



# Intel<sup>®</sup> Technology Journal

Communications Processing

## Fabrics and Application Characteristics for AdvancedTCA\* Architectures

# Fabrics and Application Characteristics for AdvancedTCA\* Architectures

Brian Peebles, Intel Communications Group, Intel Corporation  
Chuck Narad, Intel Communications Group, Intel Corporation  
Victoria Genovker, Intel Communications Group, Intel Corporation  
Karel Rasovsky, Intel Communications Group, Intel Corporation  
Jay Gilbert, Intel Communications Group, Intel Corporation

Index words: AdvancedTCA, aggregation, AS, Advanced Switching, ATCA systems, interoperable building blocks, PICMG, SERDES, switching fabric, topologies

## ABSTRACT

After years of severe downturn in business activity, the communications industry is now embracing standards-based modular communications platforms (MCP) as a way to improve profitability while the market recovers. Eclipsing traditionally closed, proprietary communications systems, the adoption of MCPs promises Telecom Equipment Manufacturers (TEMs) lower development costs and a robust ecosystem of interoperable hardware and software building blocks, while at the same time making it easier to develop and run applications. It also promises Service Providers (SPs) smaller initial outlays and reduced Total Cost of Ownership (TCO). The MCP paradigm is rooted in several industry-wide standards initiatives and spans a broad ecosystem of building blocks including silicon, blades, chassis, backplane fabrics, operating systems, middleware, applications, and more.

The shift toward standardized hardware modularity began with Rack-Mounted Servers (RMS) that were optimized for compute-centric applications in enterprise networks and back-office systems. To accommodate the extreme requirements of central-office environments, however, a more flexible, reliable, modular, scalable, and higher performance approach was needed; thus, the concept of moving from proprietary systems to a standardized “bladed shelf” was born. Recently, the industry represented in the PCI Industrial Computer Manufacturers Group (PICMG) standardized a bladed

form factor in a series of specifications known as AdvancedTCA\* (ATCA).

Founded on the requirements of the communications infrastructure, PICMG 3.0 specifies the electro-mechanical, interconnect topology, and management aspects for building a modular, reconfigurable shelf. The large form factor of ATCA blades provides ample architectural headroom for next-generation silicon and applications. Switched backplanes, agnostic of fabric technologies (e.g., InfiniBand\*, PCI Express\*, Advanced Switching, StarFabric) facilitate extensive bandwidth scalability for a broad range of applications. The robust ATCA shelf management is based on the Intelligent Platform Management Interface (IPMI) 1.5 specification and it enables hot swapping, inventory information management, power distribution, and management facilities.

ATCA takes a holistic approach to High Availability (HA) and five 9s (99.999 %) reliability required in carrier-grade equipment through the integration of features for resiliency, redundancy, serviceability, and manageability. ATCA also lays a solid foundation for scalability of system performance and capacity. Application and control processors, network processing blades, Digital Signal Processing (DSP) blades, and storage blades can be hot-swapped on demand with profile-based interoperability, maintainability, and reliability. With its emphasis on low-cost manufacturing and the economies of merchant-market

---

\*Other brands and names are the property of their respective owners.

volumes, ATCA-based platforms will lower the cost of network elements.

Advanced Switching (AS) is a new standard from the Arapahoe Working Group, which builds on the physical and link layers of PCI Express and defines a scalable, protocol-agnostic fabric featuring traffic differentiation to support QoS, congestion management, and support for easy management and HA solutions. AS is currently at Release Candidate 1.0. An emerging ecosystem of interoperable components makes AS an attractive candidate for the backplane fabric in modular communications equipment.

In this paper, we explain how MCPs are designed around standard modular building blocks: silicon, hardware modules, fabric management systems, operating systems, and middleware. In particular, we focus on the AS fabric that is designed to work in this space. Standardization and broad adoption of MCP platforms are linked to reliable interoperability. By focusing the industry on a limited subset of the myriad of options offered by ATCA, a robust ecosystem of interoperable components can develop and persist.

## INTRODUCTION

As the boundary between enterprise and wide area networks begins to blur, a convergence of communications and computing platforms is underway. To facilitate this rapid evolution and the applications that are driving it requires not only significant computing power but a modular, scalable, and standardized approach to building hardware platforms that can support a broad spectrum of requirements for both compute and communications platforms. Born through intense industry cooperation, the recently ratified PCI Industrial Computer Manufacturers Group (PICMG) series of specifications also known as AdvancedTCA (ATCA) lays a foundation for building such a standard, carrier-grade infrastructure out of interoperable modular components.

One of the most critical aspects in enabling this technology is the ability for high-performance blades in communications and server chassis to communicate between and among each other moving vast quantities of data from blade to blade within a chassis (shelf). The switching fabric within a shelf enables the communication between blades and encompasses many switching technologies and numerous implementations, both standardized and proprietary.

We discuss the basic interconnect technologies available for ATCA systems including physical fabric interfaces and topologies as well as standard fabric technologies

ranging from Ethernet through a new emerging standard referred to as Advanced Switching (AS).

AS picks up where Ethernet leaves off, providing lossless transfer, congestion management capabilities and traffic differentiation for QoS, bandwidth scalability, and other features important for communications backplane fabrics.

At the physical level, the ATCA switching architecture relies on a passive backplane with two physical interfaces, the Base interface and the Fabric interface. The Base interface defined in the PICMG 3.0 specification is 10/100/1000BASE-T Ethernet. BASE-T Ethernet provides backward compatibility with a large installed base of products and is suitable for many lower-bandwidth applications. The Fabric interface defined in the PICMG 3.1 specification utilizes the "SERDES" (Serializer/Deserializer) interfaces that can support much faster speeds and do away with the magnetics required for Ethernet fabrics. This specification supports many different topologies, from Dual Star to Full Mesh, making it a very flexible architecture. In a Dual Star fabric, each blade has a pair of redundant fabric interfaces, one connected to each of the two redundant centralized switches. In a Full Mesh system, each blade has a point-to-point connection to every other blade, and each blade has a switch fabric component to connect the on-blade interconnect to the backplane ports. Redundant paths can be supported through these switches for failover, and a Full Mesh eliminates the need for dedicated switch slots.

