HyperTransport Over Ethernet - A Scalable, Commodity Standard for Resource Sharing in the Data Center

Jeffrey Young, Sudhakar Yalamanchili*
Georgia Institute of Technology
jyoung9@gatech.edu, sudha@ece.gatech.edu

Brian Holden, Mario Cavalli
HyperTransport Consortium
{brian.holden, mario.cavalli}@hypertransport.org

Paul Miranda
AMD
paul.miranda@amd.com

Abstract

Future data center configurations are driven by total cost of ownership (TCO) for specific performance capabilities. Low-latency interconnects are central to performance, while the use of commodity interconnects is central to cost. This paper reports on an effort to combine a very high-performance, commodity interconnect (HyperTransport) with a high-volume interconnect (Ethernet). Previous approaches to extending HyperTransport (HT) over a cluster used custom FPGA cards [5] and proprietary extensions to coherence schemes [22], but these solutions mainly have been adopted for use in research-oriented clusters. The new HyperShare strategy from the HyperTransport Consortium proposes several new ways to create low-cost, commodity clusters that can support scalable high performance computing in either clusters or in the data center.

HyperTransport over Ethernet (HToE) is the newest specification in the HyperShare strategy that aims to combine favorable market trends with a high-bandwidth and low-latency hardware solution for non-coherent sharing of resources in a cluster. This paper illustrates the motivation behind using 10, 40, or 100 Gigabit Ethernet as an encapsulation layer for HyperTransport, the requirements for the HToE specification, and engineering solutions for implementing key portions of the specification.

1. Introduction

HyperTransport interconnect technology has been in use for several years as a low-latency interconnect for processors and peripherals [9] [7] and more recently as an off-chip interconnect using the HTX card [5]. However, HyperTransport adoption for scalable cluster solutions has typically been limited by the number of available coherent connections between AMD processors (8 sockets) and by the need for custom HyperTransport connectors between nodes.

The HyperTransport Consortium's new HyperShare market strategy has presented three new options for building scalable, low-cost cluster solutions using HyperTransport technology: 1) HyperTransport-native torus-based network fabric using PCI Express-enabled network interface cards implementing the HyperTransport High Node Count specification [14], 2) HyperTransport encapsulated into InfiniBand physical layer packets, and 3) HyperTransport encapsulated into Ethernet physical layer packets. These three approaches provide different levels of advantages and trade-offs across the spectrum of cost and performance. This paper describes the encapsulation of HyperTransport packets into Ethernet, thereby leveraging the cost and performance advantages of Ethernet to enable sharing of resources and (noncoherent) memory across future data centers. More specifically, this paper describes key aspects of the HyperTransport over Ethernet (HToE) specification that is part of the HyperShare strategy.

In the following sections, we describe 1) the motivation for using HToE in both the HPC and data center arenas, 2) challenges facing the encapsulation of HT packets over Ethernet, 3) an overview of the major components of this specification, and 4) use cases that

*This research was supported in part by NSF grant CCF-0874991, and Jeffrey Young was supported by a NSF Graduate Research Fellowship

demonstrate how this new specification can be utilized for resource sharing in high node count environments.

2. The Motivation for HToE: Trends in Interconnects

The past ten years in the high-performance computing world have seen dramatic decreases in off-chip latency along with increases in available off-chip bandwidth, due largely to the introduction of commodity networking technologies like InfiniBand and 10 Gigabit Ethernet (10GE) from companies such as Myrinet and Quadrics. Arguably, InfiniBand has made the most inroads in the high-performance computing space, with InfiniBand composing 42.6% of the fabrics for clusters on the current Top 500 Supercomputing list [18].

At the same time, Ethernet has evolved as a lower-cost and "software-friendly" alternative that enjoys higher volumes. The ability to integrate HT over Ethernet would enjoy significant infrastructure and operating cost advantages in data center applications and certain segments of the high-performance marketplace.

2.1. Performance

The ratification of the 10 Gigabit Ethernet standard in 2002 [1] has led to its adoption in data centers and the high-performance community. Woven Systems (now Fortinet) in 2007 demonstrated that 10 Gigabit Ethernet with TCP offloading can compete in terms of performance with SDR InfiniBand, with both fabrics demonstrating latencies in the low microseconds during a Sandia test [28]. In addition, switch manufacturers have built 10 Gigabit Ethernet devices with latencies in the low hundreds of nanoseconds [11] [31]. Recent tests with iWARP-enabled 10GE adapters have shown latencies that are on the order of 8-10 microseconds, as compared to similar InfiniBand adapters with latencies of 4-6 microseconds [12]. More recent tests have confirmed that 10 Gigabit Ethernet latency for MPI with iWARP is in the range of 8 microseconds [20].

These latencies already are low enough to support the needs of many high-throughput applications, such as retail forecasting and many forms of financial analysis which typically require end-to-end packet latencies in the range of a few microseconds. The new IEEE 802.3ba standard for 40 and 100 Gbps Ethernet also aims to make Ethernet more competitive with InfiniBand. Although full-scale adoption is likely to take several years, there are already some early products that support 100 Gigabit Ethernet [25].

