The Role of PCI Express® in Wired Communications Systems

By Ian Dobson, Principal System Architect, CTO Office

PCI Express® (PCIe®), the latest generation in the PCI family of protocols, is backed by an extensive ecosystem and offers designers a high-performance, general-purpose, industry-standard interconnect at relatively low cost. Those inherent advantages are already driving its rapid adoption in the PC, server, and storage markets. Like its predecessors in the family, PCIe will inevitably find wide use in communications systems as well.

This paper will look at some of the potential applications for this popular industry-standard interconnect in the wired communications infrastructure. It will begin by examining the various tiers of the network hierarchy and how their particular data flow requirements differ. Next, it will review the PCI interconnect family and the new capabilities PCIe brings to the specification. Building on those observations, the paper will outline some of the likely applications for PCIe in the network landscape, such as in on-card control planes and central processing architectures. Finally, it will explore how the new capabilities embedded in PCIe open the door to new, less apparent applications in wireline communications equipment, such as in control planes between intelligent cards and between system shelves.

The network view

Wire-line communications equipment is designed to match the requirements of the market segment it addresses and where it sits in the network hierarchy. Figure 1 provides a simplified view of network topology with end users at the right and bottom of the figure and the “deeper” tiers of the network to the top and left of the figure.

Generally speaking, the market segment that a piece of equipment addresses will dictate the line protocols it supports and the type of packet processing it performs. However, the tier of the network in which it resides dictates the system’s internal architecture regardless of its market segment. As seen in figure 1, a number of systems straddle the borders between the network tiers. There are also a number of hybrid systems that perform the functions of more than one tier within a single system.

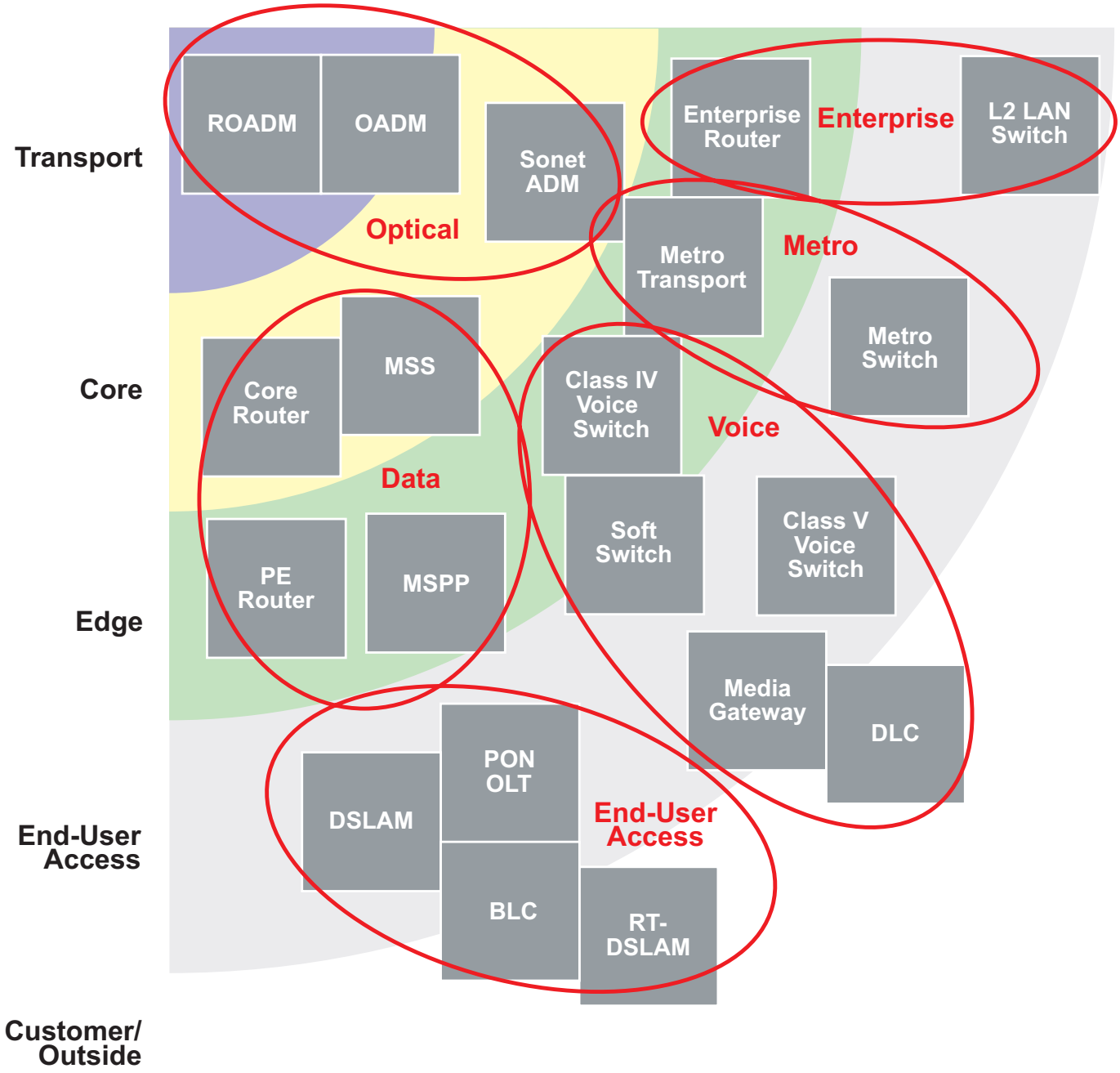


Figure 1. Communication network tiers

Transport tier

The transport tier provides the long-haul transport of data. It is generally statically configured and cares little about the content or protocols of the data it transports. Data is tunneled through operatorconfigured paths, with engineered failover schemes to support high availability. These tunnels guarantee Quality of Service (QoS) by not sharing bandwidth anywhere along their route.

Over the last few years, nodal capacities have increased in the transport tier, primarily driven by the move to dense wave-division multiplexing (DWDM). These wavelengths provide a 2.5, 10, or more recently, 40 Gbps channel between two endpoints. DWDM technology currently can carry up to 160 x 10 Gbps channels on a single optical fiber.

The major driver for equipment designers in this space is reducing operating expenses. The recent development of the reconfigurable optical add-drop mux (ROADM) has allowed network operators to remotely reconfigure paths rather than assign a technician to switch and tune a laser module and refraction grating. Architecturally, ROADMs are now extremely similar to core switches.

Core tier

Nodes in the core tier groom traffic into predefined tunnels for transport towards its ultimate destination. These predefined tunnels are provided either by the protocol-agnostic transport tier or by dedicated point-to-point links directly between core nodes. This equipment usually prioritizes functions by referring to labels or tags added to data packets at lower network tiers. Therefore, these devices are somewhat aware of the services they carry, but due to their high aggregate data rates (hundreds of Gbps—tens of Tbps), they only have a low touch on passing data.

Given the large amount of data passing through a core node, the control plane must supply enough capacity to support tables large enough to maintain the system. This large amount of data also makes the availability of the node of paramount importance. Designers must pay great attention to monitoring the health of the elements of the node and employ redundant elements to minimize data loss.

Edge tier

The edge tier is generally defined as the point where a service begins and ends in a service provider's network. Lower elements in the hierarchy aggregate or forward traffic based on very basic criteria into the edge tier where true service-aware QoS enforcement and forwarding occur. Because of the relatively expensive nature of the service-aware classification, traffic management, and forwarding functions, most network operators only deploy them on links where the relationship between peak and average loading is fairly close (on the order of 3:1). This is important because the functions must be sized to accommodate peak loads, and the disparity between peak and average loads translates into wasted capacity.

Architecturally, an edge node is very similar to a core node, but it exhibits lower aggregate capacity (tens to hundreds of Gbps) and "higher touch" on the data. Attention to monitoring and redundancy is important here also. Control plane capacity within an edge system is very similar to that in the core. While equipment in the edge tier must maintain relatively small tables, this advantage is offset by the number of exception cases in data processing with which the system must deal with in the control plane.

