Efficient Wide Area Data Transfer Protocols for 100 Gbps Networks and Beyond

Ezra Kissel
School of Informatics and Computing
Indiana University
Bloomington, IN 47405
ezkissel@indiana.edu

Martin Swany
School of Informatics and Computing
Indiana University
Bloomington, IN 47405
swany@iu.edu

Brian Tierney
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
bltierney@lbl.gov

Eric Pouyoul
Lawrence Berkeley National Laboratory
Berkeley, CA 94720
epouyoul@lbl.gov

Due to a number of recent technology developments, now is the right time to re-examine the use of TCP for very large data transfers. These developments include the deployment of 100 Gigabit per second (Gbps) network backbones, hosts that can easily manage 40 Gbps, and higher, data transfers, the Science DMZ model, the availability of virtual circuit technology, and wide-area Remote Direct Memory Access (RDMA) protocols. In this paper we show that RDMA works well over wide-area virtual circuits, and uses much less CPU than TCP or UDP. We also characterize the limitations of RDMA in the presence of other traffic, including competing RDMA flows. We conclude that RDMA for Science DMZ to Science DMZ transfers of massive data is a viable and desirable option for high-performance data transfer.

Categories and Subject Descriptors
C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*High-speed*; D.2.8 [**Software Engineering**]: Metrics—*Performance measures*

Keywords
performance measurement, networking, RDMA

1. INTRODUCTION
Modern science is increasingly data-driven and collaborative in nature. Large-scale simulations and instruments produce petabytes of data, which is subsequently analyzed by tens to thousands of scientists. Although it might seem logical and efficient to colocate the analysis resources with the source of the data (instrument or a computational cluster), this is not the likely scenario. Distributed solutions – in which components are scattered geographically – are much more common at this scale, for a variety of reasons, and the largest collaborations are most likely to depend on distributed architectures. Efficient tools are necessary to move vast amounts of scientific data over high-bandwidth networks in such state-of-the-art collaborations.

NDM'13 November 17, 2013, Denver CO, USA

http://dx.doi.org/10.1145/2534695.2534699

The Large Hadron Collider[1] (LHC), the most well-known high-energy physics collaboration, was a driving force in the deployment of high bandwidth connections in the research and education world. Early on, the LHC community understood the challenges presented by their extraordinary instrument in terms of data generation, distribution, and analysis.

Many other research disciplines are now facing the same challenges. The cost of genomic sequencing is falling dramatically, for example, and the volume of data produced by sequencers is rising exponentially. In climate science, researchers must analyze observational and simulation data sets located at facilities around the world. Climate data is expected to exceed 100 exabytes by 2020 [12]. New detectors being deployed at X-ray synchrotrons generate data at unprecedented resolution and refresh rates. The current generation of instruments can produce 300 or more megabytes per second and the next generation will produce data volumes many times higher; in some cases, data rates will exceed DRAM bandwidth, and data will be preprocessed in real time with dedicated silicon. To support these increasing data movement demands, new approaches are needed to overcome the challenges that face existing networks technologies.

There are a number of recent technology developments that make now the right time to reexamine the use of TCP for very large data transfers. These developments include:

- 100G backbone networks are now deployed in a number of countries. ESnet, Internet2, GÉANT largely consist of 100G paths. Many large research sites have 100G connections to these backbones. For example,

[1]The Large Hadron Collider: http://lhc.web.cern.ch/lhc/

as of August 2013, seven DOE laboratories have 100G connections to ESnet, and 29 Universities will have 100G connections to Internet2 by the end of 2013.
- 40G host NICs are easily available and affordable. 100G host NICs will be available in 2014. While its possible to get TCP to saturate 40G connections over high-latency paths, it is very sensitive and requires careful tuning, and even then requires significant CPU power to achieve.
- RDMA-based protocols are a well-proven data center technology offering high performance and efficiency, that can also be utilized in wide-area networks.
- Most Research and Education (R&E) network providers now support guaranteed bandwidth virtual circuits. RDMA-based protocols require the traffic isolation and bandwidth guarantees provided by virtual circuits.
- Many sites are deploying a "Science DMZ" to provide very high-speed data transfers to the wide area, and which support virtual circuit services.

As network capacity increases along with data movement demands, new approaches are needed to overcome the challenges that face existing networks technologies. The viability of many computing paradigms depends on the ability for data distribution to scale efficiently over global networks at 100 Gbps speeds. We argue that through a combination of intelligent network provisioning and the use of RDMA protocols, we can achieve significant gains over existing methods in supporting efficient, high-performance data movement over the WAN.