The challenge with using these lower-latency fabrics is in making these lower hardware latencies accessible to the application software layers without having to engage higher overhead legacy software protocol stacks that can add microseconds of latency [4] [23]. The HToE specification described here is a step towards that goal, since it focuses on using Layer 2 (L2) packets and a global address space memory model to reduce dependencies on software and OS-level techniques in performing remote memory accesses.

2.2. Cost and Market Share

While 10 Gigabit Ethernet has had a relatively slow adoption rate in the past few years, it should be noted that 1 Gigabit and 10 Gigabit Ethernet still have a 45.6% share of the Top 500 Supercomputing list [18], with a majority of these installations still using 1 Gigabit Ethernet. This indicates that cost plays an important role in the construction of computational clusters on this list (for example, for market analysis and geological data analysis in the mineral and natural resource industries). Additionally, networks composed of 1 and 10 Gigabit Ethernet also have a dominant position in high-performance web server farms. Part of this widespread market share is due to the low cost of Gigabit Ethernet and falling cost of 10 Gigabit Ethernet as well as the management and operational simplicity of Ethernet networks.

However, it should also be noted that InfiniBand still enjoys a price and power advantage over 10 and 40 Gbps Ethernet due to being first to market. A 40 Gbps, 36 port InfiniBand switch now costs around $6,500 and has a typical power dissipation of 226 Watts [8], while a 10 Gbps, 48 port Ethernet switch costs around $20,900 and has a power dissipation of 360 Watts.

One of the strongest factors for using Ethernet is the trend toward converged networks, driven in large part by the need to lower the total cost of ownership (TCO). For example, Fibre Channel (FC) has been the de facto high-performance standard for SANs for the past 15 years. The technical committee behind FC has been a major proponent of convergence in the data center with their introduction of the Fibre Channel over Ethernet (FCoE) standard [15]. This standard relies on several new IEEE Ethernet standards that are collectively referred to as either Data Center Bridging (DCB) or Converged Enhanced Ethernet (CEE) and are described in more detail in Section 3.3. The approval of this standard and subsequent adoption by hardware vendors bodes well for the continued usage of Ethernet in data centers and smaller high-performance clusters.

Possibly one of the best indicators of the future market share for Ethernet as a high-performance data center and cluster fabric is the willingness of competitors to embrace and extend Ethernet technologies. Two examples are the creation of high-performance Ethernet switches [24] and the development of RDMA over Converged Ethernet (RoCE) [3], which has been referred to by some as "InfiniBand over Ethernet" since it utilizes the InfiniBand verbs and transport layer with a DCB Ethernet link layer and physical network.

2.3. Scalability

As the most prevalent commodity interconnect technology in previous generation data centers, there has been considerable effort devoted to constructing scalable Ethernet fabrics for data centers. For instance, consider the use of highly scalable fat tree networks for data centers using 10 Gigabit Ethernet [27], while network vendors have already embraced the in-progress standards for Data Center Bridging as a way to create converged SANs and a high performance cluster fabric [21]. Other recent studies have demonstrated techniques for active congestion management to enable further scaling of topologies constructed around Ethernet [28]. We can expect to see continued efforts toward expanding the use of Ethernet in an effort to leverage legacy software, existing expertise in the networking-related workforce, and volume cost-related advantages.

2.4. The Case for HToE

As the previous sections have shown, Ethernet has significant benefits in the areas of cost, market share, and competitive performance. HyperTransport over Ethernet shares these benefits while adding the advantage of a transparent on-package to off-package encapsulation using 10, 40, and 100 Gbps Ethernet. The IEEE 802.3ba standard also includes support for short-reach (7 meter) copper cable physical layers for 40 and 100 Gigabit Ethernet, which should allow for more cost-effective implementations of 40 and 100 Gigabit Ethernet. As the penetration of these new flavors of Ethernet grows, the potential for HyperTransport over Ethernet also grows as a high-performance hardware communication and sharing mechanism. In fact, this capability for improved resource sharing is one of the best motivators for using HToE and is discussed in more detail in Section 5.

HyperTransport over Ethernet also addresses a different market space than that served by the HyperTransport High Node Count specification and HyperTransport over InfiniBand. Specifically, HToE is well suited for creating scalable, low cost clusters that rely on a converged Ethernet fabric to share resources in a non-coherent fashion. Ethernet's market share ensures that the barrier to entry in using HyperTransport over Ethernet is low in most cases, and using converged Ethernet negates the need for a custom sharing fabric like NUMAlink [16] or additional cabling for an InfiniBand or other custom network.