The Base interface is meant as a transitional interconnect that is identical to the one used in the PICMG 2.16 standard, for vendors porting products from that standard to ATCA. The system developer can decide how and when to use the Base and Fabric interconnects. Either can be used as a unified control and data fabric or the Base interface may be used for the control plane fabric with dual star, and a Fabric interface based on AS, InfiniBand, or XAUI might be used for a high-performance data plane fabric.

We describe practical ATCA switching fabric implementations, focusing primarily on the most useful topologies and switching technologies ranging from Ethernet to AS. We also include an example discussion of an AS hardware and software model architecture and give examples of communications applications.

## ATCA SWITCH FABRIC AND CAPABILITIES

Fabric switching serves as the basis for interoperable blade-to-blade communications in a healthy ecosystem of interoperable building blocks used to develop systems that are based on the AdvancedTCA (ATCA) platform. When selecting the type of fabric switching to be utilized in a system, one must consider which protocol (e.g., Ethernet (10/100/1Gb), Fibre Channel,

InfiniBand, StarFabric, Advanced Switching) and which topology (e.g., Star, Mesh) best suits the needs of the application. Additional considerations include standards-based platforms, commercial off-the-shelf silicon and blades, allowance for proprietary fabric interface implementations and protocols, and so on.

The PCI Industrial Computer Manufacturers Group (PICMG) specifications define a number of possible fabric implementations as shown in Table 1.

Fabric	Specification	Topology	Interface Type
Base Interface	PICMG 3.0	Dual Star	10/100/1000BASE-T Ethernet
Fabric Interface	PICMG 3.1	Dual Star	SERDES (e.g., 1000BASE-BX and 10 GbE XAUI) Ethernet
		Mesh/Full Mesh/Hybrids	
	PICMG 3.2	Dual Star	1X to 4X InfiniBand
		Mesh/Full Mesh/Hybrids	
	PICMG 3.3	Dual Star	1X to 4X StarFabric
	PICMG 3.4	Dual Star	1x-4x PCI Express and AS
Mesh/Full Mesh/Hybrids			

**Table 1: PICMG 3.0 specification fabric options**

To enable a rich ecosystem of interoperable building blocks, it is necessary to select a limited set of profiles that will support a smooth transition over the development cycle of the technologies selected. Two such practical alternatives are Ethernet and Advanced Switching (AS), with our primary focus on the most useful topologies as depicted by the unshaded area in Table 1.

### Fabric Physical Interfaces

The switching interfaces for ATCA backplanes utilize different physical interfaces as explained in the next two sections.

#### Base Interface

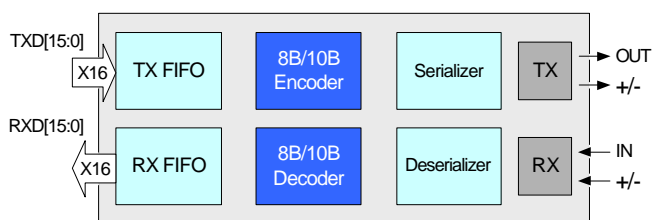
The Base interface defined in the PICMG specification is 10/100/1000BASE-T Ethernet. The Ethernet BASE-T interface employs four pairs of transmission lines (typically known as twisted-pairs in cables). The transmission lines on a backplane are implemented as pairs of copper traces whose characteristic impedance is the same as the twisted-pairs of a typical Category 5 Ethernet cable. An Ethernet port implements connections to a network or to a backplane switch port with a transformer (magnetics) to match the unbalanced transmit (output) and receive (input) of the Ethernet chip to the cable or backplane traces. The use of the magnetics is typically implemented to allow driving relatively long copper wires (or traces in the case of a

backplane) while maintaining excellent signal integrity; that is, good balance and no DC offset. Use of the BASE-T interface provides backward compatibility with a large installed base of products, making it easier to quickly move functionality, previously provided in a proprietary system or a PICMG 2.16-compliant system, to an ATCA blade.

#### Fabric Interfaces and SERDES Interfaces

The Fabric interface defined in the PICMG 3.x subsidiary specifications utilizes up to eight pairs of transmission lines and support protocols such as Ethernet, InfiniBand, and Fibre Channel over a SERDES interface. The SERDES interface eliminates the requirement for magnetics on node blades and fabric switch blades that are used on the Base interfaces. The use of SERDES interfaces reduces blade real estate and power consumption because the length of copper trace pairs on a backplane is generally quite short compared to typical cable runs such as Category 5 Ethernet. SERDES interfaces can support much faster speeds than long cable runs and have been tested up to 3.125 Gbps with technology and roadmaps extending out to 12 Gbps.

[Figure 1](#) illustrates a typical SERDES interface.