Aggregation nodes

When a large number of small port-count access nodes exist in reasonably close proximity, it often makes economic sense to add a node between them and the edge node. These aggregation nodes smooth out the bursty but low-average-rate traffic from each access node, combining it with traffic from other access nodes. By regulating the discrepancy between peak and average loads into the edge node, these elements help the system more efficiently utilize relatively expensive edge ports. It should be noted that access nodes with large port counts inherently provide this level of averaging themselves and, therefore, connect directly to the edge nodes.

Since the entire rationale for this class of nodes is purely economic, low cost is crucial. Designers achieve this by developing aggregation nodes that support a relatively small number of protocols and provide only a minimal examination of the data before executing discard decisions. In addition, these devices tend to provide redundancy only for the central functions and uplink cards, and not for the downlink cards.

Access tier

Defined as the network end of the "first mile," the access tier terminates the service provider's end of the user's connection. In this environment, the physical placement of an access node is often dictated by the physical limitations of the protocol supported. For example, a VDSL link may be able to achieve a transfer rate to the user of 100 Mbps, but only over a distance of 300m. In contrast, a gigabit passive optical network (GPON) service can often reach 20km or more from the optical line terminal to the user's premises. Most protocols will fall between these two extremes. These physical considerations usually dictate a designer's decisions on numbers of ports, nodal capacity, control plane capacity, and redundancy strategies. Therefore, designers typically employ a wide range of architectures in these systems.

In addition, traffic loading on any individual customer connection in this tier of the network is highly unpredictable on a moment-by-moment basis. On average, however, it tends to represent a very small fraction of the available bandwidth on any given link. This results in access system designs that are typically oversubscribed (total user bandwidth > uplink bandwidth to the network). Historically, oversubscription rates of 50:1 are common in access nodes. However, these oversubscription rates are expected to drop in the near future as users run more delaysensitive services over these links.

The PCI family

Figure 2 and table 1 illustrate the relative speed of members of the PCI family of interconnects and compare their key characteristics to some other familiar interconnects.

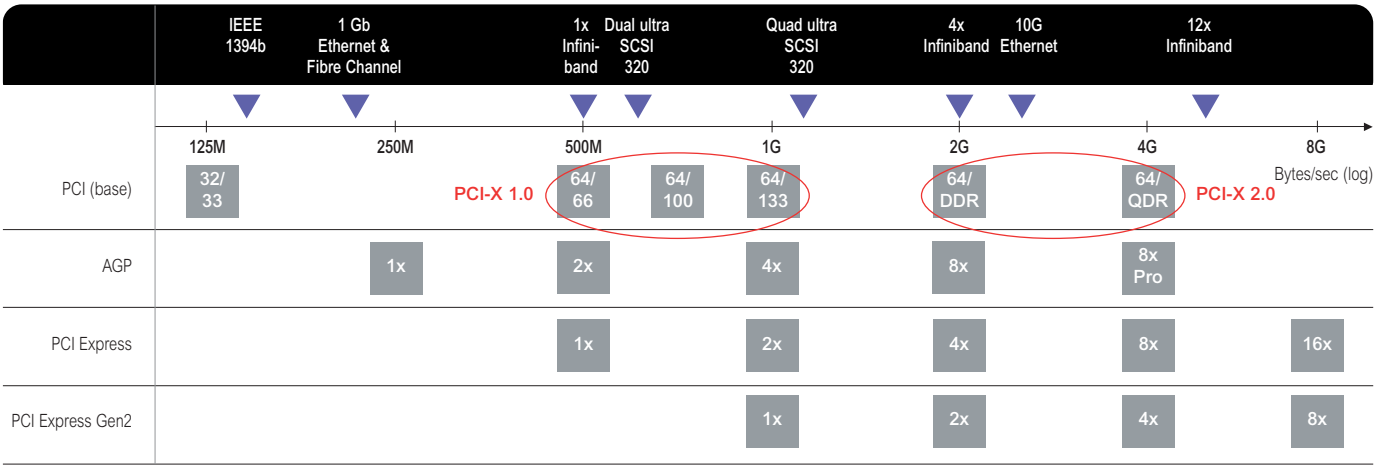


Figure 2. Comparative bidirectional bandwidth chart

The PCI family was originally designed to provide a standards-based expansion bus within personal computer architecture. The architecture supports up to six plug-in modules (with the seventh entity being the motherboard), and its electrical characteristics define a copper-based interconnect that runs across connectors.

The original PCI specification also assumed that a single CPU complex would manage setup (enumeration) of all other modules. This assumption dictated that the addressing scheme would be a direct one, with only one shared address space throughout. However, the designers did reserve some of the bits of the addressing to support bridge entities. While the overall scheme assumes that the majority of all traffic runs from the master CPU, peripheral devices can access one another in a peer-to-peer fashion over the interconnect.

Protocol	Raw speed	Number of pins	Max trace length
PCI 32/33	33 MHz	47	40"
PCI-X 1.0	133 MHz	80+	6"
PCI-X 2.0 QDR	533 MHz	80+	6"
1x PCIe Gen1	2.5 GHz	4	40"
1x PCIe Gen2	5 GHz	4	11"
Serial RapidIO	3.125 GHz	4	40"
SPI 4.2	500 MHz	42	12"
1x Infiniband	3.125 GHz	4	Variable
Gigabit Ethernet (CX)	1.25 GHz	4	7m
10GE	9.953 GHz	16	-
1x AGP	66 MHz	80+	6"

Table 1. Characteristics of key interconnects

As the most recent addition to the PCI family of protocols, PCIe embodies all the concepts embedded in earlier versions, with a few minor exceptions. Instead of a parallel topology, however, it uses a serial point-to-point approach. Table 1 illustrates how, as the PCI family has evolved, designers have traded off increases in speed against signal reach and the number of interconnected entities. PCIe takes the next evolutionary step in this process by introducing a switch for interconnecting entities. However, it also supports a longer reach for signals and, with the assistance of the switch, increases the number of connected entities.

Use of PCIe features in communications systems

A number of features at the physical and data link layers of the PCIe protocol can be used to enhance the availability of a communications system. PCI Express packets are protected by a link-layer cyclic redundancy check (CRC) and sequence number when moving between two entities. The bus also provides an optional end-to-end CRC capability to supply an extra layer of data integrity protection. As another related option, an advanced error reporting capability can help isolate the location of any failure.

Additionally, any PCIe link that consists of multiple PCIe lanes, such as an x8 width link, supports automatic lane reversal. This feature allows the user to define if Lane 0 is on the logical left or logical right of the link and is very helpful in preventing the "crossover" of traces when laying out a PCB. This capability is especially important in modular systems like those often used in communications applications, because it supports optimal layouts on individual modules, rather than forcing designers to comply with the same link order. To this end, PCIe also supports automatic polarity reversal. In communications systems, these features may also prove useful in the reduction of signal coupling on dense backplane layouts. PCIe links also support automatic link width negotiation.

When a link is brought up, the two ends exchange packets to train the link for optimal signal propagation. As part of this process, the two end points also agree on the number of lanes they will support across the link. This capability is very useful when supporting a range of optional plug-in modules that may require different link widths. Unused lanes are powered down once the negotiation is completed. However, it is important to note that the PCIe specification only requires a device to negotiate to full width or an x1 width. Other intermediate widths are optional.

Communications systems designers can use this feature, in conjunction with auto-lane reversal, to permit the system to continue to operate in the presence of a failure of a single lane within a link. The link can be reset and automatically negotiate to use the half of the link that remains usable. While this feature utilizes only half the bandwidth, it does provide an open channel for recovery of key information or for partial functioning of the module.

This is unclear: What is doing the negotiating?

