managed using a ring in memory. Rings (also known as circular queues) are often used to pass data or messages from a producer of packet data to a consumer. The ring provides for rate matching enabling the long-term matching of production and consumption rates over short-term mismatches. The ring itself is a block of memory that is accessible to both producer and consumer. (Possibly by a window into memory through the PCI bus.)

The producer maintains a tail pointer, which indicates where the next message should be put onto the ring. The producer writes the message into memory at the address indicated in the tail pointer and then increments the tail pointer. The consumer maintains a head pointer, which indicates where the next message should be removed from the ring. The consumer reads a message from memory at the address indicated in the head pointer, and then increments the head pointer. The producer and consumer periodically read the head and tail pointers (respectively) to ensure that the ring does not overflow or underflow.

The two processors need to pass not only packets but also state. State is generally held per flow and can be quite large for edge applications. Per flow state sizes of 1,000 bytes or greater are not uncommon. Since average HTTP *Get* requests are about 400 bytes in length, state traffic may easily overwhelm packet traffic. Current edge platforms can use the same circular queue described above for passing state. The following options are available:

- The required state can be passed along with the packet. Here the network processor is the owner of

the state. State is appended to packets as it is passed to the CPU. The NP must make sure that all the required state is included and labeled. The CPU must then interpret state in the message as necessary and pass all state modification messages back to the NP. The NP must decode and act upon these messages, updating its local state. Multiple copies of some state variables may be required to correctly handle multiple packets on the same flow.

- The NP and the CPU may each hold local copies of state. Here each processor may consult its own local state copy when processing a packet. State updates require both writes to memory to update local state and the creation and passing of a message to the other processor. The receiving processor decodes the message and performs the state update. Careful design is required to ensure the two state copies remain coherent.

Using the example methods described in this section, the Intel Xeon processor and the IXP processor can collaborate to deliver excellent edge application performance and features, better than either processor could deliver on its own.

IA/IXP Communication—Next Generation

As discussed above, efficient state sharing is essential to achieving high performance for edge applications. The state sharing schemes outlined above make use of the strong write performance of current NP to CPU connections. They explicitly minimize the number of reads required.

Current state-sharing techniques have a number of drawbacks. First they require processors to construct messages around the state to be shared. These messages identify the state and the operation required on the distant state. Constructing and sending these messages incurs both processor instruction and bus cycle overhead. Similar overheads are imposed on the receiving processor to unpack and process the state message. Current state-sharing techniques also add code complexity to edge applications. Developers are required to write and debug code to implement messaging schemes that ensure coherency between multiple copies of state.

Future generations of edge platforms could make sharing of state between the IXP and the Intel Xeon processor more efficient by providing shared memory like that provided on the dual Xeon processor platform. A portion of memory would be shared with coherency enforced by hardware. Both processors would be provided high-bandwidth and low-latency reads and writes to this portion of memory. Moreover, each processor would be able to perform atomic read-modify-write operations on

this memory. Such a shared memory would allow NP and CPU programs to simply read and write shared state, removing the platform performance overheads and programming overheads inherent in state sharing on current edge platforms. This shared state has been prototyped on Twin Cities (see Appendix A) where strong application-level performance advantages were measured.

Shared state would also enable more natural migration of functionality between the two processors. It is likely that new edge functionality will be placed first on the Intel Xeon processor. As this functionality matures and becomes important to a larger fraction of packets, it may be migrated to the IXP. If the process implementing the functionality communicates with other application processes through shared memory, then it may be migrated without altering those other processes. Processing is migrated from one processor to the other, but state and packet handling interfaces remain unchanged in the shared memory. Such a programming model promises to deliver time-to-market advantages for the introduction and acceleration of new edge functionality.

CONCLUSION

The enterprise edge is getting smarter and getting simpler. Intelligence deployed in the network is seen to deliver a number of important benefits, including improved security, better end-to-end (user-to-user) performance and reduced management complexity. Greater intelligence allows for the deployment of fewer, more converged systems at the edge.

There are tremendous benefits to be realized as a result of convergence at the edges of the enterprise data center. Convergence at the platform level is seen as a way to reduce capital and operating expenses by reducing the number and type of platforms that must be deployed and managed. In addition, these converged platforms are being designed to deliver a range of new services that will offer improved security, better reliability and manageability, and the flexibility to support new services via field upgrades.

Delivering these converged platforms will require a paradigm shift in the platform architecture and the underlying silicon. The ability to consume and process multi-gigabit network data streams while delivering rich application performance is beyond the capabilities of today's general-purpose CPU architectures alone. Intel has recognized this necessity and has developed a family of network processors optimized for managing the high-performance network streams. The network processor family is designed to complement the application-

processing capabilities of the embedded Intel Architecture CPUs.

Edge platforms that exploit the IXP and IA processor families offer the high-performance packet and application processing demanded by today's and tomorrow's converged edge platforms.

As the services deployed become more complex and the bandwidth of the network increases, more work is needed to keep pace. The converged platforms at the edge will increasingly require very tight coupling between the applications and packet processors. As more of an application is offloaded to a multi-processor or multi-threaded architecture to address throughput and latency, the degree of interaction (typically sharing of state) between the two domains will grow.