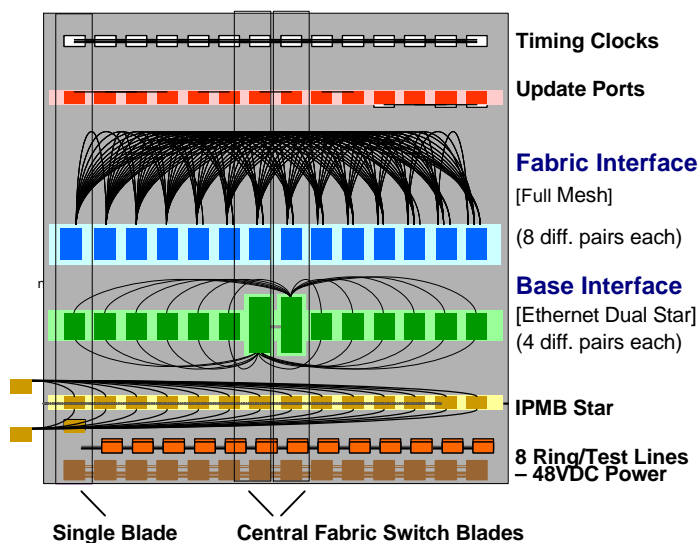


**Figure 1: Block diagram of a typical SERDES interface**

Differential signaling standards such as low-voltage differential signaling (LVDS) make it possible to transmit high-speed serial data at higher rates than conventional single-ended I/O standards.

**Star and Mesh Topologies for AdvancedTCA**

Figure 2 illustrates a typical ATCA backplane implementing a full mesh Fabric interface. Designed for an EIA 19” rack, this backplane can support up to 14 ATCA-compliant slots. The standard supports other topologies that are subsets of the mesh implementation such as Dual Star and Dual-Dual Star (two switches in each of the two switch slots) with implementations generally dependent on the platform’s intended application, such as wireless network access or edge routing.



**Figure 2: ATCA backplane with a mesh fabric interface**

**Backplane Fabric Bandwidth**

The bandwidth of the fabric backplane is a function of the access aggregate bandwidth per port (or per node) and the total offered bandwidth of the backplane. The most strenuous I/O applications come from multi-service switches and edge routers where these

applications may require a shelf full of OC48 interfaces or perhaps multiple OC192 and several OC48/12/3 interfaces to support an access distribution or fan-out. Wireless infrastructure applications currently require up to eight OC3s and two OC12s per shelf. Eventually, wireless applications will require more bandwidth as data-enabled handsets become more prevalent and drive infrastructure requirements.

A typical deployment might be two to four OC192 ports and eight to sixteen OC48 (or 32 to 64 OC12) ports. The presence of OC192 ports requires the access rate per port to be at least 10 Gbps plus some speed-up to account for backplane protocol overheads. It is obvious that when using 10 Gb Ethernet as the fabric, due to pin limitations it simply may not be possible to employ a dual star topology while supporting the 15-20 Gb/Sec needed per slot. However, as faster SERDES devices emerge along with more exotic backplane Printed Circuit Board (PCB) dielectric materials and connectors, which enable higher signaling rates, they will allow higher performance in simpler topologies.

**Allowance for Proprietary Fabric Interface Implementations**

Support for a proprietary fabric interface implementation may be needed for some systems in the near term. Since the ATCA backplane fabric interface is fabric agnostic, there are proprietary fabric implementations that can be implemented on an ATCA backplane provided they meet the fabric interface’s electrical properties and support a SERDES interface.

**E-Keying to Provide Safety**

E-Keying is a form of shelf management software control that provides safety to both the shelf and the blades inserted into them. E-Keying prevents blades from inadvertently being enabled with the wrong capabilities by the Chassis Management Module (CMM) (e.g., incorrect power level) when initially plugged in to the chassis. E-Keying is controlled by the CMM with information derived from the Chassis Data Module (CDM), often called the CDM “FRU” (Field Replaceable Unit). E-Keying, along with other blade and chassis safety mechanisms, such as mechanical alignment pins, ensures that blades are properly placed within the chassis slot that is intended to support the operational requirements of that blade.

**THE ADVANCED SWITCHING FABRIC**

Gb Ethernet works as a fabric for many applications. More demanding applications require traffic differentiation and congestion management in order to implement Quality of Service (QoS). Support for data

integrity, lossless transmission, fabric manageability, and other high-end features, plus the ability to scale bandwidth in reasonable increments, are all important features for many applications where simple over-provisioning of Ethernet bandwidth no longer suffices.

The Advanced Switching (AS) fabric is an evolution of the PCI Express fabric with which it shares the physical and link layer. AS goes beyond the load and store nature of PCI Express by providing a number of features that enable it to operate as both a data and control plane fabric for low-, medium- and high-end server and communications applications.

Among some of the features of AS are its self-routing nature, congestion management, scalability, multi-hosting and QoS.

AS uses the same 2.5 GHz SERDES using an 8B/10B block coding physical layer interface as PCI Express and InfiniBand. Each link is configurable in 1x, 2x, 4x, 8x and higher lane aggregations to provide considerable bandwidth scalability at 2 Gbps delivered per lane. The physical layer is isolated so that evolution to faster links is straightforward.

AS utilizes a link-level Credit-Based Flow Control (CBFC) scheme and has provisions for optional Explicit Congestion Notification (ECN) and Status-Based Flow Control (SBFC) as congestion management facilities.

AS provides for Virtual Channels (VC) as well as Traffic Classes (TC) to enable traffic isolation for QoS. AS also implements a Protocol Interface (PI) mechanism that allows easy upper-layer protocol encapsulation.

An AS fabric comprises end nodes and switch elements. A fabric might have one level of switching, or switch elements can be cascaded in a variety of topologies including Star and Full Mesh.

AS can be used as a light-weight-overhead fabric in and of itself or it can encapsulate other protocols and forward traffic transparently across the fabric.

AS traffic can be unicast or multicast and is self routing. Each packet contains a header that indicates the path the packet is to take from the source end node to the destination end node, and that identifies the payload type with a Protocol Interface (PI) number. In addition to the unicast header there is a "Path Building" header format and a separate multicast header.