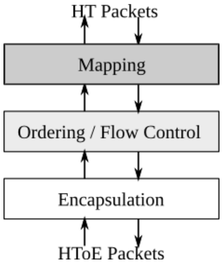

Figure 1. **HyperTransport Over Ethernet Layers**

3. HToE Specification Requirements

Due to the differences between the point-to-point communication of HyperTransport and the switched, many-to-many communication of Ethernet, the HyperTransport over Ethernet specification needs to address several key requirements to ensure correct functionality. To manage the traversal of packets between these fabrics, we focus on a bridged implementation using encapsulation of HT packets (typically up to 64 Bytes of data) in larger Ethernet packets (up to 1500 Bytes or larger in some cases). If we are to remain faithful to end-to-end HT transparency at the software level, the requirements of the HT protocol now translate into requirements for Ethernet transport that are realized in Layer 2 switches.

Furthermore, to productively harness the capabilities of HToE, it must be implemented in the context of a global system model that defines how the system-wide memory address space is deployed and utilized. Toward this end, we advocate the use of global address space models and specifically the Partitioned Global Address Space (PGAS) model [30]. In particular, we are concerned about the portability of the model and application/system software across future generations of processors with increasing physical address ranges.

To illustrate the differences between HyperTransport and HToE and to help illustrate how HToE supports global address models, we have divided the core functionality of HToE into three "layers": the "mapping" layer, the "ordering and flow control" layer, and the "encapsulation" layer, as shown in Figure 1.

3.1. On-package and Off-package Addressing

HyperTransport address mapping allows for I/O devices and local DRAM to be mapped to physical addresses that are interpreted by the processor for read and write operations. This physical address mapping is hidden from applications using standard virtual addressing techniques in the operating system.

HToE supports a global, system-wide, noncoherent address space. Addresses must be transparently recognized as either local or remote, and the latter must be mapped to memory or device addresses on a remote node. Implicitly, this mapping must translate between address spaces and Ethernet MAC addresses and vice versa. Consequently, this mapping between local HT addresses and the global HToE address space is necessary to encapsulate and transmit HT packets from a local node to a remote node's memory. Additionally, the remote node must not require modification to its local HyperTransport links in order to route packets that have been sent from a remote node – that is, any remote requests must appear to the local HT link as an access by a local device to a local address. For more details on the specific mapping used by HToE, see Sections 4.2 and 4.3.

3.2. Scaling HyperTransport Ordering and Flow Control for Cluster Environments

HyperTransport is a point-to-point protocol that uses three virtual channels to send and receive command and data packets. The HT protocol has been designed to ensure that packet ordering on these channels is preserved on local links via the HT Section 6 Ordering algorithm [7]. This algorithm ensures not only that packets arrive in a logical order but also that deadlock freedom is ensured. In a switched Ethernet environment with the possibility of packet loss, preservation of ordering becomes a much more difficult problem. Thus, our HToE solution must ensure that packets remain ordered correctly within their virtual channels. For more information on maintaining order, see Section 4.4.

In addition to packet ordering, the HT 3.1 specification also defines a multi-channel, credit-based flow control algorithm. Credits typically flow between two point-to-point links based on the receipt and processing of packets within each virtual channel. In a scalable, switched Ethernet environment, packets could conceivably flow from multiple sources to one destination. Furthermore, since HyperTransport packets are much smaller than Ethernet packets, another requirement is that multiple HyperTransport packets can be encapsulated in one Ethernet packet to reduce the overhead of encapsulation. Both of these requirements indicate the need for a careful rethinking of how to send HyperTransport credits and packets when using HToE. The requirement is that the sender must possess credits for all HyperTransport packets that it encapsulates. HyperTransport packets that are encapsulated in a single Ethernet packet must be of the same virtual circuit and headed for the same destination.

3.3. The Benefit of a Congestion-Managed Ethernet Network for Flow Control

One recent development that was investigated for this specification was the introduction of several IEEE specifications, collectively known as Data Center Bridged (DCB) Ethernet or sometimes Converged Enhanced Ethernet (CEE), depending on the company promoting it.

Data Center Bridged Ethernet aims to provide a congestion-managed Ethernet environment to support converged fabrics in the data center and was motivated by the convergence of the Fibre Channel standard onto Ethernet fabric, aka FCoE [29]. These fabrics aim to prevent packet loss due to congestion but do not prevent packet loss due to bit errors or other sources such as equipment failure or fail-over. Data Center Bridged Ethernet incorporates several specifications including per-priority flow control (IEEE 802.1Qbb), congestion notification (IEEE 802.1Qau), and Data Center Bridging Capabilities Exchange Protocol and Enhanced Transmission Selection (IEEE 802.1Qaz) [17]. These congestion-management algorithms are especially helpful in high-performance computing because of the intensely self-similar nature of HPC traffic.

3.4. Recovery from Failures

HyperTransport 3.1 has several methods for recovering from errors. A special "poison" bit can be set in HT response packets to indicate to the source processor or device that an operation failed (e.g., a read failed to complete). This error notification typically is passed upstream to the initial requesting device without any notion of the initial request's address. In addition, HyperTransport can use the HT 3.1 retry mechanism to resend packets between source and destination HT devices based on a Go-Back-N algorithm that relies on sequence numbers included in packets. If this mechanism should fail to recover from errors, the host processor has the option to issue a reset using a warm or cold reset that is communicated to devices via separate physical signals.