Intel's research has identified a number of architectural approaches that can be employed to allow the accelerated packet and application processing described above, while maintaining the ease of programming and the wealth of available tools and applications available for the Intel Architecture processors.

A system that combines the IA and IXP products represents the fusion of two different architectural approaches. Harnessing the best attributes of each in a single system helps deliver new levels of performance for existing applications while enabling new classes of application that were not previously possible.

Appendix A

Twin Cities: An Experimental IA and IXP Convergence Prototype

The Twin Cities prototype⁶ was constructed to investigate the viability and impact of a higher performance platform coupling between the Intel[®] Pentium[®] processor and the IXP. The prototype demonstrates that these two processors can be attached with a high-throughput, low-latency, cache-coherent bus and that such a connection can deliver higher application performance.

Twin Cities connects the IXP1240's SRAM and DRAM busses to the Pentium[®] III processor's CPU bus as shown

in [Figure A-1](#). A dual port SRAM (DPSRAM) connects directly to the IXP1240. The other side of the dual port SRAM connects to the CPU bus (a.k.a. Front Side Bus or FSB) through translation logic implemented in the Twin Cities Field-Programmable Gate Array (FPGA). Twin Cities employs a *copy* architecture. Writes of data to be shared between the processors go to both the IXP1240's SRAM (or DRAM) and the Pentium III processor's DRAM. The dual writes don't waste instruction cycles on either processor, since they are hardware managed. Reads of shared data always come from the processor's local memory.

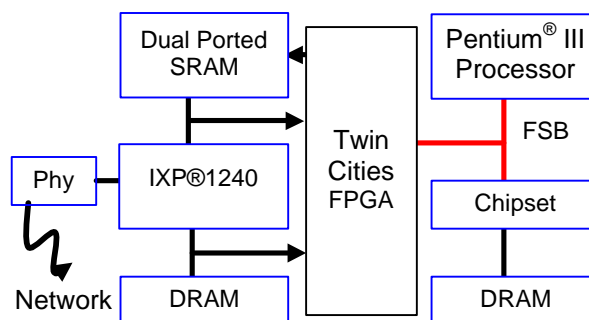


Figure A-1: Twin Cities architecture

[Figure A-2](#) is a picture of the Twin Cities system. The Twin Cities board plugs into the IXP board's SRAM and DRAM connectors through 5 Miktos connectors and into the motherboard through a Slot 1 card edge connector. The compact PCI backplane only provides power to the IXP card; the PCI protocol signals are left unused.

Portions of the Intel[®] IXA SDK 2.0⁷ Resource Manager were ported to Twin Cities to enable structured communication between the two processors and to enable reuse of existing IXA code. The IXA SDK 2.0 Network Address Translation (NAT) application runs on top of this infrastructure. For NAT, the microengines independently perform packet modification and movement for known flows. The core of a virus-scanning firewall application runs on top of NAT. Packets on identified flows are forwarded to the IA processor, which reassembles these packets, and searches them for 32 different virus strings using a Boyer-Moore

⁶ F. Hady, T. Bock, M. Cabot, J. Chu, J. Meinecke, K. Oliver, W. Talarek. "Platform Level Support for High Throughput Edge Applications: The Twin Cities Prototype." *IEEE Network*, July/August 2003, Vol. 17, No. 4.

[®] Pentium is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

[®] Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

⁷ Intel Corporation, "IXA Software Developers Kit 2.01 for IXP1200," <http://www.intel.com/design/network/products/npfamily/sdk2.htm>

string search algorithm. Packets containing viruses are dropped by the IXP, while clean packets are forwarded.

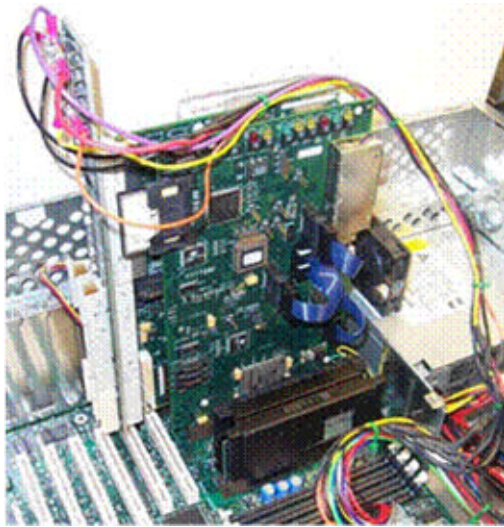


Figure A-2: Twin Cities prototype

Our low-level performance characterizations show that Twin Cities achieves four times the throughput and a quarter of the latency of an IO bus (PCI) connected platform. We measured CPU to NP and NP to CPU write performance of 266 MB/sec on Twin Cities. Karlin⁸ measured transfer rates for very similar IXP1200 and Pentium III processor systems connected over a PCI and found a maximum CPU to NP write throughput of 69 MB/sec, a 4x Twin Cities advantage. Latency was measured by passing control, implemented as a shared semaphore, from an IXP microengine to the Pentium III processor and back. Twin Cities has a round-trip synchronization time of 1.4 microseconds. This same measurement on a PCI connected platform shows a latency of 5.5 microseconds, 4x the latency of Twin Cities.