Several PIs are reserved for management packets. All devices must support PI-0 and PI-4 and must be capable of generating PI-5, (although they are not required to respond to PI-5 should it be received). Most of the PIs

are reserved for designation by the industry or are vendor specific. Table 2 gives a breakdown of the various PIs.

**Table 2: Protocol interface values**

PI Index	Protocol Interface
0 (0:0) (0:8-126)	Path Building (Spanning Tree Generation) (Multicast) (Note that "0:x" notation indicates a PI-encapsulated PI.)
1	Congestion Management (Flow ID messaging)
2	Segmentation and Reassembly (SAR)
3	Reserved for future AS Fabric Management Interfaces
4	Device Management
5	Event Reporting
6-7	Reserved for future AS Fabric Management Interfaces
8-95	Governing Industry (AS) SIG-defined PIs
96-126	Vendor-defined PIs
127	Invalid

### Turn-Based Routing

The AS fabric has a self-routing mechanism that is based on a technique called *turn-based routing*. The header contains three fields related to this mechanism: the *turn pointer*, the *turn pool*, and the *direction flag*. The turn pool contains the path from the source to the destination node. Each switch element (SE) along the path taken by the packet interprets a field in the turn pool (that SE's *turn value*) containing enough bits to select one of its ports; the width of that field varies from SE to SE. The turn pointer points to the start of the next field to interpret. After an SE has moved a packet from the ingress port to the egress port it moves the pointer by the number of bits it has consumed to point to the start of the turn value field for the next SE to use.

Rather than identifying a port by an explicit port number, path routing treats the ports in a switch element as a circular list and uses the turn value to indicate the relative position of the egress port to the ingress port. This mechanism eases path discovery and also simplifies the creation of reverse paths.

A path can be up to 31 bits in length.

As an example in the forward direction, if the turn pointer was pointing at bit<8> of the path and an SE

has eight ports (three bits), then on egress from that SE, the turn pointer will point at bit<5> of the turn pool, which is the start of the next SE's turn value. Note that this mechanism allows for switches in multiple hops to have different port sizes, but the aggregate size of the switch ports combined in all hops is limited to the 31-bit size of the turn pool. If for example all switches had thirty-two ports (five bits), then a maximum of six switch hops could be employed in any one path before exhausting the turn pool.

If the direction flag is set, the packet is said to be moving in the "backwards" direction and the turn pool list is read from least to most significant bit. Backwards traversal of a path is used for example when sending a Backwards Explicit Congestion Notification packet, as explained later in the Congestion Management section of this paper.

If the direction flag is set to 0, the packet is moving in the forward direction and the turn pool is read from most to least significant bit of the valid field in the pool.

## QoS in AS

Quality of Service (QoS) in AS is provided by a set of logical channels within a physical link known as Virtual Channels (VC). Each VC provides its own queue so that blocking in one VC doesn't cause blocking in another. Since each VC has independent packet ordering requirements, each VC can be scheduled without dependencies on the other VCs.

There can be as many as 20 VCs in a given link. A minimum of one and up to eight bypassable VCs (BVC) are allocated to both ordered and potentially bypassable unicast traffic, up to eight ordered VCs (OVC) are allocated to ordered-only unicast traffic and up to four are allocated to multicast traffic (MVC). BVCs and OVCs are further described below.

Vcs can be used to establish connections between endpoints for control, management, and data.

Typically, when establishing a communications fabric, traffic will be segregated into service classes. These service classes correspond to the importance of the traffic or to delivery parameters such as QoS associated with that traffic. High-priority traffic, or traffic upon which the system (or the revenue-generating application that runs on the system) depends is considered "guaranteed" traffic. Guaranteed traffic must meet some set of delivery criteria in terms of overall latency and assurance that the packet is delivered. Less important traffic, such as real-time media may have low latency requirements, but can tolerably be undelivered in the event of congestion without too much

degradation in quality. Some management and house-keeping traffic or some service traffic (e.g., web browsing) may be re-transmittable and have no real latency requirements. This traffic is considered "best effort" traffic in that it is transmitted when there is no other, more important traffic pending in an output queue.

The packet header indicates which of eight Traffic Classes (TC) the packet is to use to traverse the fabric. Each transmitting port maps that TC to a VC for that link.

The eight TCs are distributed across the supported VCs; when fewer than eight VCs are supported, then multiple TCs can map to a VC. The TC can be used to apply particular serving policies at any point (i.e., a switch) in the fabric. There is no provision in AS to have independent queuing mechanisms for TCs in the fabric, so different traffic classes sent over a common VC will share queues and can experience head of line blocking among those TCs. Traffic differentiation really requires independent VCs (queues) for each non-interfering class of traffic.

There are two types of Unicast Virtual Channels (VC) in AS: Ordered-Only (OVC) and Bypass Capable (BVC). The packet header identifies which type of VC the packet is to be enqueued on, an OVC or a BVC. An endpoint must support a minimum of one BVC. An OVC contains only one queue, while a BVC contains two queues: *ordered* and *bypass*. The packet header indicates if a packet destined for a BVC should be considered an ordered or bypassable packet.

Due to the nature of some load and store protocols (such as PCI Express "non-posted requests"), packets can get stuck in a queue pending sufficient credit. Packets that are sent to a BVC propagate through the ordered queue just like they would through an OVC; however, when they get to the head of the queue the packets are evaluated to see if they are bypassable. If it is marked bypassable, and there is insufficient credit to forward it, a packet is moved into the tail end of the bypass queue, where it waits until sufficient bypass credit exists for it to be forwarded. This allows ordered packets with sufficient credits to bypass stalled bypassable packets.