In the HToE environment, these requirements for

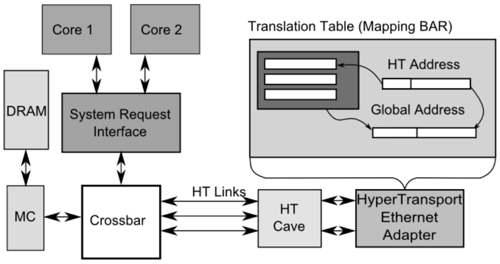

Figure 2. HyperTransport Ethernet Adapter with Opteron Memory Subsystem

recovery from errors become more complex due to the nature of HyperTransport transactions and due to the fact that Ethernet does not support the HyperTransport physical signals. Thus the HyperTransport over Ethernet specification must ensure that 1) errors can be appropriately reported to the requesting remote node, 2) resets can be accurately communicated to remote nodes when otherwise unrecoverable failures occur, and 3) resets for traffic between one source and destination HTEA does not affect traffic from other HTEAs.

3.5. Requirements for Retry in HToE

The HT specification defines a retry mechanism that resends packets when errors are discovered using a Go-Back-N algorithm and sequence numbers for HyperTransport packets. This mechanism must be extended to function over Ethernet and thereby becomes part of the HToE specification. We did not want to rely on TCP's retry algorithm, but Ethernet does not define a Layer 2 error retry protocol. Therefore, we created a variant of HyperTransport 3.1's retry algorithm that would function across an Ethernet fabric in the presence of packet loss due to congestion or due to bit errors.

4. The HyperTransport Over Ethernet Specification

The HyperTransport over Ethernet specification outlines the basic functionality of the HToE bridge device, or HyperTransport Ethernet Adapter (HTEA), that is used to encapsulate HyperTransport 3.1 packets into Ethernet packets. The location of this device in relation to a typical Opteron system is shown in Figure 2. Note that a normal Ethernet MAC can be shared for both HToE traffic and TCP/IP traffic, although the implementer should decide on how to prioritize each traffic type.

To assist with the implementation of each of the specification's requirements, functionality in the HTEA is divided into separate "layers" that are implemented in the hardware of the HTEA and that communicate with other layers when processing incoming or outgoing HT packets. Here we describe some of the more interesting aspects of the "mapping" layer, the "ordering" layer, and the "encapsulation" layer. Full details are available in the HToE specification [32].

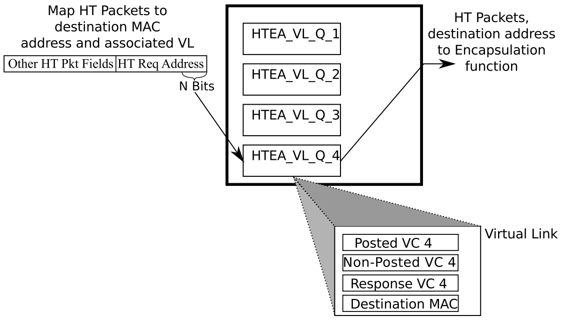

Figure 3. HyperTransport Ethernet Adapter Virtual Link

4.1. HToE's Relationship with DCB

HyperTransport over Ethernet is intended to be used with switches that have been designed for Data Center Bridging environments, such as those explicitly created to support Fibre Channel over Ethernet. However, some of the DCB specifications would interfere with the normal ordering and priority requirements specified by the HT Section 6 Ordering Requirements. For this reason, many of the solutions specified for ordering and flow control do not explicitly require features like per-flow flow control. This means that HToE could likely be supported on normal 10 GE hardware, but it could also be enhanced by allowing for the usage of the DCBX protocol, per-flow priorities (for packet flows between different sources and destinations), and with Enhanced Transmission Selection for usage with other types of network traffic.

4.2. Mapping HT Addresses into the Global Address Space

HyperTransport over Ethernet assumes that the range of memory addresses on each node form a subset of a global, 64 bit physical address space. In order to map the local HyperTransport address to a global memory address, such as those used with some PGAS models [30], and to a destination Ethernet address for remote nodes, a few of the upper bits from the physical address are used to select among potential remote nodes in the mapping layer of the HTEA as shown in Figure 2. This mapping allows for a processor on a local node to make a remote noncoherent "put" or "get" operation to the memory of a remote node.

While the creation of a mapping table is left up to implementers of the HTEA, the selection of global addresses for a particular HTEA and node can be defined using OS-level communication and subsequent PCI-style Programmed I/O commands to write to the HTEA or by using the new capabilities of the Data Center Bridging Capabilities Exchange Protocol (DCBX) [17] to communicate mapping parameters at the link layer level between DCB-enabled switches.

This mapping of local HT packets to remote nodes also requires the creation of a logical organization scheme to keep track of distinct source and destination pairs, known as a Virtual Link in the specification. As shown in Figure 3, a Virtual Link couples information such as available credits and buffers for the three virtual channels on the local link as well as information like the destination MAC address. Once the mapping layer of the HTEA decides which destination MAC address a particular HT request address maps to, the HyperTransport packet is queued according to available credits and associated buffer space at the remote HTEA. These credits are discussed more in Section 4.4.

