CompactPCI®

PICMG® 2.20 R1.0

Serial Mesh Backplane Short Form Specification

October 21, 2002

FOR INFORMATION ONLY; DO NOT ATTEMPT TO DESIGN FROM THIS DOCUMENT

NOTE: This short form specification is a subset of the CompactPCI Serial Mesh Backplane specification, PICMG 2.20 R 1.0. For complete guidelines on the design of CompactPCI Serial Mesh Backplane implementations, the full specification is required.

For a full copy of the PICMG 2.20 specification, go to www.picmg.org, or contact the PCI Industrial Computer Manufacturers Group at 401 Edgewater Place, Suite 500, Wakefield, Mass., 01880. Phone 781-246-9318, fax 781-224-1239, email info@picmg.org.

INTRODUCTION

Objectives of the CompactPCI Serial Mesh Backplane Specification

The complete 2.20 specification defines a high-speed serial fabric for *CompactPCI* ® platforms. The purpose of this fabric is to enhance the data transport capability of *CompactPCI* ® platforms for high-end applications like telecom multi-service routers and gateways while maintaining compatibility with existing *CompactPCI* ® standards to protect investment. The goals of this fabric are:

- To increase backplane data transport capacities to a potential 700Gb/s
- Support multiple simultaneous transport protocols like ATM, IP, Frame Relay, GPRS Tunneling Protocol, and others
- Leverage the current evolution of standards in network processing and switch fabric technology
- Maintain maximum backwards compatibility with existing PICMG ® *CompactPCI* ® standards (e.g. 2.0, 2.1, 2.9, 2.12, 2.16, 2.17)

This enhancement to *CompactPCI* ® provides a parallel capability consistent with the AdvancedTCA™ specifications. The differences in form factor and capacity allow the 2.x family to continue to serve a separate marketplace. PICMG 2.20 allows common technology to transition between the two families.

The full 2.20 specification outlines a layered approach to provide interoperability for the mechanical, electrical, signaling, and packet protocol.

OVERVIEW

The complete 2.20 specification defines a high-speed serial fabric for CompactPCI platforms. This fabric enables high performance transport of data for a wide variety of protocols. The fabric can serve applications for ATM, Frame Relay, and many proprietary packet protocols like those used in wireless telecom applications.

The fabric consists of a rich set of differential serial signals capable of 1.25Gb/s, or 2.5Gb/s. The channels are arrayed in a "mesh" configuration that gives each slot a full set of interconnects to every other slot. This mesh arrangement supports a distributed switch fabric architecture. It has advantages in scalability and traffic management. The mesh may also be used as a subset of smaller fabrics in "star" configurations.

The high-speed serial fabric is facilitated by a new connector family. The Zd connector family is supported by multiple vendors. This connector is specifically designed for high-speed differential signals up to 5Gb/s. The HM2 family currently used by CompactPCI will not support these kinds of signals.

The fabric is located entirely within the space occupied by J4. It follows the precedent set by PICMG 2.5 (H.110) to use J4 as a common transport layer. This fabric is specifically oriented to packet-based applications. By utilizing J4, general-purpose boards (with no J4) will interoperate in standard CompactPCI systems, those with H.110 backplanes, and those with this high-speed serial fabric. This specification also co-exists in systems supporting PICMG 2.16.

Switch Fabric Architecture

Switching systems can be characterized in layers. Protocol specific switching systems interface to specific layer 2 packet formats like Ethernet or ATM. Many low to medium performance fabrics can now be implemented in a single device making all the specifics of the switching architecture internal.