## Congestion Management

There are several features (some mandatory and others optional) of AS that affect congestion management. The purpose of these mechanisms is to provide a robust fabric that will reliably forward data despite network irregularities. While some mechanisms merely preserve the operation of the fabric (at the expense of QoS),

other mechanisms are designed to potentially prevent the fabric from becoming congested in the first place.

At the link layer, a Credit-Based Flow Control mechanism prevents packet loss by queuing packets until sufficient transfer credit is available. Credit information is exchanged between the two endpoints of every link. Credits are created per VC at the receive side of the link and communicated upstream to the transmit side of the link where they are accumulated. Credit is created at a particular destination queue when an amount of traffic has been forwarded or consumed at the egress side of that queue, thereby freeing up queue space.

The packet is transmitted only if there are enough credits accumulated for the particular VC queue in which the pending packet is queued. Once enough credits are accumulated, the transmit side of the link sends the packet, then deducts an amount of credit (for that VC) by the packet size of the transmitted packet.

While the CBFC ensures that no packets will be lost due to congestion, it tends to violate any service class agreements during times of congestion. To provide a more fair mechanism, and one that can help prevent congestion in the first place, Explicit Congestion Notification (ECN) and Status-Based Flow Control (SBFC) may be employed. Both of these mechanisms are optional in the AS fabric. The ECN mechanism works end-to-end in the fabric, whereas SBFC is more local to a particular link.

ECN is an optional threshold-based mechanism that monitors the downstream (egress) queues of a switch to detect when they reach a certain (tunable) threshold level, indicating that the downstream could be experiencing congestion. When this threshold is reached, the next packet is marked by setting its Forwards Explicit Congestion Notification (FECN) flag, and a Backwards Explicit Congestion Notification (BECN) message is sent by the switch back to the source. Upon receiving a BECN, the source should throttle back its packet stream and must respond to the switch with a Congestion Management Message (CMM) in acknowledgement.

Downstream switches seeing that the FECN bit is set on this packet will not send further BECN messages upstream to the source. The end point receiving a FECN-marked packet may choose to do something at an application level, but any such action is beyond the scope of the AS specification.

Optional SBFC employs special messages sent between link partners that are used to manage the transmit end of flows entering a congested queue at an egress port of

downstream switch element. A target queue at the egress of the next-hop switch can throttle transmission from a specific upstream link partner queue by using transient or persistent XOFF and XON messages. The idea is to manage potential congestion conditions at a local level, before they impact the greater fabric.

With all three congestion management methods in place, the SBFC would attempt to predict congestion before it happens and alter the ordering of queued packets targeting different destination queues to prevent it. ECN would identify congestion as it is happening and notify the source so that it can prevent further congestion by altering its injection rate. CBFC would activate once congestion is already in progress and it can act to prevent packets from being lost, but does nothing to compensate for the congestion in the event that the source is still misbehaving and could cause congestion to spread through the fabric. Ideally the other mechanisms would prevent CBFC from ever kicking in.

If a packet is marked as perishable, then an SE can choose to discard it in the presence of congestion.

## **FABRIC MANAGEMENT IN ATCA USING EXAMPLES OF ADVANCED SWITCHING SOFTWARE ABSTRACTION LAYERS**

AdvancedTCA (ATCA) features and capabilities mentioned in previous sections mostly concern themselves with ATCA hardware, while standardization of software features, most importantly fabric management in ATCA, is still off in the future. Although ATCA defines a very powerful infrastructure, the new technology's value will be fully realized only with intelligent software. In the rest of this section we explore the option of Fabric Management (mainly Fabric Discovery and Multicast) in ATCA by using the Advanced Switching (AS) software abstraction layers and algorithms.

### **Fabric Management Architecture for ATCA (Multi-Fabric Model)**

To foster standardization and to realize the benefits of software commonality, a flexible Fabric Manager (FM) for ATCA provides an abstraction layer which as much as possible hides the details of the underlying fabric. Among services provided by a fabric manager are fabric discovery and management of multicast through the fabric

### Distributed Fabric Discovery Algorithm

The very first step an FM has to do is discover the fabric it is managing. Fabric Discovery (FD) is one of the key software components of the fabric management suite. During FD, the FM records which devices are connected, collects information about each device in the fabric, and constructs a map of the fabric. There are several approaches to how the FM might collect the information about all the devices. The approach chosen in this paper is a fully distributed mechanism, meaning, at any given time, the FM might be collecting information about more than one device. For each device in the fabric, the FM takes the following steps:

- Locates the device (active ports analysis).
- Reads the device's capabilities (including writing fabric-specific information into certain capabilities).
- Reads tables referenced by these capabilities.
- Updates its own tables based on information read from the device's capabilities/tables.
- If all devices have been discovered, constructs the shortest path table between every pair of devices in the fabric.

AS devices provide data structures similar to PCI capability registers to describe supported functionality. The first 256 bytes of an AS device's configuration space are identical to a PCI device's configuration space, which categorize the device.

The unique set of features supported by a particular device can be extracted from a linked list of capabilities located in the device's configuration space and initialized by the device during power-up time. Each capability is distinguished by a unique Capability ID and provides an offset to the next capability in the list. An offset equal to 0 indicates that the end of that capabilities list has been reached.

The first device FM discovers is the switch to which it is connected. For each capability read, the FM does the following:

- Checks whether this capability references tables, and if so sends packets to read the tables.
- Checks whether it needs to update its tables based on information found in the capability.
- Sends a packet to read the next capability.

If all capabilities have been read for a particular device, and the device is a switch or a multi-ported endpoint, the FM sends out packets on all active ports of that device (except for the port through which the device

itself has been discovered) to find new devices. This is the distributed nature of the algorithm. Instead of discovering devices one port at a time, the FM discovers devices on all active ports simultaneously.