4.3. Tag Remapping for Higher Performance

In addition to mapping local HT requests into the global address space supported by HToE, the HToE specification also supports mapping optimizations for the HTEA that allow for increased scalability while still preserving the local link's ability to transparently handle remote HT packets without needing knowledge of their source.

One of the limits to scalability in an HToE implementation is related to the number of outstanding Non-Posted requests that can be issued by a HyperTransport device at one time. Since the HTEA interface with the HyperTransport link follows all the normal protocols of a HyperTransport device, it is limited to sending a relatively small number of Non-Posted requests (that require a response packet) to the local link using unique Source Tag (SrcTag) bits. Furthermore, packets that are received at a HTEA may have their own Source Tag bits that conflict with requests from other source HTEAs. For this reason, the HToE standard implements a technique called tag remapping [30] to maximize the number of Non-Posted requests that can be sent to the local HT link. Figure 4 shows how tag remapping works with two conflicting incoming requests. The original SrcTag, Unit ID, and source MAC address are stored in a pending request table on receipt. If a newly arrived request conflicts with a pending request, its SrcTag and Unit ID bits are remapped and the mapping is maintained in the pending request table. On completion of the servicing of a request, the corresponding responses are matched up against this table to restore the SrcTag and Unit ID fields as well as to determine the correct destination HTEA for a response.

The HyperTransport specification also specifies an optional technique called Unit ID Clumping that can be used with tag remapping to give the HTEA additional Source Tags for use with the local HyperTransport link. Unit ID Clumping is not a requirement for HToE implementations, but it provides an example of how HToE can be scaled to handle additional sending HTEAs while conforming to the requirements of the original HyperTransport specification.

4.4. HToE Ordering and Flow Control for Multiple Senders, Single Receivers

HToE ordering relies on the HT 3.1 ordering requirements, also known as HyperTransport Section 6 Ordering Requirements. Although there are no requirements for packets going to different destinations (from different VLs), ordering of packets within a VL are preserved by the HToE retry algorithm and by sending all Ethernet packets for a specific source/destination pair on the same Ethernet priority level.

In contrast to point-to-point communication, a HTEA must receive packets from multiple source HTEAs. To handle this many-to-many communication pattern, the HToE specification uses a very simple credit-based principle for end-to-end buffer management – any HyperTransport packets that are sent to a remote node must have a standard HyperTransport credit for the Virtual Link before they can be encapsulated into an Ethernet packet. Additionally, each HT credit is equal to one buffer in the receiving HTEA.

Unlike HT links where HT credit-carrying NOP packets continuously flow on the physical link, credits are passed in the HToE environment only when the receiving HTEA has available buffers for incoming HT packets. A certain number of buffers must be reserved to allow sending HTEAs to initiate new connections, but additional buffers and credits are allocated by the receiving HTEA as its flow control and credit allocation schemes specify.

As buffers are filled in a receiving HTEA, the lack of available credits introduces backpressure on the sending HTEAs. Figure 5 shows how this backpressure causes buffers in the sending HTEA at Node 1 to become full, pausing transactions until more credits are

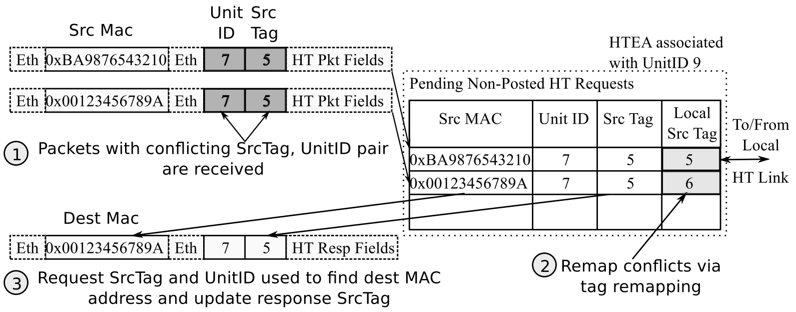
Src Mac
Unit ID
Src Tag
Eth
0xBA9876543210
Eth
7
5
HT Pkt Fields
Eth
0x00123456789A
Eth
7
5
HT Pkt Fields
1 Packets with conflicting SrcTag, UnitID pair are received
HTEA associated with UnitID 9
Pending Non-Posted HT Requests
Src MAC
Unit ID
Src Tag
Local Src Tag
0xBA9876543210
7
5
5
0x00123456789A
7
5
6
To/From Local HT Link
Dest Mac
Eth
0x00123456789A
Eth
7
5
HT Resp Fields
2 Remap conflicts via tag remapping
3 Request SrcTag and UnitID used to find dest MAC address and update response SrcTag

Figure 4. Tag Remapping in the HTEA

available. Note that since each Virtual Link has its own set of credits, lack of credits for one source-destination pair should not affect the traffic for another VL.

The HToE specification defines the minimum required flow control mechanism. However, it mentions and leaves open many opportunities to optimize the allocation of credits and buffers to multiple senders.