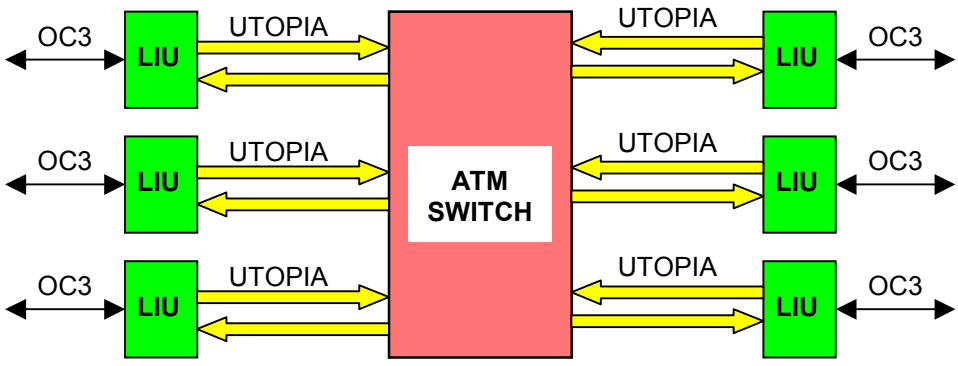


Figure 1: ATM Switch Example

Higher end switches utilize more layers in their architecture. This is to facilitate higher data rates and more complex traffic classification and management. Higher end fabrics define devices like traffic managers and switch fabrics.

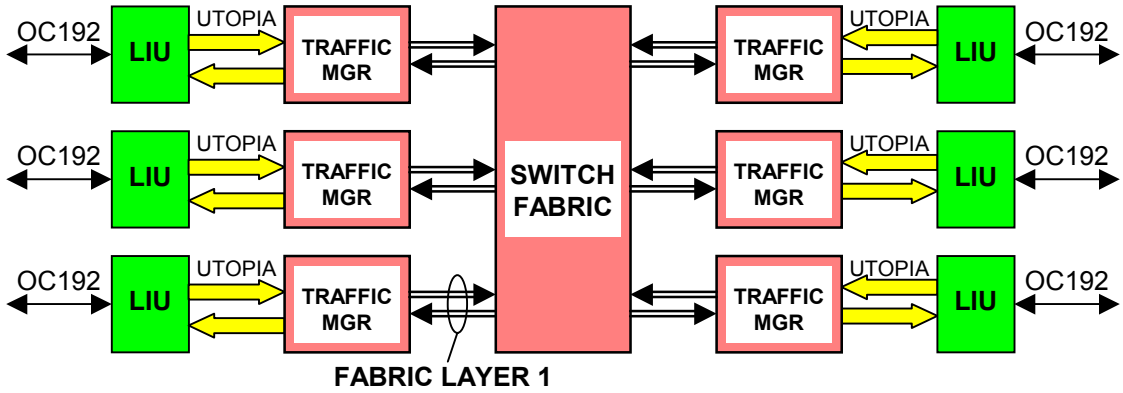


Figure 2: High End ATM Switch

The traffic managers interface to the layer 2 packet. They classify, queue, and switch (or route) the layer 2 packet. The traffic managers encapsulate the layer 2 packet into a "layer 1" packet that contains information about how to transition through the switch fabric. The traffic managers perform the more complex tasks so the fabric does not have to. The fabric just transports the data to the appropriate destination.

Many fabrics extend the interface between the fabric and managers to implement complex performance enhancing features. Most fabrics of this class utilize a proprietary protocol between the traffic managers and fabric requiring that these components come from the same vendor.

A Protocol and Vendor Independent Fabric

In the telecom market, for example, many different networks exist and many different protocols exist. Equipment is required to inter-work between these protocols. Either the fabric must allow multiple protocols to co-exist, or all elements of the platform must convert to a single common protocol.

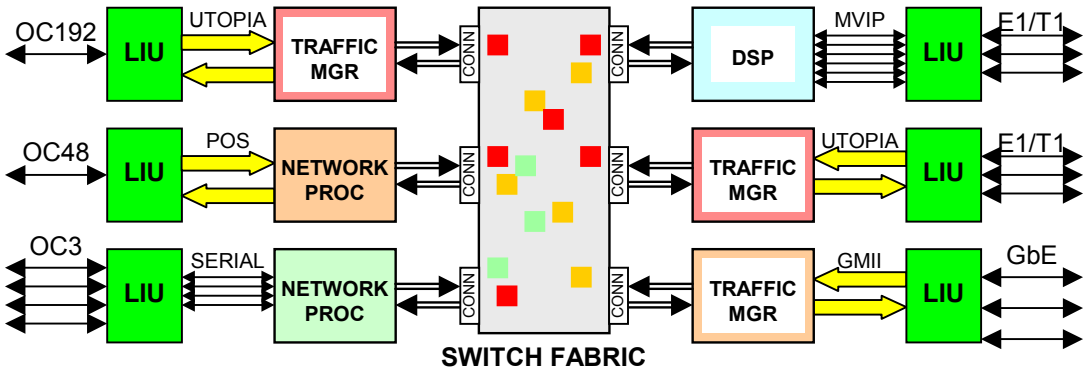


Figure 3: Multiple Protocol System

In an open architecture system like CompactPCI, it is also necessary to distribute functions across multiple boards. For interoperability, boards must be able to share data at the lowest layer in the fabric. A common layer 1 protocol that encapsulates any layer 2 packet format provides for protocol independence.