Finally, PCI Express also provides a link-level retry capability. This function retries the transfer of packets between two adjacent PCIe devices without any intervention of higher level software when an error or a lost acknowledgement occurs.

I/O virtualization extensions

The PCI special interest group (SIG) is defining some extensions to the PCI family of specifications that will allow multiple independent CPUs to share I/O resources. These new capabilities will be extremely useful in blade server applications by allowing each CPU blade in the system, for example, to operate as if it has sole control of an expensive storage area network (SAN interface), when in fact, that interface is shared by all CPUs in the system. Since most communications systems dedicate I/O to individual CPUs, or a group of CPUs cooperatively shares resources, these I/O virtualization extensions to PCIe will probably not be widely used in the communications infrastructure.

Primary PCIe applications

The adoption of the original PCI protocol in PCs and servers paved the way to the widespread availability of components with PCI interfaces for communications systems. While early device development focused on typical PC components such as Ethernet network interface card (NIC) chips, PCMCIA card controllers, and disk controllers, the bus eventually became the protocol of choice for CPU ports on almost all communications devices including framers and NPUs. Given the performance, low cost, and supporting ecosystem the PCI interface offers, this trend will likely continue with PCIe. Moreover, PCIe offers an additional compelling advantage—low pin count. While a CPU port required 47 signal pins to implement PCI 33/32, it needs only four pins to support an x1 PCIe port. Even though the new interface uses higher speed signals, its reduced pin count and its impact on board layout, routing, and connector design will prove highly attractive to board designers.

On-card control plane

Control plane accesses to devices usually involve access to registers or the movement of blocks of data into and out of memory. This function is well suited to the memory-based paradigm used in PCIe. In addition, designers of control plane functions within a card tend to use a hierarchical model where a single housekeeper CPU manages each individual device on the card. This approach is also well matched to the PCIe model. Finally, traffic patterns in a control plane on a single card are almost exclusively between the housekeeper CPU and each individual device, rather than peer to peer between devices.

While PCIe allows multiple priorities of traffic, control plane traffic within a single card is usually low enough in bandwidth to avoid the need to segregate different types of traffic from one another. Over time, designers will migrate from current CPU port protocols on communications cards.

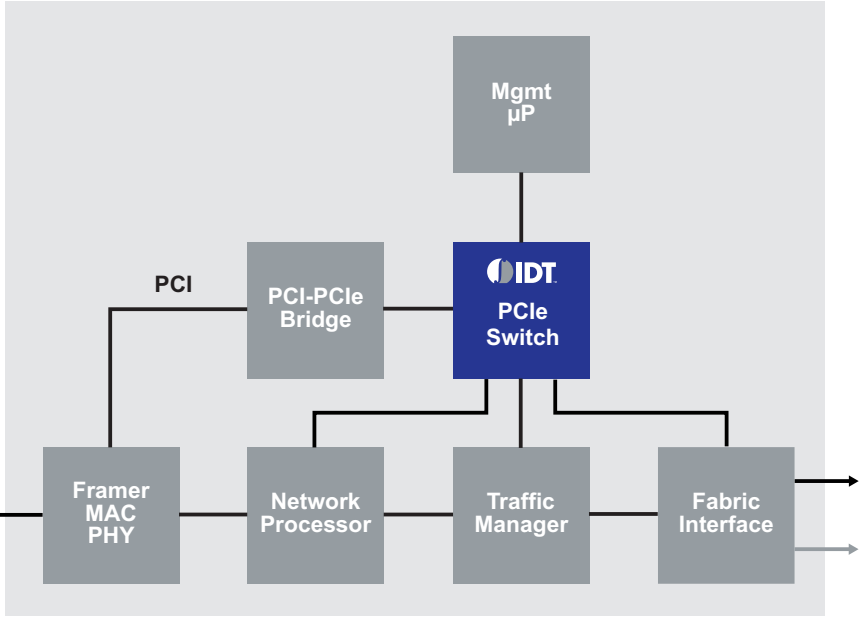


Figure 3. PCIe for on-card control plane connectivity

Intercard control plane to "dumb" I/O cards

In many communications systems, devices on one module are controlled by a CPU on another module. Two examples are illustrated below: The line interface module model (see figure 4) and the central processing architecture model (see figure 6).

LINE INTERFACE MODULE MODEL

One of the most common customizations in any communications system is selection of I/O. For this reason, line interfaces are often packaged on their own modules. Depending on the system type, this configuration may be limited to the physical aspects of the I/O or allow selection of a number of different protocols. In the first case, users often wish to select or change features like the physical connector type, optical frequency, or output power. In some systems, packaging decisions may dictate separating the line driver/receiver and connectors into a removable module (see option b in figure 4). Other architects may choose to partition their system in a similar way to simplify maintenance, since the line driver often exhibits the highest failure rate.

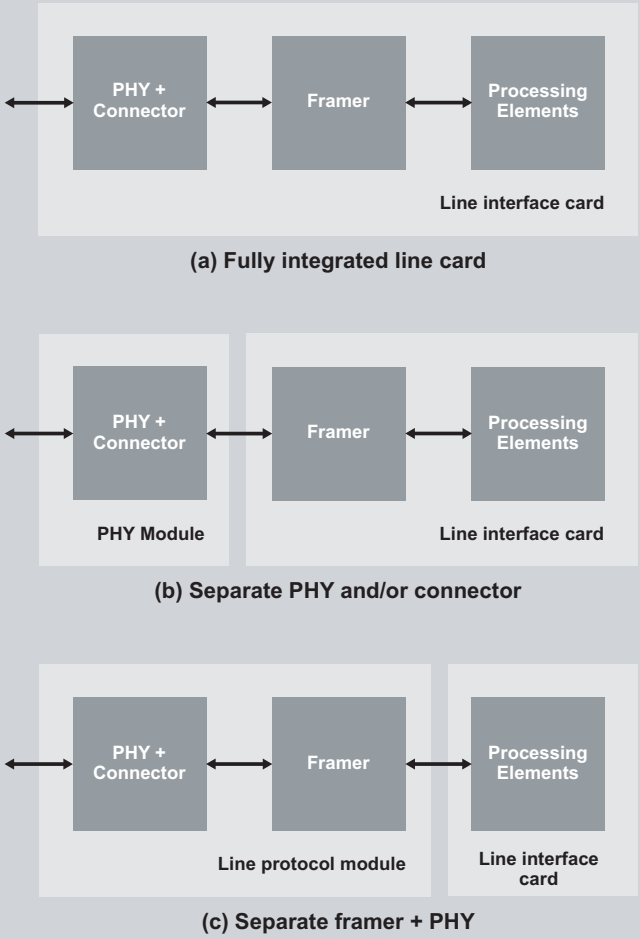


Figure 4. Line card partitioning options

Some designers will decide to include a framer function with those components on a single removable module (see option c in figure 4). This strategy allows the customer to select not just the reach and frequency of the interface, but also the protocol supported (e.g., ATM over SONET vs. packet over SONET). Again, the architect may partition the system in this manner to simplify component replacement and manage module failure rate estimates.

Conceptually, both of these approaches are an extension of the on-card control plane model that PCIe supports so well. The long reach and low pin count of the interconnect make this approach even more attractive. It should be noted that this paradigm also applies to cards that hold relatively intelligent data plane elements such as I/O processors, digital media adapters (DMAs), NPUs, and ASICs. The test is how much general-purpose processing capability such an element provides on the module. In many of these types of systems, the designer will use a PCI Express switch on the local module to fan out the PCIe connection to as many devices as the system requires.