Information collected by the FM about the devices includes the number of physical ports on a device, the status indicating which ports are active, events supported by a device, and more. If a device is an endpoint, then the FM also gathers information on which PIs that endpoint supports. If the device is a switch, the FM needs to read the switch's multicast support.

In order for an FM to distinguish between new and already discovered devices, the serial number of each device must be read. There are three cases that need to be considered when evaluating the serial numbers:

1. The serial number is all Fs:
  - a. In this case, the FM has encountered a new device and its serial number has not been hard-wired by a manufacturer.
  - b. The FM writes a fabric-unique serial number into that device and proceeds with the device's discovery.
2. The serial number is not all Fs and there is no record of it in FM's tables:
  - a. In this case, the FM has encountered a new device. It makes a record of that device's serial number in FM's tables and proceeds with the device's discovery.
3. The serial number is not all Fs and there is a record of it in FM's tables:
  - a. In this case, the FM has discovered a duplicate path to an already discovered device. The FM makes a note of it in its tables and *stops* discovering this device.

The FM keeps a list of the devices that are currently being discovered. When that list becomes empty, all reachable devices have been discovered. At this point, the FM calculates shortest paths between every pair of devices in the fabric, which will be used later for peer-to-peer communications, for example. Any duplicate paths found during discovery might be utilized during the run time of the fabric for fault resiliency or for traffic engineering to relieve chronic congestion. With a path-routed AS fabric, the path between any two end nodes is always unique. For efficiency and other reasons, some nodes might wish to run their own fabric discovery and collect information about the devices in the fabric.

## Multicast Algorithm

One of the important features an FM should be able to manage during the run time of the fabric is updating the appropriate devices during the multicast group changes, i.e., a device has left or joined the group, or has changed its status (writer, listener, both) in the group. For the AS fabrics, the devices requiring updates are AS switches.

One approach is to keep the amount of the information required by the FM to perform the updates to a bare minimum. At the very least, the FM would have to maintain the number of paths going through the ingress and egress switch ports for a given multicast group. Each time a member joins or leaves a group, or changes its status in the group, all the FM needs to do is perform a simple check of its tables to determine if any switch's multicast table needs an update.

Also, since a mechanism to avoid looping in multicast is not included in AS, one of the software solutions is to build a Spanning Tree of the fabric and use that Spanning Tree for the shortest paths between the devices.

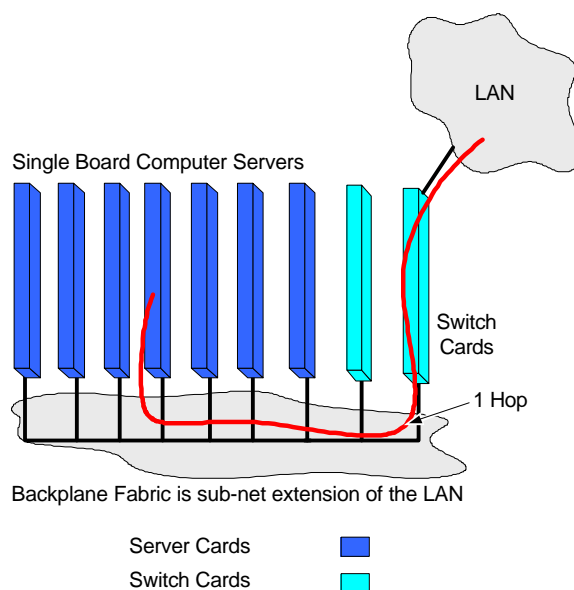
Combining the innovations of ATCA with sound software support will enable a smooth convergence between the communications and computing platforms.

## APPLICATION EXAMPLES UTILIZING ATCA

There are two primary architectural models for AdvancedTCA (ATCA): server and communications. In the communications architecture, there are two models: transit and transformation. Each of these models has unique attributes that affect the usage and performance of the backplane fabric.

### Server Applications

The server model consists of a chassis of single blade computers, each operating as a self-contained server. These servers use the backplane fabric as an extension of the Local Area Network (LAN) to which the ATCA chassis is attached. [Figure 3](#) illustrates this concept. Typically, only a switch blade separates the LAN from the backplane, allowing the backplane to operate as a subnet or as part of the actual LAN. Switch blades are shown in pairs for redundancy. Redundancy can be served in many forms: (1) hot spare with all traffic identical across both switches, (2) warm spare where traffic only passes through the active switch, and (3) load balanced where different loads pass over each switch, but managed so that if one fails, the remaining switch can handle all traffic.



**Figure 3: Server model application for ATCA**

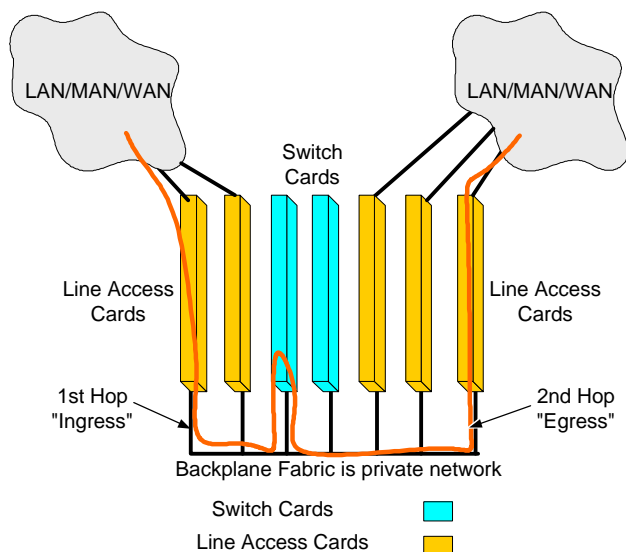
The backplane can also be used as a Storage Area Network (SAN) access media in this application. Signal flow originates and terminates at each individual server blade, flowing through the local switch blade to and from the LAN. The bandwidth demand of the backplane scales with the number of servers in the chassis.