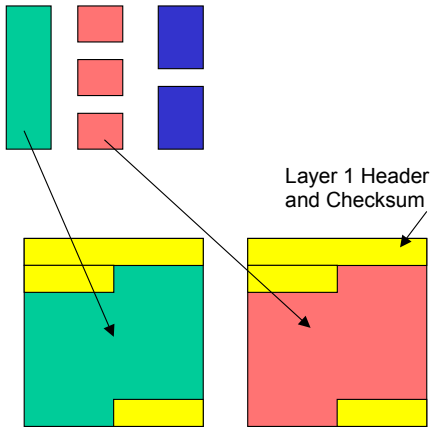


Figure 4: Layer 1 Encapsulation

The Network Processor Forum has defined a common switch fabric interface in its CSIX-L1 specification. This specification provides a common interface between all types of "traffic managers" and the switch fabric. The goal of this specification is to allow interoperability between different fabrics and different vendors. With the advent of network processors, this kind of open architecture fabric is important to an open architecture system.

A Distributed Fabric Architecture

Fabric architectures are classically built around "star" architectures. All the devices in the system send their data to a central resource to be forwarded to a different device.

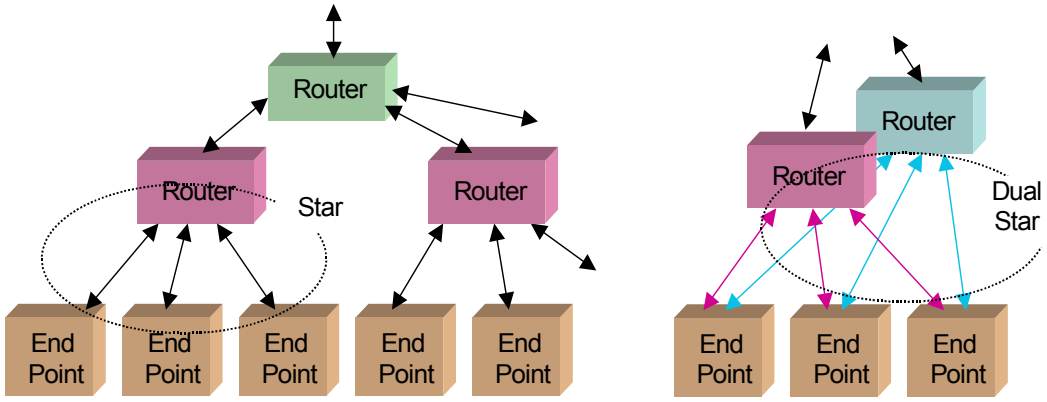


Figure 5: Star and Dual Star Topologies

Another configuration for a fabric is a "mesh". A mesh topology is a superset of a star topology. Connectivity is increased where all nodes have connections to all other nodes. Each node can be an endpoint, a router, or both. The figure below illustrates how a node can act as a star point for all other nodes in the system.

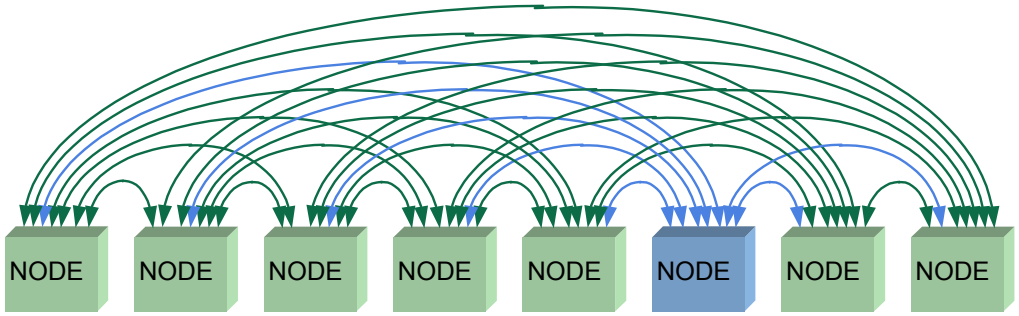


Figure 6: 8 Way Mesh Example

A mesh network "distributes" the fabric among the node elements. Instead of a single N x N switch, the fabric contains M ⇒ 1xN switches, where M ≤ N.