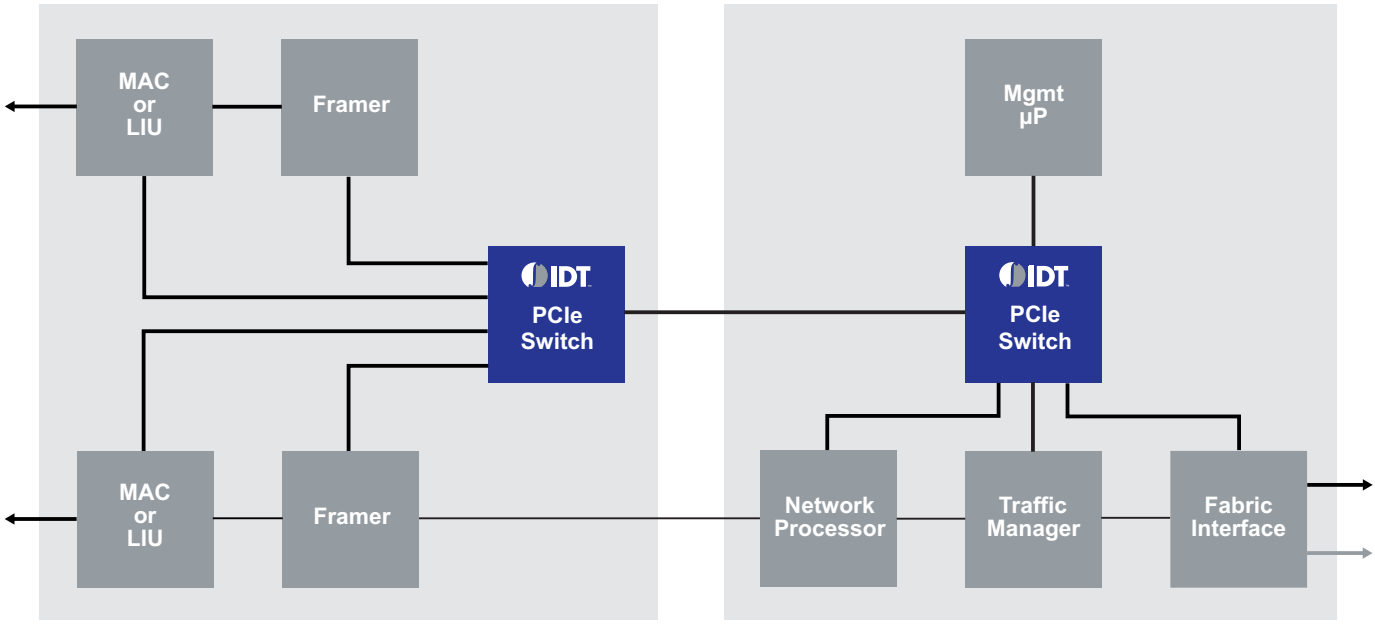


Figure 5. PCIe control plane in modular line interface card

CENTRAL PROCESSING ARCHITECTURE MODEL

In devices such as access and aggregation nodes, designers can manipulate data moving through the system with a single or a small number of data processing elements. This approach allows the designer to use a highly centralized architecture and to keep individual line interface cards very simple (see figure 6). Typically, these systems use a centralized control processor, since lower data manipulation complexity usually reduces the complexity of the control system.

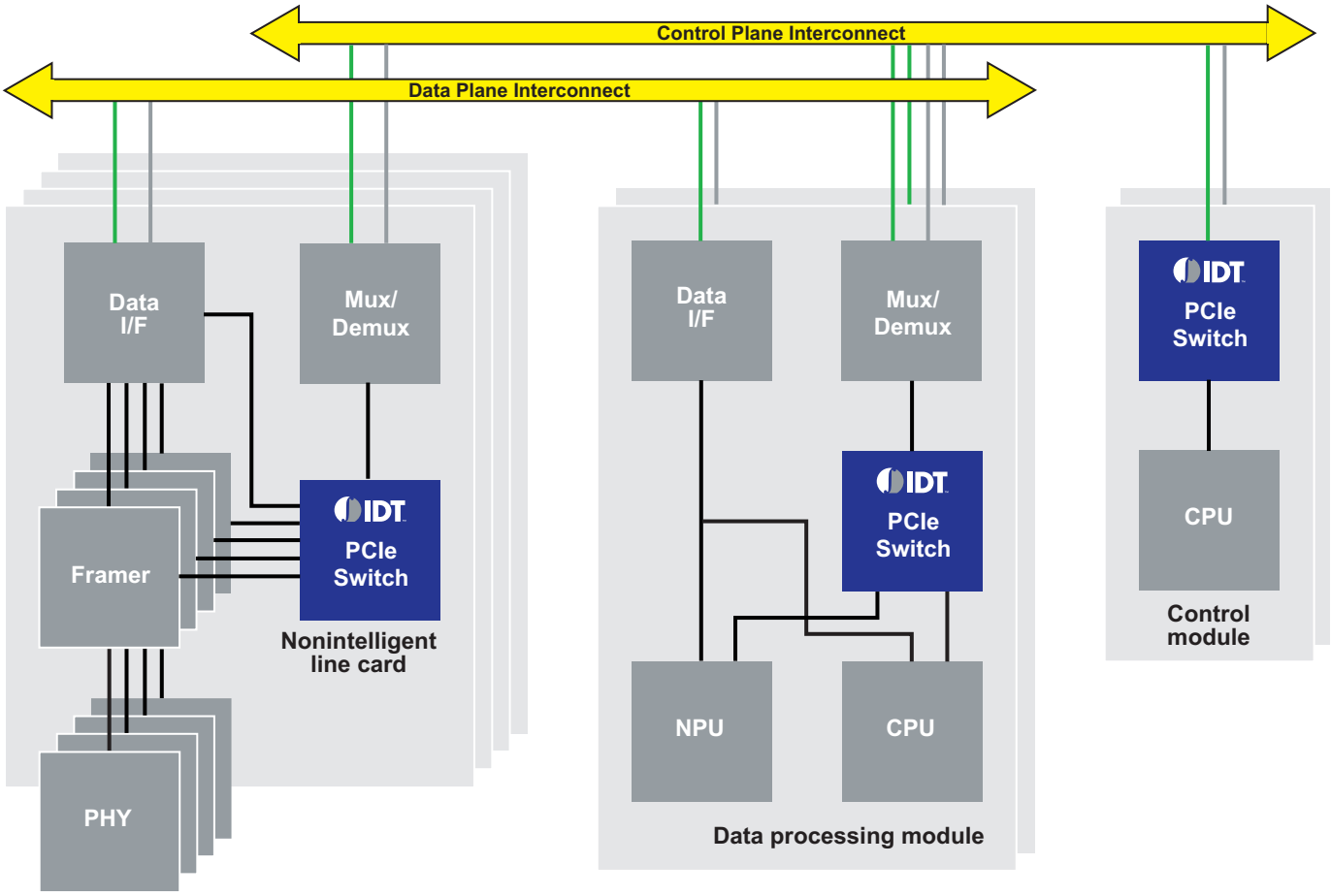


Figure 6. Central processing architecture

Secondary or niche PCIe applications

Systems in the first tier of enterprise networks and in the access tier of wide area networks (WANs) often employ only a small number of line interface protocols on their downstream (user-facing) ports and one protocol on their upstream port(s). This decision is often dictated by tight cost constraints that place limits on the system's complexity or by the physical reach limits of the protocol the system supports. Such systems will maintain the data in as close to its native form as possible and perform most translation and data processing on the uplink card (see figure 6). PCIe will usually not be used in these types of systems.

However, in applications that interwork multiple protocols, such as in aggregation and edge nodes, there is no obvious format for internal data transfers. Interworking implementations logically terminate the connection on the ingress interworking device (usually a CPU, NPU, or ASIC) and then generate a new connection to the egress interworking device, which in turn terminates and generates its own new connection on the output port.

On-card data planes tend to use point-to-point protocols based around blocks of data and do not need routing headers. Usually designers employ a standards-based physical or transport protocol and perform higher layer functions with a proprietary protocol on either end of the link. One example of this approach would be the use of XAUI physical specs, MAC-based framing, and proprietary packet headers between two ASICs. The nonstandardized nature of these protocols is a significant liability.