The low-level performance differences measured translate into application-level performance differences. NAT packet forwarding for already established connections is handled by the IXP1240's microengines, so throughputs didn't differ between Twin Cities and PCI connected platforms. Packets sent on new connections, however, require the packet to be sent to the Pentium III processor, which then sets up an entry in the NAT table.

We measured 45,000 new connections per second for Twin Cities and 23,000 new connections per second for our PCI-connected platform. For the virus scanning

firewall, all packets must be sent to the IA processor. Control traffic is sent in both directions. For this application, Twin Cities was able to maintain a throughput of 200 Mbits/sec while the PCI connected platform achieved 45 Mbits/sec.

ACKNOWLEDGMENTS

There are a few key individuals who have provided tremendous support and path finding. The authors thank Debra Towle, Myles Wilde, Tony Bock, Duane Galbi, Mason Cabot, and Jon Krueger.

REFERENCES

- [1] B. Carlson, "Intel Internet Exchange Architecture and Applications, A Practical Guide to IXP2XXX Network Processors," *Intel Press*, 2003.
- [2] Erik J. Johnson and Aaron R. Kunze, "IXP2400/2800 Programming, The Complete MicroEngine Coding Guide" *Intel Press*, 2003.
- [3] F. Hady, T. Bock, M. Cabot, J. Chu, J. Meinecke, K. Oliver, W. Talarek, "Platform Level Support for High Throughput Edge Applications: The Twin Cities Prototype," *IEEE Network*, July/August 2003, Vol. 17, No. 4.
- [4] Intel Corporation, "IXA Software Developers Kit 2.01 for the IXP1200," <http://www.intel.com/design/network/products/npfamily/sdk2.htm>
- [5] S. Karlin, L. Peterson, "VERA: An Extensible Router Architecture," *Computer Networks*, Volume 38, Issue 3, 2002.

AUTHORS' BIOGRAPHIES

Matthew Adiletta is an Intel Fellow and is responsible for Intel's network processor development. This responsibility includes evangelizing Chip-Level Multi-Processing (CMT) and hardware-based Light Weight Threading. He led the development of the IXP1200, the first fully programmable network processor for Intel. Adiletta graduated with Honors from the University of Connecticut in 1985 (B.S.E.E.) and currently holds 24 patents. His e-mail is matthew.adiletta at intel.com

John Beck is an engineering manager in Intel's Communication Infrastructure Group Technology Office. He joined Intel in 2000 and has managed the development of IXP1200 and IXP2800 families of network processors. Prior to Intel, he was Director of DSP design at Analog Devices Inc. and Engineering Manager in Digital Equipment Corp. Semiconductor Group. John has B.S.E.E. and M.S.E.E. degrees from

⁸ S. Karlin, L. Peterson, "VERA: An Extensible Router Architecture," *Computer Networks*, Volume 38, Issue 3, 2002.

Cornell University. His e-mail is john.c.beck at intel.com.

Douglas Carrigan is a strategic marketing manager in Intel's Communications Infrastructure Group Technology Office. He has held this position at Intel for the last five years and has led the marketing of the IXP family of network processors since the family's inception at Digital Semiconductor in 1996. Prior to joining Intel, Doug was responsible for marketing of DEC's StrongARM processors into embedded applications and led strategic marketing of MIP processors at IDT. Doug has over 20 years of experience in design, sales, and marketing of embedded microprocessors into communications applications. His e-mail is douglas.carrigan at intel.com.

Mark Rosenbluth is an architect in the Network Processor Division. He has been at Intel for five years and prior to that worked at Digital Equipment Corporation, where he was an architect for PCI Bridges. He also worked on VAX and Alpha microprocessors. He holds a B.S.E.E. degree from Rutgers University. His e-mail is mark.rosenbluth at intel.com.

Bill Tiso is the engineering manager at Intel's Network Building Block Division (NBD) responsible for Host Media Processing Technology planning and development in the NBD CTO Office. Mr. Tiso joined Dialogic Corporation in 1993 and managed the development of many products released as Dialogic and Gammalink. He holds a B.S.E.E degree from Florida Institute of Technology and a M.S.E.E. degree from Polytechnic Institute of New York. His e-mail is bill.tiso at intel.com.

Frank Hady is a principal engineer at Intel leading a small research group focused on providing high-performance network edge platforms. He also serves as the NPF benchmarking Task Group chair. Frank holds a Ph.D. degree from the University of Maryland. He has three patents (with nine pending) and has authored articles on networking platforms, network processor benchmarking, PCI usage, network interface design, and MPP network design and performance. His e-mail is frank.hady at intel.com.

Copyright © Intel Corporation 2003. This publication was downloaded from <http://developer.intel.com/>.

Legal notices at <http://developer.intel.com/sites/developer/tradmarx.htm>.

For further information visit:

developer.intel.com/technology/itj/index.htm