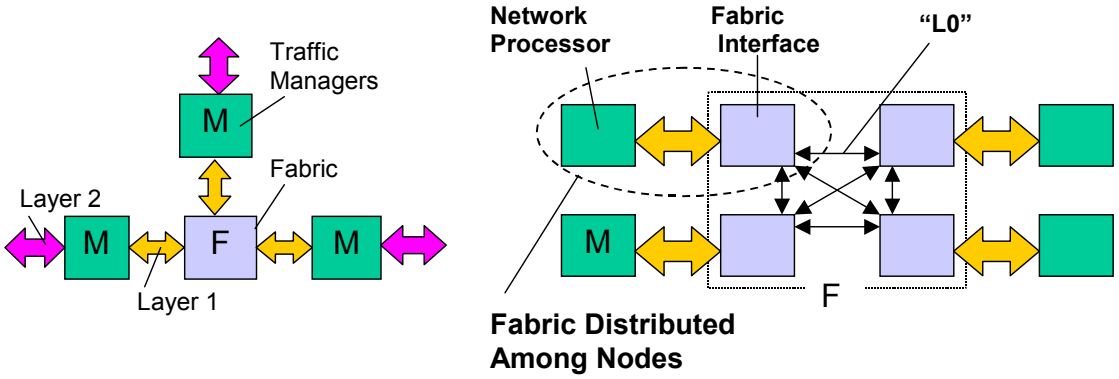


Figure 7: Distributed Fabric

This distributed fabric uses the richness of interconnect in the mesh to eliminate some of the contention that a crosspoint switch encounters. An NxM switch, for example, allows only a single connection from one of N sources to each of M destinations (M total packets in transit). Sources must arbitrate for a connection. As fabrics become more sophisticated, they must participate in the traffic management. A distributed fabric allows each source to transfer data to the destination. Each node implements its own traffic management according to its own needs.

Mesh networks offer more resilience than star networks. There is no dependence on a central resource. If a node fails, only that traffic associated with that node is affected. The incremental capacity to protect a node is 1/N (for N+1 systems) instead of N (in 2N systems).

In addition, mesh networks are more scalable. Once the mesh network interconnect is in the system, then capacity is added with each card. With a star network, the central resource has to have full system capacity, even if it is not immediately used.

Mesh Topology

There are many different ways to interconnect a mesh. The mesh defined in the full 2.20 specification uses a "physical" channel arrangement. Each channel is physically associated with a slot.

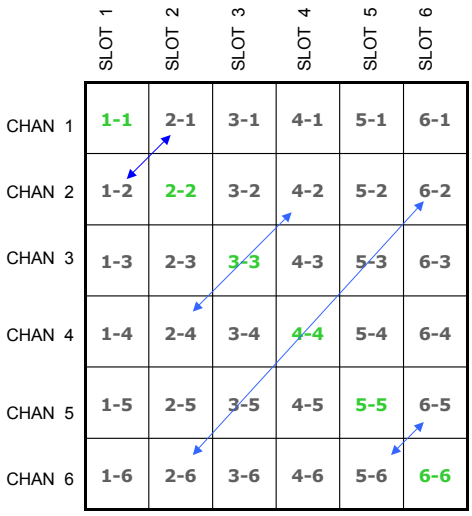


Figure 8: Square Mesh Topology

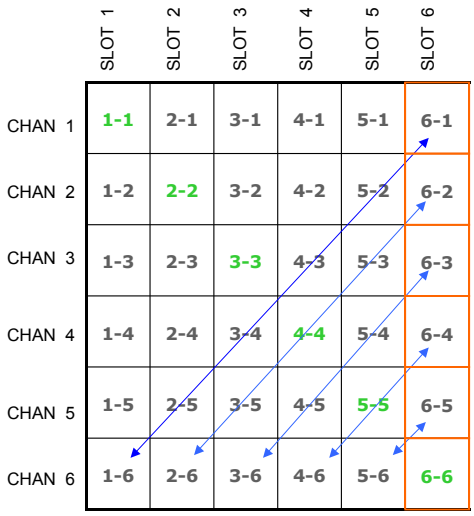
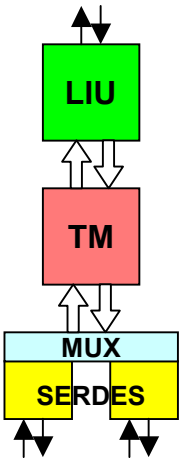