As communications systems have evolved toward providing services to the user instead of the simple transport of data, designers have introduced an increasing number of compute-type elements into their systems. These compute elements are no longer limited to peripheral functions like management and billing. Today, soft-switches, IP multimedia subsystems (IMS), fixed-mobile convergence (FMC) systems, session initiation protocol (SIP) proxy servers, and VoIP messaging subsystems insert compute elements into the data path of key communications services. These systems are essentially servers with specialized I/O modules. Some of these functions are being addressed by telecommunications-focused blade server systems. A number of low-end routers are also built in this way and use PCI as their interconnect.

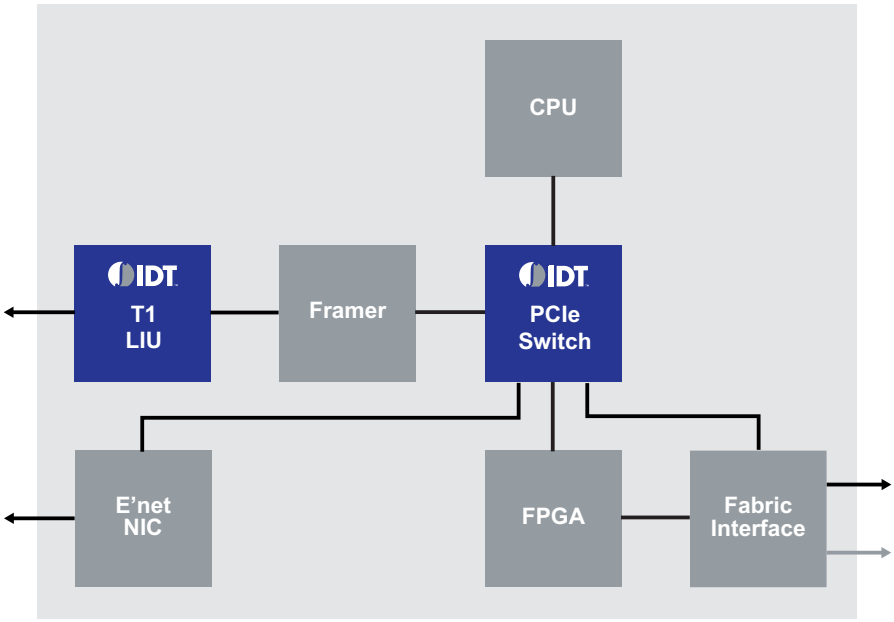


Figure 7. PCIe in an on-card data plane application

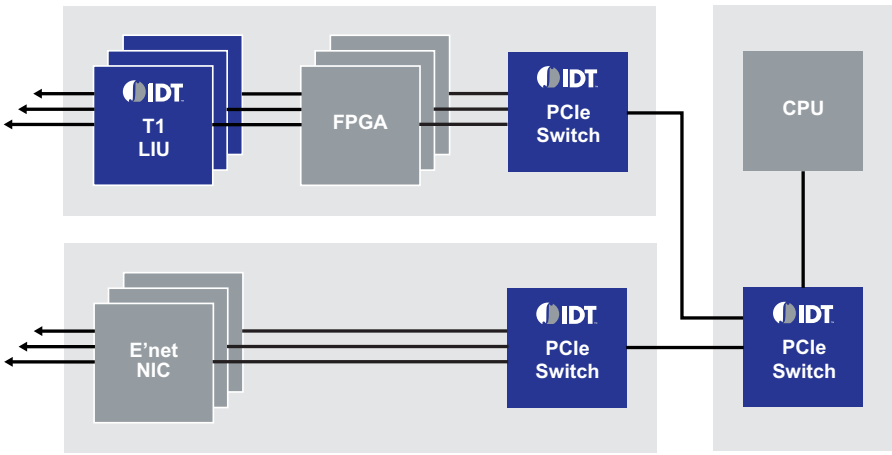


Figure 8. PCIe in a low-end router

This trend is likely to result in the development of a growing number of chips with communications protocols on the line side and PCIe on the internal side to mesh more closely with these new compute-type structures used in network elements.

MERGED CONTROL AND DATA PLANE ARCHITECTURE

Given the high availability requirements of most communications systems, functions controlling the system must have access to every module in the system regardless of what is happening on the data plane. Historically, designers have addressed this need by using physically separate pathways across the backplanes of their systems or within highly complex individual cards (see figure 6 and figure 9).

PCs and servers have traditionally used a single plane for both control and data movement functions. While there remains a high degree of commitment in the communications industry to physically separate control and data planes, this merged control and data plane architecture may become increasingly acceptable for several reasons:

1. Backplane switches are now capable of maintaining traffic separation (security).
2. Backplane switches can guarantee high-priority traffic will get through even if there is a denial-of-service attack on lower priorities (guaranteed access).
3. As PC architectures penetrate further into the communications segment, there will be more system and chip architectures that don't support the separation (supply of components).
4. The cost contribution of maintaining two fabrics, each with its own redundant elements (fabric management SW) and mean time between failures (MTBF), is high.

Backplane switching control plane between smart cards

The distributed processing model (see figure 9) has always been the architecture of choice for larger communications systems, such as edge and core nodes, and it is becoming increasingly common in access nodes. In this architecture, each card's address domain remains separate so that the same software can be run on multiple instances of the same card type without changes.

The more complex data processing demands of edge and core nodes usually require that each line interface card perform most or all of the processing of its own data. Given the number and complexity of the devices present on each card, these cards usually feature their own housekeeping processor. In many cases, these processors also handle any exception cases referred from the data path. The presence of these processors on most or all of the line interface cards changes the nature of control plane communications across the backplane. It is now based on an interprocessor communications model, which usually involves message passing and interrupts, rather than direct register access.