For these applications, Ethernet or InfiniBand tend to work best, as such fabrics can be made homogeneous with an external LAN of the same type. Advanced Switching (AS) can also be used in this application, but it requires additional functionality to bridge between the AS backplane fabric and the external LAN to make the AS fabric appear as a subnet of the LAN.

### Transit Applications

Transit applications operate as a core router or switch at the edge or on an internal core network. In such applications, the ATCA chassis sits between two or more Wide Area Networks (WANs), Metropolitan Area Networks (MANs), or LANs—in some combination. Each line card (blade) functions as an access to and from one of those external networks. The signal flow requires a hash or table look-up that is local to the ingress line card that subsequently forwards the packet to the switch, which then forwards the packet to the egress line card. Additional management packets propagate the table updates, and some amount of protocol translation may occur between the ingress network and the egress network. If the application encapsulates the ingress traffic, then there will be a net

increase in overhead and bandwidth across the backplane. [Figure 4](#) illustrates an example configuration of the transit application.



**Figure 4: Transit application in ATCA**

In addition to the data traffic flowing between the external networks, the shelf also generates internal control information (e.g., state, management, table updates) that needs to be passed between control endpoints in the fabric. This data can be merged onto the common fabric interconnect or can be routed separately via the base interconnect.

The line cards can be symmetrical in terms of the access rates to the various external networks, or they can be asymmetrical where lower rate line interfaces on one network may aggregate up to a higher rate interface on another network. All such traffic is peer-to-peer with the routing path determining what traffic aggregates to what egress line card.

Performance for these applications is very demanding, requiring very high bandwidth accompanied with flow control and backpressure mechanisms in order to keep the throughput to a maximum level by avoiding head of line blocking in the fabric.

In order to function well, this application requires the more advanced features of AS, such as flow control and Quality of Service (QoS) support, which Ethernet lacks.

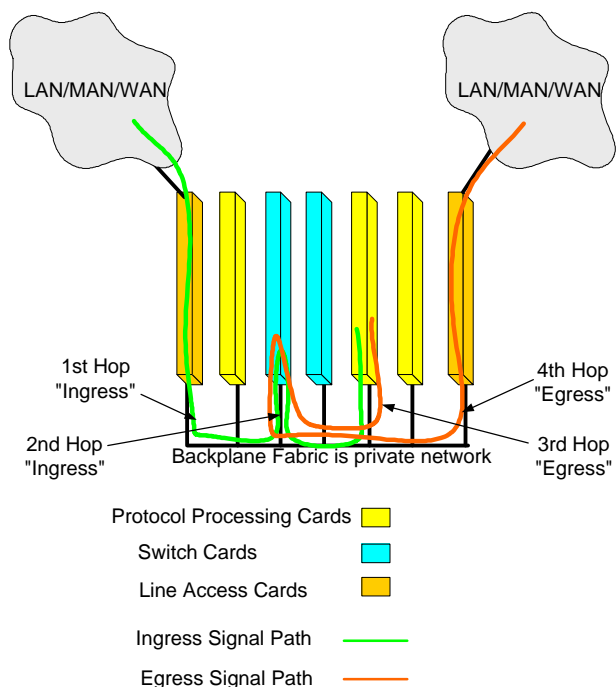
### Transformation Applications

Examples of a transformation application consist of wireless infrastructure or enterprise content processing equipment. Content processing applications include

Virtual Private Networks (VPN), firewalls, virus scanning, intrusion detection, and others. A transformation application functions as edge or access equipment that sits between two or more networks much like the transit application. Wireless applications have a relatively constant traffic arrival rate (not as bursty as core network applications). Some content processing applications collect long windows of data using a SAN and conduct trend analysis or pattern matching on the incoming traffic, which can be very time consuming.

Packets are processed in blades within the chassis. The signal flow typically requires the line (network access) cards to terminate the session, normalize the ingress network protocols, and load balance the ingress traffic across one of several processing blade destinations. The normalization of the incoming traffic translates the protocol of the ingress network to a protocol that is optimal for routing within the backplane. This may involve a simple encapsulation, but typically the ingress network protocol is replaced with an internal protocol. This action can actually reduce the net overhead of the incoming traffic, leading to an efficiency improvement on the backplane fabric.

Processing blades typically terminate the session at some level or process the upper-layer protocols before forwarding the traffic to the destination line access blade on the egress side. The egress line card terminates the internal session and specializes the protocol to match the external network. This flow is illustrated in [Figure 5](#). Notice that the local switch handles two passes of the traffic between ingress and egress. The protocol processing blades can consume or generate traffic or they can inspect or modify the upper layers of the packets they process. This type of operation distinguishes it from the transit applications in that system end-to-end latency is typically much greater due to the processing, and the ingress and egress flows are not necessarily symmetric. This leads to a more complex traffic model, but one that typically does not suffer from the congestion issues faced by transit systems at the core network.



**Figure 5: Transformation applications in ATCA**

The ingress line card fabric access bandwidth is typically many times the capacity of the protocol processing blades. Therefore, the balance between processing blades and line cards is typically many to one. Fabric performance is relatively high, using per-flow management and a speed-up ratio of 2 or 3 to 1 over the ingress line rates. Ingress and egress line cards might not be symmetric. Sometimes, for example, several OC3 lines aggregate into a fewer number of OC12 lines.