Figure 9: The Mesh as a Hub

Figure 8 illustrates that each channel is associated with a slot. Channel 2 in Slot 1 connects with Channel 1 in slot 2, and so on. This physical arrangement is symmetrical around the diagonal axis. Every connection to a specific slot is in the same physical location. It permits any slot to act as the hub for any subset of the mesh signals.

Distributed Fabric Interfaces

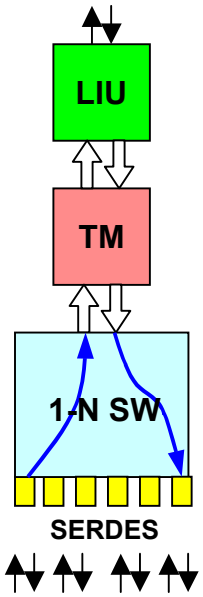
Each node in a distributed system can act as a node, as a switch, or both. As a node, a board may send all traffic to a specific channel without any content related switching. This is the simplest form of the fabric interface. The interface simply encapsulates the traffic into the desired protocol and sends it to a fixed port. The board in the slot corresponding to this channel becomes the switch for the traffic. If boards are hard wired for specific channels, the switch boards must be placed in specific slots.



A node may contain a full population of serial interfaces and a multiplexer. This is the same as the previous example, but with the connectivity to allow any board to be provisioned in any slot.

A node may implement a basic degree of channelization by sending traffic to different destinations based on different sources. A UTOPIA port address, or a SONET sub-channel may be the determining factor. All of

these methods allow the board designer to use a relatively fixed packet encapsulation mechanism. This may be the preferred method based on the capabilities of the traffic management. The local traffic manager might still need to do traffic queuing and shaping.



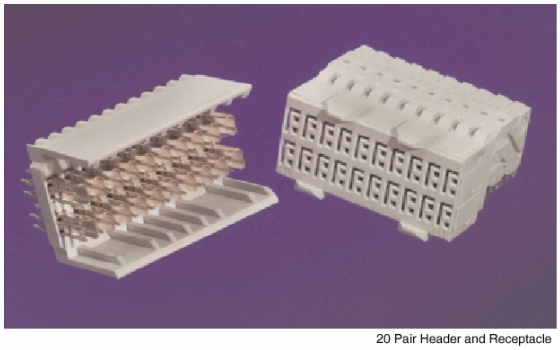
To implement a full switch, where packets are sent to specific destinations based on the packet itself, requires a more sophisticated interface. The traffic manager must encapsulate the traffic dynamically on a packet-by-packet basis. The fabric interface must examine each packet and multiplex (switch) the packet to the correct port.

In a fully switched application, the fabric interface still only needs to distribute (aggregate) traffic to (from) the head end interface. This is 1-to-N switching. In a fully distributed system, traffic would not be switched between ports as they have their own direct route. In the node-switch relationship, traffic would still need to go to the traffic manager to have Quality of Service rules applied before transitioning the fabric again.

The 1-to-N configuration simplifies the node card switch, but it does remove the ability to "bypass" down links by routing through another card. If the Traffic Manager (TM) has the reserve capacity, it could provide the additional forwarding. The failure rates of the links should be very low, but if this additional capability is desired, the switch can be expanded to a full N x N switch.

Interoperability

The full 2.20 specification defines interoperability in two layers: the physical layer and the protocol layer. The physical layer defines the mechanical and electrical requirements for the signaling. This layer is common to all implementations of the fabric. It utilizes a multi-sourced connector (Zd) for the mechanical interconnect.