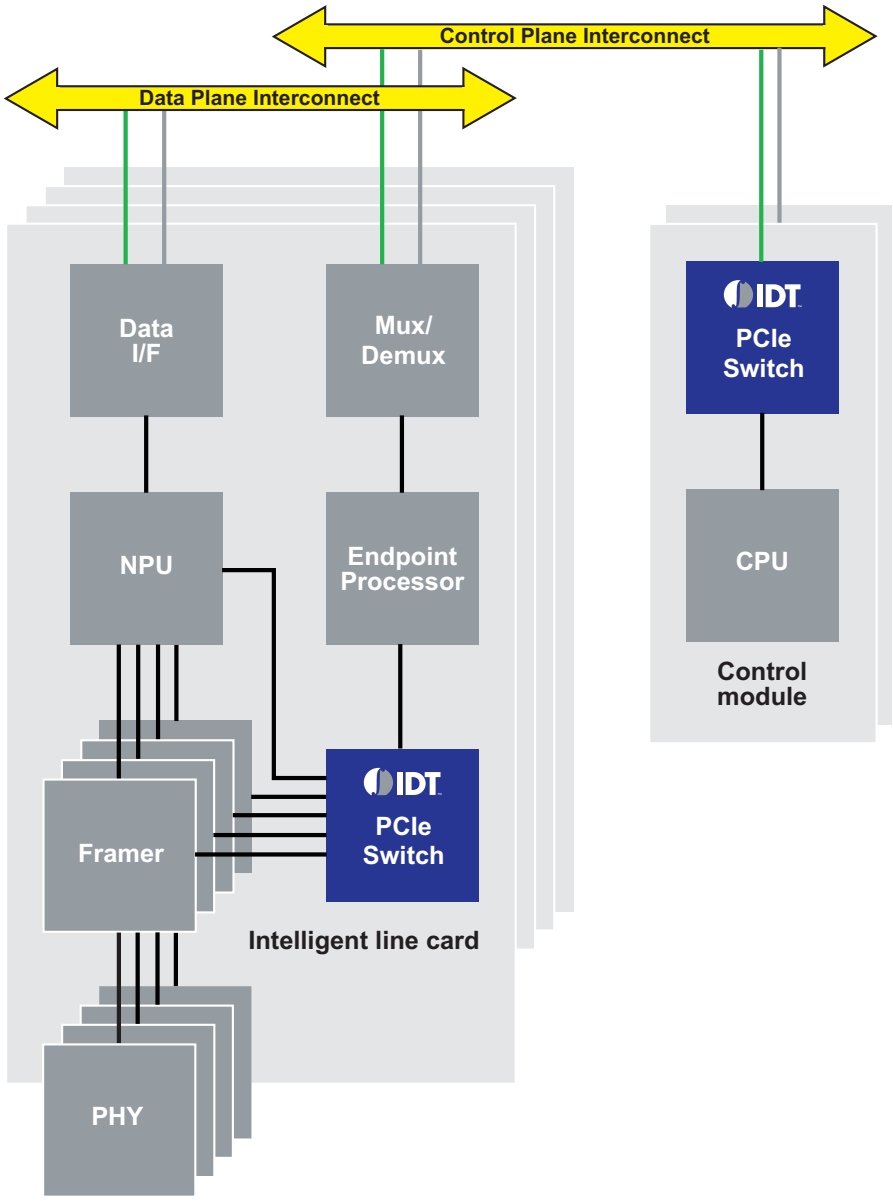


Figure 9. Distributed processing architecture

Within this model, the housekeeping processor on each card is responsible for management of all the devices local to that card (and any subtended I/O modules). It is also responsible for management of all messages exchanged between the processors and other housekeepers processors on other cards as well as for communication between the card and the main control processor in the system.

There are a number of approaches to this type of communication. One approach, which will not be discussed here, is to treat each card as an independent node in a network and communicate between cards using an address-independent networking protocol such as Ethernet. While workable, these approaches can add considerable unnecessary overhead within a chassis, such as network topology discovery.

Within the PCI family, interdomain communication can be facilitated by the use of a function that will isolate the topology and addressing scheme of the domain behind one of its ports from the details of domains behind its other ports. These functions are widely referred to as nontransparent bridges. This function can be implemented within an embedded processor or separately. When it is integrated with a processor, the resulting device is called an endpoint processor.

Interprocessor communication via endpoint processors

A detailed discussion of how to implement this may be found in the IDT white paper, “Enabling Multi-peer Support with a Standard based PCI Express Multi-ported Switch.”

An increasing number of processors, such as the Freescale MPC854xE family, and endpoints, such as the Intel IOP80332/3, are now capable of supporting address remapping for their transactions. With these devices, designers can now use a memory-based addressing scheme to move data directly from the memory of one module into the memory of another, even though they use separate addressing domains.

This technique allows designers to use PCIe, with its memory-based addressing scheme, as a systemwide interconnect for control plane transactions. This approach offers several advantages over solutions in use today. First, it simplifies the job of hardware and software designers by eliminating the need for them to learn a proprietary protocol. It also simplifies the system design by eliminating the need to translate packet encoding, priority classifications, and flow control between two protocols. Finally, the use of PCIe as a systemwide interconnect for control plane transactions also promises to reduce transfer latency and memory bandwidth requirements by removing translations and multiple handling of data inherent in message queuing (see box on the multiple copy bottleneck). Finally, a memory-based interconnect scheme can enhance diagnostic capability through the main control CPU by supporting direct access to devices on the local card and the placement of predetermined patterns into memory structures on a card under test.

Interprocessor communication via stand-alone nontransparent bridges

A nontransparent PCIe bridge device appears to the domain on each port as a PCIe endpoint. It consumes any transactions directed to it and generates new, altered transactions on its other port for many of these transactions. The manipulations for altering those transactions are not standardized, nor are the registers for setting them up, although most component suppliers offer fairly similar transpositions and other interprocessor communications services, such as scratchpad registers, doorbell interrupts, etc.

A detailed discussion of how to implement this may be found in IDT App Note AN-510, “Usage of Nontransparent Bridging with IDT PCI Express NTB Switches.”

MULTIPLE COPY BOTTLENECK

The server and storage markets have historically relied on the use of multiple interconnected, general-purpose CPUs to perform the necessary manipulations on data, rather than the fixed-function devices prevalent in the communications world. While this paper will not debate the efficiency of using CPUs for data manipulation, it is clear that techniques for efficient movement of data between those devices (up to 128 CPUs in some storage applications) have been examined in detail by architects in that space.

Recent research and development efforts have focused on resolving this "multiple copy" bottleneck. This problem is best illustrated with a sample data flow. If a block of data needs to be transmitted from an application running on one CPU to an application running on another over a message-based transport such as Ethernet, the data is often copied six or more times in the process:

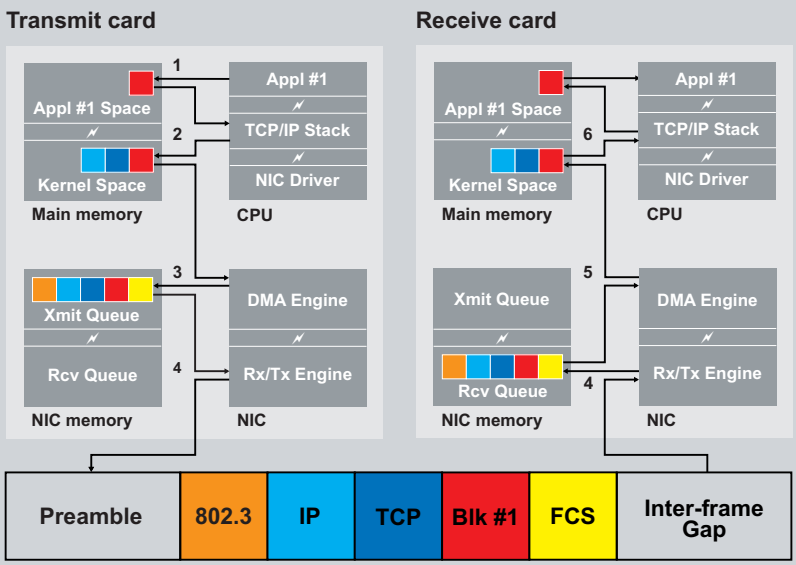


Figure 10. Ethernet NIC multiple copy flowchart

1. A message is created and queued in application memory space.
2. Data is copied from application memory space to a message queue in protected kernel memory. As it is copied, it will be encoded in some sort of message frame (such as TCP/IP), with a header and checksum.
3. The data is then copied into memory in the transmit queue of the Ethernet NIC, usually by the DMA engine within the NIC, and segmented into Ethernet frames with their headers and frame checksum.
4. Once at the head of the transmit queue, it is read from NIC memory, transmitted over the Ethernet network, and copied into a receive buffer at the destination Ethernet NIC.
5. The receive NIC's DMA will then move the data block into protected kernel space on the destination node, stripping the Ethernet framing as it does so.
6. The kernel software will then check the received data block for integrity and check its headers to determine its destination application. It will move it into the memory space of the destination application, where it may join a queue of previously received messages.

These multiple copies are fairly typical of data movement in many Ethernet-based servers. Several of the steps are necessitated by the need to maintain separate memory spaces for application and kernel. But many are introduced to support multiple layers of framing, sequencing, and protection for a message-based interconnect. These redundant processes may be necessary in a box-to-box transmission, where error rates and the possibility of outside interference make multiple security layers prudent. However, in the constrained environment of an intrasystem interconnect, these levels of protection are unnecessary.

Recently designers of server and storage systems have moved towards a shared-memory model, usually with address translation. In this model, applications pass data pointers to intelligent I/O processors that move the data directly into the memory for the destination application, translating the addresses as each word of data is moved. While this description ignores exchanges of pointers between the various processing elements, the data block itself is only moved once.