4.5. Encapsulation and Support for Recovery and Resets

In addition to specifying how HyperTransport packets are packed into Ethernet packets, the encapsulation layer also interacts with recovery and reset mechanisms that have been adapted from HT 3.1 to handle HToE packets. Each HToE packet contains a special sequence number that is used by the HToE retry algorithm to determine if HToE packets are received in order. This sequence number and retry algorithm are very similar to the 3.1 Go-Back-N algorithm, but each sequence number refers to an entire HToE packet, not just one HT packet. Further error checking is provided by CRCs on both the HToE Payload and the use of the normal Ethernet CRC.

In the case of an unrecoverable error that leads to reset, the encapsulation layer specifies a method for performing link-level resets of one or more Virtual Links that is similar to HyperTransport's concept of cold and warm resets. Since HyperTransport over Ethernet does not include the additional physical sideband signals that HyperTransport devices normally include (such as the power and reset signals), resets must be passed using packets or using OS-level communication. A special encapsulation packet header defines fields for these selective resets, limits their scope, and keeps the entire HTEA from having to reset due to an error between one source and one destination.

While some errors lead to reset, many errors just require a response to notify the original requesting processor that a request packet has not received a valid response. Similar to how HT 3.1 specifies a method for sending responses with error bits to notify of errors, HToE allows for remote transactions to be terminated and handles error notification. To do this the HTEA must keep track of sent HyperTransport packets that require a response (Non-Posted packets), and if it receives a notification that the response has been lost or the remote node has been reset, it can then reply with a normal HT 3.1 packet with the "poison" or error bit set. This additional state for remote requests allows for easier error detection and detection of request timeouts.

4.6. Security in HToE-enabled Data Centers

Since HyperTransport over Ethernet enables easy, transparent (OS interaction is not necessarily needed) hardware sharing of noncoherent memory between nodes, more care must be taken to make sure that malicious HyperTransport packets are not inserted into an Ethernet packet and sent to a remote node. While certain HPC-oriented clusters that are not used to handle web-related data may not have as high security requirements, networks exposed to the Internet may require additional security measures. Fortunately, HToE defines the use of IEEE 802.1ae MACsec to provide for encrypted 10 Gigabit Ethernet traffic between nodes.

5. Resource Sharing with HToE - Use Cases

The creation of a high-performance, scalable, commodity network using HyperTransport over Ethernet opens up the possibility of many application models that are based on low-latency noncoherent communication. Here we present two potential usages of this commodity standard to promote resource sharing within a data center or HPC environment. Both are predicated on the assumption that future clusters will be limited not nec-

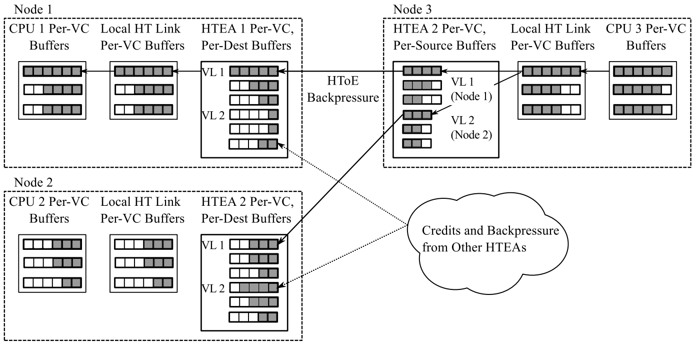
Node 1
CPU 1 Per-VC Buffers
Local HT Link Per-VC Buffers
HTEA 1 Per-VC, Per-Dest Buffers
VL 1
VL 2
HToE Backpressure
Node 3
HTEA 2 Per-VC, Per-Source Buffers
Local HT Link Per-VC Buffers
CPU 3 Per-VC Buffers
VL 1 (Node 1)
VL 2 (Node 2)
Node 2
CPU 2 Per-VC Buffers
Local HT Link Per-VC Buffers
HTEA 2 Per-VC, Per-Dest Buffers
VL 1
VL 2
Credits and Backpressure from Other HTEAs

Figure 5. HToE Backpressure-based Flow Control

essarily by processing power but rather by factors like TCO and power usage.

5.1. PGAS Support for Virtualizing DIMMs and DRAM Power Efficiency

Previous research has examined the use of HyperTransport over Ethernet as the hardware support for a PGAS implementation that can be used to reduce DRAM overprovisioning in servers in data centers [33]. DRAM in data centers is typically overprovisioned to handle infrequent peaks in workloads, but low-latency memory transfers can help reduce the need for overprovisioning while also providing much lower latency than swapping data out to disk.

These low-latency remote memory accesses provide an alternative to existing RDMA models and also allow for the "virtualizing" of DIMMs on remote nodes. This means that a node could request the use of part of a remote DIMM for noncoherent accesses to grow its own available memory temporarily. At the same time, applications running on the local node are unaware of the DIMM's actual location due to the transparent address mapping of a local HT request address into the global address space, the transmission of a low-latency HToE packet, and traditional CPU techniques that are used to hide normal memory access latency.