20 Pair Header and Receptacle

The physical layer utilizes an industry standard physical signaling layer (802.3z) for the electrical layer. (802.3z is incorporated in *IEEE 802.3-2000* sections 39 – 43). The 802.3z physical layer can be supported by any number of SERDES available from multiple sources. These interfaces are also available in programmable logic.

The physical layer is contained entirely within the envelope of J4. The Zd connector will not mate with the 2mm Hard Metric (HM2) connector, however, boards with J4 depopulated will mate with backplanes supporting this fabric. The HM2 and Zd connectors do not engage sufficiently to bend pins or damage plastic housings.

The PICMG 2.20 configuration allows coexistence with other fabrics like PICMG 2.16 and PICMG 2.17. The switch fabric does not need to encompass all slots in the system. A

backplane can contain standard CompactPCI slots, or 2.16 or 2.17 fabric slots in addition to the 2.20 mesh slots. The mesh slots can contain the 2.16 and 2.17 node connections.

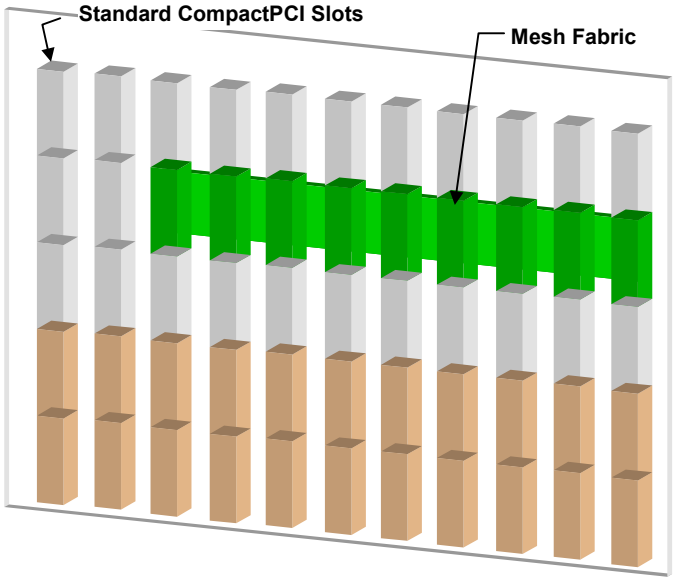


Figure 10: Mesh Fabric Location

Table 1: Relationship to Existing PICMG Specifications

Specification	Compatible?	Comment
2.0 - CPCI	Mostly	J4 is not HM2
2.1 - Hot Swap	Fully	Point to Point network adds no new hot swap requirements
2.5 - Telephony	No	
2.9 - Mgmt	Fully	
2.10 - Keying	Partially	Zd keys all modules for this backplane, does not use HM2 keys in J4/P4.
2.12 - Software	Fully	
2.16 - CPSB	Mostly	Mesh need not extend into fabric slots, J4 not HM2
2.17 - Star Fabric	Partially	Compatible with the centralized fabric approach

The protocol layer is defined separately. The 2.20 specification defines a protocol layer based on the Common Switch Interchange (CSIX) protocol to leverage multiple switch fabric vendor support. The complete specification document contains extensions to the CSIX definition for a serial version of the protocol.

The PICMG 2.20 specification is layered to allow other protocols to be supported. The mesh architecture allows subsets of the fabric's interconnects to be used with different protocols. They can all co-exist as long as the physical layer remains consistent.

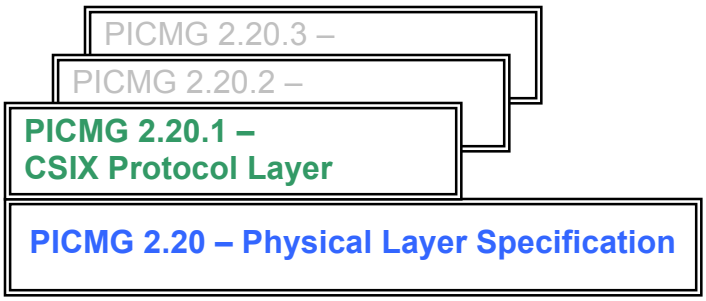


Figure 11: High Speed Serial Fabric Specifications

For specifying interoperability, PICMG 2.20 is numbered with an extension to specify the protocol. For example, a PICMG 2.20.1 compliant board uses the CSIX packet protocol.

###