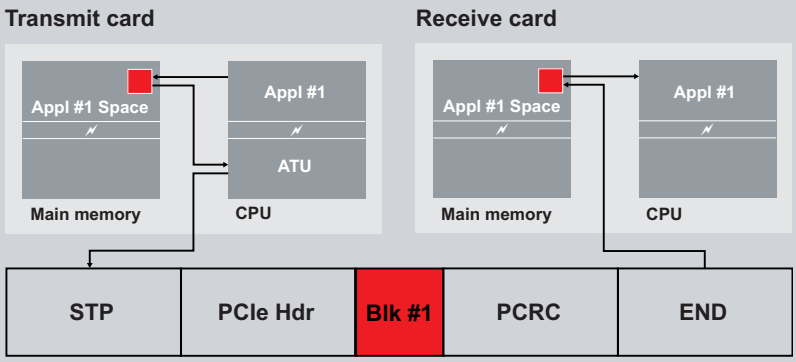


Figure 11. PCIe direct copy flowchart

This particular model of data movement uses a memory–address based interconnect scheme, rather than a message-based one. This is why blade server systems are moving towards PCIe backplanes, and away from Ethernet-based ones. As system architects become better acquainted with the efficiency and lower latency advantages of this data movement scheme, a similar scheme may be used in communications systems.

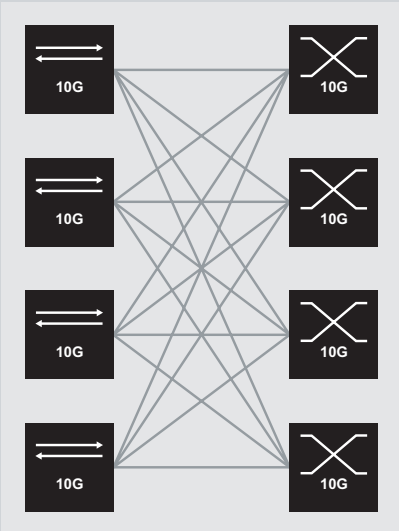
Redundancy and PCIe

Any discussion of backplane switching in a communications environment has to address the topic of redundancy. This paper has previously described a number of the features in PCIe that help monitor system operation, including advanced error reporting. Support for operation at reduced bandwidth in the face of a single lane failure in a PCIe link was also discussed. However, what can be done in a PCIe system that requires a module to be switched-out?

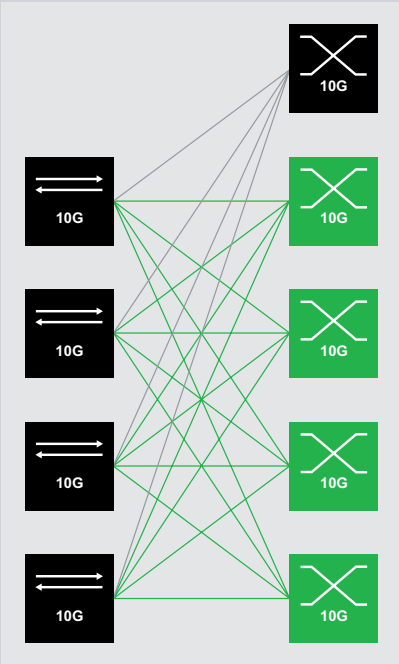
Any redundancy scheme (see the inset below for a description of the models available) involving PCI Express must address the specification's rule that only allows one port of a PCI Express switch or bridge to be used as the upstream port. This rule was designed to prevent inadvertent or malicious reconfiguration of PCIe switches and bridges by peripheral devices.

REDUNDANCY MODELS

Typically, communications systems use any of four models for redundancy. The diagrams below illustrate how these different models could be used in a simple system that supports 4 x 10 Gbps line cards. In this figure, the green switch elements are active and the grays are in standby mode.

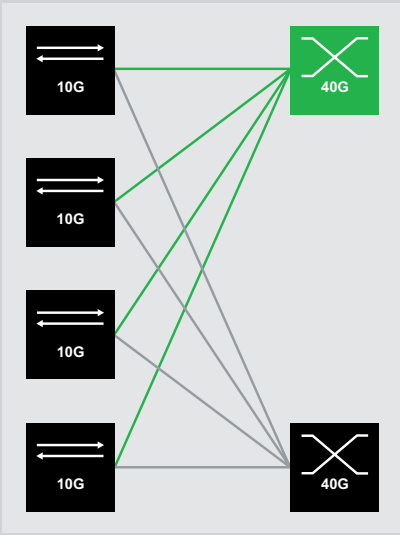


N-1 Redundancy: Also known as "load sharing" or "graceful degradation," this scheme spreads the load equally across all available elements. The number of available elements is sized to exactly match the peak capacity required in the system. Failure of any shared element will result in a loss of capacity for the system, but not any loss in connectivity. A variation of this model spreads the load across more elements than are required for the nominal capacity, so that the loss of one element brings the system down to its nominal capacity with no loss of capability.

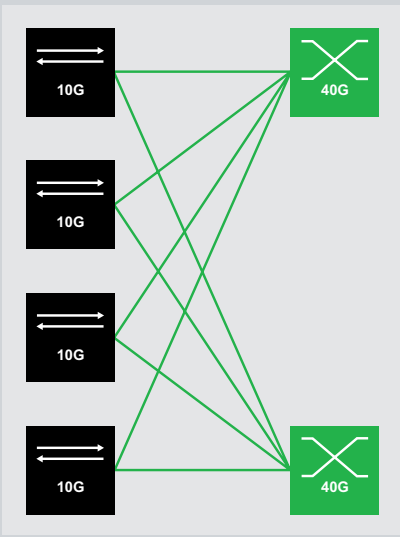


N+1 Redundancy: This approach is similar to the N-1 model in that all normal traffic is shared across a number of elements equal to the nominal system capacity. However, this approach adds an extra element, unused during regular operation, to take up the loss. Some system monitoring function must be present in this model to detect the failure and to signal all the line cards to switch over to the redundant element.

REDUNDANCY MODELS *(continued)*



1:1 Redundancy: Also known as "warm standby," this strategy provides a complete backup of the function being protected. Upon detection of a failure, the line cards are directed to route all traffic to the backup. Like N+1, this system has the drawback of losing all "in-flight" data since the switching is done at the source.



1+1 Redundancy: Called "hot standby," this technique also provides a complete backup of the function being protected, but line cards send data to both switching elements at all times. On receipt, they determine which element to pay attention to and which to drop redundant data from. Once a failure is detected, the receiving line cards switch the element they pay attention to. This approach eliminates any potential loss of "in-flight" data since the switch occurs at the destination. However, it can be tricky to ensure no loss or duplication of data on switchover if one path is faster than the other

When the application involves switching out of a failed downstream element, both the N+1 and 1:1 redundancy models can be supported simply by reprogramming the switch to redirect the traffic from the failed element to the standby element. Some examples of this approach include a line processing card talking to several I/O modules or replacement of a line card in a central processing architecture system.