This application can make use of either Ethernet or AS (or even InfiniBand) equally well. Since ingress traffic is relatively non-bursty and load balanced at the line interface, statistical blocking is not a major issue; thus, the more advanced flow and QoS features of AS are not required. However, the link granularity of AS in this space is highly beneficial as this class of applications requires finer increments of fabric bandwidth to better match the wide range of expected performance. Furthermore, the switching function is a simple Layer 2 operation that ideally would take advantage of virtual output queuing to minimize the amount of speed-up that is required.

Lastly, it bears mentioning that ATCA has a fair amount of bandwidth-growth potential. Although 2.5 GHz SERDES are identified today, we can project the bandwidths that may be achieved in the future given better SERDES rates and using Mesh topologies. Table 3 illustrates this point.

**Table 3: Projected bandwidths with faster SERDES and mesh topologies**

Fabric	Per Node Access BW for Redundant Star	Per Node Access BW for Mesh
<b>Standard Gigabit Ethernet</b> (1 Gbps)	1 Gbps	14 Gbps
<b>AS, IB or Ethernet over SERDES w/ link aggregation</b>		
2x 2.5 Gbps SERDES	4 Gbps	56 Gbps
4x 2.5 Gbps SERDES	8 Gbps	112 Gbps
4x 6.25 Gbps SERDES	20 Gbps	280 Gbps

**CONCLUSIONS**

The AdvancedTCA (ATCA) standard provides a highly flexible and scalable architecture for fabrics in its backplane. While still considered “new” by industry standards, it is quickly gaining in popularity because it was designed from the beginning with the server and telecommunications equipment markets in mind.

The new AS fabric definition provides a low-overhead, scalable fabric that meets the demands of high-end transit applications while not being overly complicated for lower performance, transformation, and server applications.

In addition, it can be seen that fabric management in ATCA is a fairly straight-forward process, especially when using the AS fabric.

With a wide range of fabrics to choose from, a vendor can apply ATCA across numerous applications in both the server and telecommunications markets. Common interoperable profiles might use just the Base or Fabric interface as a unified data-plus-control fabric, or might use Base for a control fabric and Fabric with InfiniBand or AS as a high-performance data plane fabric. While Ethernet, InfiniBand, and AS are all defined to operate in ATCA, we find that AS is particularly well suited to communications as it provides the greatest flexibility and desirable fabric features; AS was designed predominantly as a backplane and chip-to-chip interconnect, as opposed to a LAN fabric that was extended to the backplane. As communications and compute continue to converge, we expect that the needs of the more demanding communications applications will drive the choice of the shared backplane fabric.

**ACKNOWLEDGMENTS**

Narjala Bhasker of CIG’s Technology Office also contributed to this paper.

## REFERENCES

- [1] Arapahoe Working Group, Advanced Switching Core Architecture Specification, Release Candidate 1.0, October 2003.
- [2] PICMG, PICMG 3.0 AdvancedTCA™ Base Specification, Revision 1.0, December 30, 2002.
- [3] PICMG, PICMG 3.1 Ethernet/Fibre Channel for AdvancedTCA™ Systems, Revision 1.0, January 22, 2003.
- [4] PICMG, PICMG 3.2 InfiniBand for AdvancedTCA™ Systems, Revision 1.0, January 22, 2003.
- [5] PICMG, PICMG 3.4 PCI Express and AS for AdvancedTCA™ Systems, Draft Revision 0.8, March 19, 2003.
- [6] IPMI-Intelligent Platform Management Interface Specification, V1.5, Revision 1.1, February 20, 2002.

## AUTHORS' BIOGRAPHIES

**Brian Peebles** is a principal architect in the Network Building Blocks Division of CIG with nearly 20 years of experience in the communications field. Brian's range of expertise includes radio, telephony, and data communications. His current activities focus on future wireless communications architectures, communications fabrics, and system-level hardware and software architecture and design. Brian has M.S.E.E./B.S.E.E degrees from New Jersey Institute of Technology. His e-mail is Brian.Peebles at intel.com.

**Chuck Narad** is a principal system architect in Intel's Communications Infrastructure Group Technology Office. His technical interests and background span workstation, server, and supercomputer systems architecture, CPU design, hardware/software interfaces, interconnects, network processing, and network processors. He has a B.S.E.C.E. degree and an M.S.E.C.E. degree from UC Santa Barbara, holds 17 patents and has over a dozen more pending. His e-mail is Chuck.Narad at intel.com.

**Victoria Genovker** is a software engineer in the Embedded IA Division of the Network Processing Group in CIG. She is a member of the Advanced Switching team which is simulating AS fabrics in software to validate the Advanced Switching specification. Victoria holds a B.S. degree in Computer Science from the University of Arizona and is finishing her M.S. degree in Computer Science from the National

Technological University. Her e-mail is Victoria.V.Genovker at intel.com.

**Karel Rasovsky** is a technical marketing engineer in Intel's Marketing and Platform Programs. He is currently responsible for introducing Modular Communications Platforms to the market. Prior to Intel, he was a product marketing manager at Lucent Technologies, focusing on converged voice and data solutions for next-generation networks. His other roles at Lucent included systems engineering, product management, and strategy development in the area of communications systems. He holds an M.S. degree in Computer Engineering from Florida Atlantic University, and a B.S. degree in Electrical Engineering from Technical Institute of Brno, Czech Republic. His e-mail is Karel.Rasovsky at intel.com

**Jay Gilbert** is a senior technical Marketing Manager within Intel's Communications Infrastructure Group. He is responsible for standards development, customer training, and technology evangelism in support of Intel's AdvancedTCA communications infrastructure development efforts. He has been with Intel Corporation for more than 12 years and holds a B.S.E.E degree from the Oregon Institute of Technology and an MBA from Portland State University. His e-mail is Jay.gilbert at intel.com

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

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

For further information visit:

[developer.intel.com/technology/itj/index.htm](http://developer.intel.com/technology/itj/index.htm)