Initially we envision the primary use case for RDMA-based transfers is for Science DMZ to Science DMZ transfers, where it is easy to set up end-to-end virtual circuits. Many science communities replicate the large data sets on multiple continents for better access by local scientists. For example the climate community plans to replicate hundred of exabytes of data to multiple sites around the world by the year 2020. [12] These transfers will likely use a cluster of 5-10 Data Transfer Nodes (DTNs) at one site transferring files in parallel to a DTN cluster at the remote site. Our results show that this sort of cluster to cluster transfer using RDMA over the WAN works well in the right environment.

Another initial use case for RDMA transfers is to transfer data from a large scientific instrument to remote storage. For example, currently one of the beamlines at the Advanced Light Source at LBNL[2] has a virtual circuit connection to the NERSC Supercomputer center 10 miles away in order to quickly analyze the data collected. While this is currently done using TCP, in the near future data rates will be 10-100x larger, and lower CPU usage of RDMA will be very beneficial.

In this paper we expand on our preliminary work on 40 Gbps RDMA which demonstrated its efficacy over wide-area virtual circuits, providing good performance and using much less CPU than TCP or UDP [27]. We also characterize the limitations of RDMA-based protocols in the presence of other traffic, including competing RDMA traffic. This paper demonstrates that using RDMA protocols such as RoCE (RDMA over Converged Ethernet) [7] one can achieve significant gains over existing methods of high-performance data movement over the WAN. In particular, our results show that RoCE over virtual circuits is worth further evaluation and consideration.

While other papers have shown that RDMA over the WAN works well, this paper is the first to seriously examine the limitations of multiple flows on the same circuit, and the limitations of bottleneck links to a RDMA flow.

[2]ALS, http://www-als.lbl.gov/

2. BACKGROUND

It is well known that the ubiquitous Transmission Control Protocol (TCP) has significant performance issues when used over long-distance, high-bandwidth networks [10, 18, 22]. With proper tuning, an appropriate congestion control algorithm, and low-loss paths, TCP can perform well over high-speed links. Yet, the system overhead of single and parallel stream TCP at 10 Gbps and higher is capable of fully using a core on modern processors, raising questions about the viability of TCP as network speeds continue to grow. The administrative burden of ensuring proper TCP settings for various network scenarios is also a persistent challenge. UDP-based protocol implementations such as such as UDT [17] and Aspera's *fasp* [1] provide benefits in certain cases but suffer from increased overhead due to user space buffer copying and context switching, which limits their use for many high-performance applications.

One solution to this problem has been the use of *zero-copy* techniques within the Linux kernel. Introduced in the 2.6 kernel, the *splice()* and *vmsplice()* system calls use a kernel memory pipe to "splice" or connect file descriptors, which may refer to network sockets, and memory pages while avoiding costly copying to and from user space buffers. As we demonstrate, splice support provides significant benefits for TCP sender applications, but it is not a complete solution as network speeds and application demands continue to increase.

2.1 Virtual Circuit Services

Most R&E backbone networks today support virtual circuits in order to provide bandwidth guarantees and traffic isolation. One example is ESnet's OSCARS service [23]. The characteristics of the OSCARS service, which are similar to other, compatible virtual circuit services, are guaranteed, reservable bandwidth with a particular start and end time; resiliency (explicit backup paths can be requested); data transport via either Layer-3 (IP) or Layer-2 (Ethernet) circuits, and integrity of the established circuits. Traffic isolation is provided to allow for use of high-performance, non-standard transport mechanisms that cannot co-exist with commodity TCP-based transport in the general infrastructure. The underlying mechanisms allow for traffic management, which means that explicit paths can be used to meet specific requirements — e.g. bypassing congested links, using higher bandwidth paths, explicit engineering of redundancy, and so on.

Even though virtual circuits have been deployed by wide-area R&E network providers, constructing a guaranteed bandwidth path all the way to the end host systems is chal-

lenging. Fortunately ESnet's "Science DMZ" model[3] is being deployed at a number of research institutes.

2.2 The Science DMZ

Despite provisioning high-capacity connections to their sites, scientists struggle with improperly built cyberinfrastructure that cripples their data transfer performance and impedes scientific progress. ESnet's Science DMZ model comprises a proven set of network design patterns that collectively address these problems [14]. The Science DMZ model includes network architecture, system configuration, cybersecurity, and performance tools that create an optimized network environment for science data.

In the Science DMZ model, fast "Data Transfer Nodes", or DTN's, are deployed at the border of the campus network. Placing these nodes near the site's WAN connection has a number of benefits, making it easier to (1) create and maintain a high quality path for a DTN, free of packet loss, underpowered devices, etc., (2) create end-to-end virtual circuits, and (3) justify and maintain a security policy that does not impede high-speed flows.