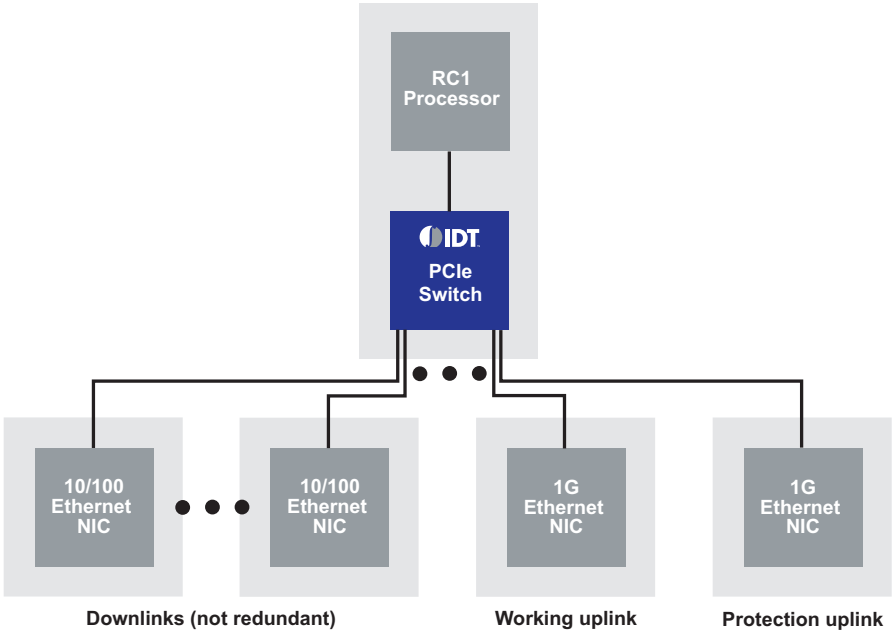


Figure 12. Downstream 1:1 redundancy in PCIe

If a situation involves the replacement of a failed upstream element, such as a central control or processing card or a switch fabric element, then the downstream elements must be fooled into believing they are still receiving commands and data from the same upstream element. This can be accomplished by using a 2:1 mux designed for PCIe signals to indicate which of two upstream elements will actually drive traffic into the downstream element. This structure can support only the 1:1 redundancy model.

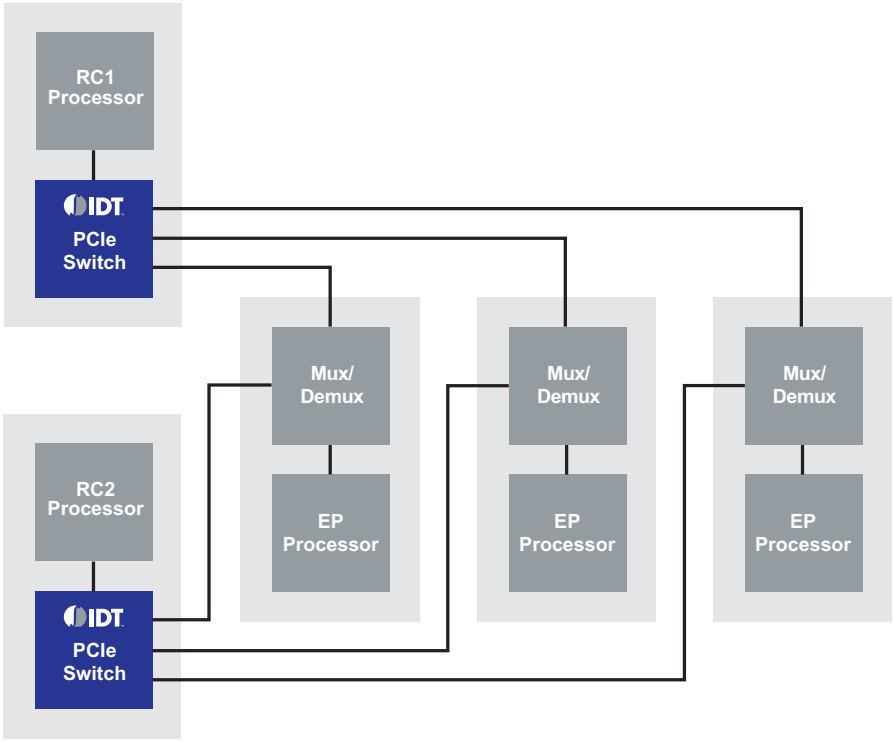


Figure 13. Upstream 1:1 redundancy in PCIe

Backplane switching data plane

Data plane movement between cards in communications systems usually uses a proprietary message-passing scheme implemented using ASICs or off-the-shelf chip sets. Similar to control plane transactions, data blocks are moved into separate physical memory locations for queuing on both transmitter and receiver ends of the transfer. In such an environment, they can benefit from a direct-memory transfer scheme implemented with a memory-based interconnect such as PCIe.

It should be noted that this scheme is useful only when working with devices that store and retrieve data blocks in memory. This approach would not be directly applicable to devices with "streaming" interfaces, which constitute the bulk of communications-focused data plane elements today.

Many designs in the PC and server arenas are now moving to schemes such as these. An increasing number of blade server systems now use PCIe interconnects instead of Ethernet as a backplane. As more server-type boxes migrate into communications networks, these concepts will become increasingly acceptable to communications equipment designers.

Interchassis interconnect

Many communications systems at the high end of the performance range either require multiple chassis or scale up from single to multichassis configurations as port counts or speeds rise. Typically, the protocol used to communicate between the systems is a cable-based or optical version of the protocol used to carry the same traffic within the system. Often one data plane protocol and another control plane protocol are extended across interchassis links. In systems where PCIe is used internal to the chassis, the interconnect can be readily extended using cable PCIe to link multiple chassis elements into a single logical system at distances up to 7m.

Conclusion

Given its ability to extend signal reach and reduce pin count, PCIe will quickly become the CPU port protocol of choice. The availability of multiple CPUs with PCIe ports, including the CPU and housekeeper types favored in communications systems, will lead to a rapid adoption of PCIe in the on-card control plane. Wherever a CPU on one card controls devices directly on another card, PCIe be used to extend the control protocol across the backplane as a natural extension of the on-card model.

While communications systems designers will continue to use separate data and control planes both on and off-card, over time there will be some bleedover of components from the server and storage worlds where only a single plane exists. As communications networks increase usage of compute-type elements for applications such as soft switches and IP multimedia subsystems (IMS), this trend will accelerate. Eventually, the communications arena will see a growth of islands of devices that use PCIe as their primary protocol for data plane purposes.

Data planes between cards and control planes between intelligent cards in communications systems typically use a message-passing paradigm that has been supported by Ethernet or proprietary backplane fabric protocols. In intelligent cards, this approach is driven in large part by the need to isolate the address spaces on each card from one another. However, as designers migrate to main and I/O processors with embedded address translation capability, they will be able to achieve this goal by using a shared memory model of data transfer. This shared memory model can provide higher data transfer efficiencies and lower latencies than an application-toapplication viewpoint by eliminating the multiple-copy bottleneck present in most message-passing models. This, in turn, will further drive the development of islands of PCIe-based equipment across the communications landscape.

Glossary

ADM	Add-drop multiplexer
ASIC	Application specific integrated circuit
ATM	Asynchronous transfer mode
BLC	Broadband loop carrier
CPU	Central processing unit
CRC	Cyclic redundancy check
DLC	Digital loop carrier
DMA	Digital media adapter
DSL	Digital subscriber line
DSLAM	Digital subscriber line access multiplexer
DWDM	Dense wave-division multiplexing
FMC	Fixed mobile convergence
Gbps	Gigabits per second
GPON	Gigabit passive optical network
IDT	Integrated Device Technology, Inc.
IEEE	Institute of Electrical and Electronics Engineers
I/F	Interface
IMS	IP multimedia subsystem
I/O	Input-output
LIU	Line Interface Unit
MAC	Media access controller
MSPP	Multiservice provisioning platform
MTBF	Mean time between failures
NIC	Network interface card
NPU	Network processing unit
OADM	Optical add-drop mux
OLT	Optical line terminal
PCB	Printed circuit board
PC	Personal computer
PCI	Peripheral component interconnect
PCIe	Peripheral Component Interconnect—Express version
PCMCIA	Personal Computer Memory Card International Association
PHY	Physical layer of open system interconnection model
QoS	Quality of Service
ROADM	Reconfigurable optical add-drop mux
RT-DSLAM	Remote terminal—digital subscriber loop access multiplexer
RX/TX	Receive/transmit
SAN	Storage area network
SCSI	Small computer system interface
SIG	Special interest group
SIP	Session Initiation Protocol
SONET	Synchronous optical network
STP	Signal transfer point
SW	Software
Tbps	Terabits per second
TCP/IP	Transmission Control Protocol/Internet Protocol
VDSL	Very high speed DSL
WAN	Wide area network
XAUI	Xilinx 10 Gigabit Attachment Unit Interface

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.