DIMM virtualization can provide opportunities for reducing the amount of installed DRAM in a data center, based on average memory requirements rather than peak requirements. For instance, a 10,000 core data center might currently consist of 625 individual blades, each with 4 sockets and quad-core CPUs. Based on previous estimates of memory requirements for data center workloads [6], each blade would require anywhere from 32 to 64 GB of DRAM in an overprovisioned scenario. The current retail price of a registered 8 gigabyte DDR3-1333 DIMM is around $300 [10], so reducing the amount of memory by 50% (from 64 GB to 32 GB) would save $750,000 over the entire data center. A 75% reduction would save $1,125,000 in memory costs alone, not to mention TCO related to cooling and power. Using HP's online power calculator, we can also estimate that this reduction in memory would save either 8,500 Watts (50% reduction) or 12,750 Watts (75%) due to related reductions in idle memory power [19].

5.2. Pooled Accelerators to Reduce Cluster TCO and Power Usage

In addition to virtualizing DRAM, there are also several researchers interested in virtualizing and sharing accelerators, such as GPUs. Provisioning an entire cluster with GPU cards can prove to be cost- and power-inefficient, especially in situations where only a few applications can take advantage of the benefits of better performance on these accelerators. In the same vein as other approaches that utilize MPI or sockets to access remote accelerators [13], HToE can be used as an enabling technology to allow for pooling accelerators (i.e., sharing a few accelerators between a larger number of general purpose nodes) and reducing cost and power inefficiency in the cluster.

While current approaches to use remote accelerator access would likely rely on using HToE packets to perform remote reads and writes to shared CPU-GPU memory pages, it is foreseeable that GPUs could be accessed directly using HyperTransport packets either natively or after being translated over the PCI Express bus. The availability of direct access to GPUs using HyperTransport packets would allow remote nodes to be able to directly read or write GPU DRAM and would provide a much higher performing model for sharing remote accelerators between nodes in a cluster.

Using our example cluster from Section 5.1 with mid-range GPUs, we can give a simplistic approximation of how pooled accelerators could be used to reduce overall cost and power usage. We assume that a blade could potentially house two PCIe-based GPU cards and that these GPUs are not typically fully utilized. The Fermi-branded, NVIDIA GeForce GTX 570 GPU currently retails for around $350 and has a maximum power dissipation of 220 Watts [26] and an idle power dissipation of around 30 Watts [2]. In our pooled accelerator scenario, one GPU could be shared between two adjacent blades, providing a 75% reduction in cost ($328,125 for the entire data center). More importantly, the power consumption due to idle GPUs would be reduced by at least 28,125 Watts (assuming each GPU uses 30 Watts when inactive).

These savings are highly dependent on the expected workload, but the existence of pooled accelerators would allow for much greater flexibility in the initial provisioning and upgrading of clusters to meet computational, power, and TCO requirements.

6. Conclusions

As part of the new HyperShare strategy, HyperTransport over Ethernet (HToE) provides a low-cost, commodity standard that can be used to enable new higher performance models of resource sharing in clusters and data centers. This specification proposes several engineering solutions for encapsulating HyperTransport packets over a highly scalable, many-to-many interconnect, and it provides cost- and performance-related motivation for using HToE in environments where 10 Gigabit Ethernet is already deployed and where 40 or 100 Gigabit Ethernet is likely to gain future market share.

Additionally, we have proposed several usage cases to demonstrate how HToE can be utilized to dramatically improve resource sharing for overprovisioned hardware such as DRAM and expensive accelerators such as GPUs. The HToE standard can enable these sharing techniques in data centers while taking advantage of the cost, scalability, and management benefits associated with Ethernet interconnect technology.

References

[1] IEEE 802.3ae 10Gb/s Ethernet Task Force. 10 gigabit ethernet 802.3ae standard. 2002. http://grouper.ieee.org/groups/802/3/ae/index.html.

[2] Nvidia's geforce gtx 570: Filling in the gaps - power, temperature, and noise. 2011. http://www.anandtech.com/show/4051/nvidias-geforce-gtx-570-filling-in-the-gaps/15.

[3] InfiniBand Trade Association. Rdma over converged ethernet specification. 2010. http://www.infinibandta.org.

[4] Pavan Balaji, Wu-chun Feng, and Dhabaleswar K. Panda. Bridging the ethernet-ethernot performance gap. *IEEE Micro*, 26:24–40, May 2006.

[5] Ulrich Bruening. The htx board: The universal htx test platform. http://www.hypertransport.org/members/u_of_man/htx_board_data_sheet_UoH.pdf.

[6] S. Chalal and T. Glasgow. Memory sizing for server virtualization. 2007. http://communities.intel.com/docs/.

[7] HyperTransport Consortium. Hypertransport specification, 3.10. 2008. http://www.hypertransport.org.

[8] HyperTransport Consortium. Clustering 360 market analysis. 2010. http://www.hypertransport.org/default.cfm?page=Clustering360.

[9] Pat Conway and Bill Hughes. The amd opteron northbridge architecture. *IEEE Micro*, 27(2):10–21, 2007.

[10] Crucial memory 8 gb, ddr3 pc3-10600 memory module pricing. 2011. http://www.crucial.com/server/index.aspx.

[11] Uri Cummings. Focalpoint: A low-latency, high-bandwidth ethernet switch chip. In *Hot Chips 18*, 2006. http://www.hotchips.org/archives/hc18/3_Tues/HC18.S8/HC18.S8T1.pdf.

[12] D. Dalessandro, P. Wyckoff, and G. Montry. Initial performance evaluation of the neteffect 10 gigabit iwarp adapter. In *Cluster Computing, 2006 IEEE International Conference on*, pages 1–7, 2006.

[13] J. Duato, A.J. Pea, F. Silla, R. Mayo, and E.S. Quintana-Orti. rcuda: Reducing the number of gpu-based accelerators in high performance clusters. In *High Performance Computing and Simulation (HPCS), 2010 International Conference on*, pages 224 –231, July 2010.

[14] J. Duato, F. Silla, S. Yalamanchili, B. Holden, P. Miranda, J. Underhill, M. Cavalli, and U. Bruning. Extending hypertransport protocol for improved scalability. In *First International Workshop on HyperTransport Research and Applications*, 2009. http://ra.ziti.uni-heidelberg.de/coeht/pages/events/20090212/whtra09-paper16.pdf.

[15] Fibre channel over ethernet fc-bb-5 standard. 2010. http://www.t11.org/fcoe.

[16] Silicon Graphics. Sgi numalink: Industry leading interconnect technology (white paper). 2005. http://www.sgi.com.

[17] IEEE 802.1 Working Group. Ieee 802.1qaz standards page (in progress). http://www.ieee802.org/1/pages/802.1az.html.

[18] Interconnect share of top 500 for november 2010 - hpc top 500. 2010. http://www.top500.org.

[19] Hp power advisor. 2011. http://h18000.www1.hp.com/products/solutions/power/advisor-online/HPPowerAdvisor.html.

[20] Swamy N. Kandadai and Xinghong He. Performance of hpc applications over infiniband, 10 gb and 1 gb ethernet. 2010. http://www.chelsio.com/assetlibrary/whitepapers/HPC-APPS-PERF-IBM.pdf.

[21] M. Ko, D. Eisenhauer, and R. Recio. A case for convergence enhanced ethernet: Requirements and applications. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 5702 –5707, May 2008.

[22] Rajesh Kota and Rich Oehler. Horus: Large-scale symmetric multiprocessing for opteron systems. *IEEE Micro*, 25(2):30–40, 2005.

[23] Jiuxing Liu, Jiesheng Wu, Sushmitha P. Kini, Pete Wyckoff, and Dhabaleswar K. Panda. High performance rdma-based mpi implementation over infiniband. In *Proceedings of the 17th annual international conference on Supercomputing*, ICS '03, pages 295–304, New York, NY, USA, 2003. ACM.

[24] Myricom's myri-10g 10-gigabit ethernet solutions. 2010. http://www.myri.com/Myri-10G/10gbe_solutions.html.

[25] Juniper Networks. Press release for juniper network's t1600 100 ge core router. 2009. http://www.juniper.net/us/en/company/press-center/press-releases/2009/pr_2009_06_08-09_00.html.

[26] Nvidia geforce gtx 570 specification. 2011. http://www.nvidia.com/object/product-geforce-gtx-570-us.html.

[27] M. Schlansker, J. Tourrilhes, Y. Turner, and J.R. Santos. Killer fabrics for scalable datacenters. In *Communications (ICC), 2010 IEEE International Conference on*, pages 1 –6, May 2010.

[28] Woven Systems. 10 ge fabric delivers consistent high performance for computing clusters at sandia national labs. 2007. http://www.chelsio.com/assetlibrary/pdf/Sandia_Benchmark_Tech_Note.pdf.

[29] Jon Tate. An introduction to fibre channel over ethernet, and fibre channel over convergence enhanced ethernet. 2009. http://www.redbooks.ibm.com/redpapers/pdfs/redp4493.pdf.

[30] Sudhakar Yalamanchili, Jose Duato, Jeffrey Young, and Federico Silla. A dynamic, partitioned global address space model for high performance clusters. Technical report, 2008. http://www.cercs.gatech.edu/tech-reports/tr2008/git-cercs-08-01.pdf.

[31] Yasushi Umezawa Yoichi Koyanagi, Tadafusa Niinomi. 10 gigabit ethernet switch blade for large-scale blade servers. *Fujitsu Scientific and Technical Journal*, 46(1):56–62, 2010.

[32] Jeff Young and Brian Holden. Hypertransport over ethernet specification, 1.0. 2010. http://www.hypertransport.org.

[33] Jeffrey Young and Sudhakar Yalamanchili. Dynamic partitioned global address spaces for power efficient dram virtualization. In *Works in Progress in Green Computing, 2010 International Green Computing Conference*, 2010.