To support RDMA transfers, having the DTNs much closer to the termination point of the WAN virtual circuit makes is much easier to extend the guaranteed bandwidth all the way to the end hosts. In addition, these nodes can be customized to maximize performance. Various HPC organizations have been pushing the limit on the maximum I/O bandwidth for a single host. NASA Goddard has demonstrated 115 Gbps in back to back memory-to-memory testing, and 96 Gbps disk-to-disk tests using SSD drives [6]. Caltech has reported 80 Gbps with a single host [11], and in our own testing we have seen similar performance.

2.3 Remote DMA

The InfiniBand Architecture (IBA) [4] and related RDMA protocols have played a significant role in enabling low-latency, high-throughput communications over switched fabric interconnects, traditionally within data center environments. RDMA operates on the principle of transferring data directly from the user-defined memory of one system to another. These transfer operations can occur across a network and bypass the operating system (OS), eliminating the need to copy data between user and kernel memory space. Direct memory operations are supported by allowing network adapters to register buffers allocated by the application. This "pinning" of memory prevents OS paging of the specified memory regions and allows the network adapter to maintain a consistent virtual to physical mapping of the address space. RDMA can then directly access these explicitly allocated regions without interrupting the host operating system.

Our RDMA implementations make use of the InfiniBand Verbs, *ibverbs*, and RDMA Communication Manager, *rdmacm*, libraries made available within the OpenFabrics Enterprise Distribution [5]. OFED provides a consistent and medium-independent software stack that allows for the development of RDMA applications that can run on a number of different hardware platforms. The IBA specification itself supports both reliable (**RC**) and unreliable (**UC**) RDMA connections. In addition, two different transfer semantics are available: 1) "two-sided" RDMA SEND/RECEIVE references local, registered memory regions which requires posted RECEIVE requests before a corresponding SEND, and 2) "one-sided" RDMA READ/WRITE operations can transfer buffers to and from memory windows whose pointers and lengths have been previously exchanged. The transfer tools developed for our evaluation use RDMA over RC and implement the RDMA WRITE operation. We note that RDMA READ would perform similarly in our bulk data evaluation since the additional signaling cost [21] would be amortized over the duration of our transfers.

The emerging RDMA over Converged Ethernet (RoCE) [7] standard allows users to take advantage of these efficient communication patterns, supported by protocols like InfiniBand, over widely-deployed Ethernet networks. In effect, RoCE is the InfiniBand protocols made to work over Ethernet infrastructure. The notion of "converged Ethernet", also known as enhanced Ethernet or Data Center Bridging (DCB), is that of including various extensions to the IEEE 802.1 standards to provide priority-based flow control (PFC), bandwidth management, and congestion notification at the link layer. Since the InfiniBand protocols operate in networks that are virtually loss-free, the motivation for this is clear. The protocol, however, does not directly require any of these extensions and thus it is possible to use RoCE in WAN environments. Until the recent introduction of RoCE, InfiniBand range extenders such as Obsidian Longbow routers were one of the few options that allowed for an evaluation of InfiniBand protocols over long distance paths.

Certain path characteristics are necessary to effectively use the RoCE protocol over wide-area networks, however. The path should be virtually loss-free and should have deterministic and enforced bandwidth guarantees. Even small amounts of loss or reordering can have a detrimental impact on RoCE performance. As we will show, link layer flow control (i.e. PAUSE frames) is required if supporting competing flows across the same aggregation switch. In such scenarios, the now common IEEE 802.1Qbb PFC capability in the data center bridging standards is recommended to classify and prioritize RoCE flows, but again not necessary. Note that the ability to do RoCE also requires RoCE-capable NICs, such as the Mellanox adapters used in our evaluation [2].

2.4 Tuning RDMA for the WAN

Beyond the logistics involved with traditional host tuning, a number of important considerations govern the ability of RDMA-based applications to achieve expected performance over the WAN, especially as latency increases. It is well understood that the transmission of buffers over the network must be pipelined in order to keep enough data "in-flight" and saturate the capacity of the given network path. TCP solves this by using a sliding window protocol tuned to the round-trip time (RTT) of the path. In contrast, applications using RDMA are directly responsible for allocating, managing, and exchanging buffers over networks that can have dramatically different requirements, from interconnects with microsecond latencies to RoCE over WANs with 100ms or greater RTTs.

[3]Science DMZ: http://fasterdata.es.net/science-dmz/

There are two controllable factors that influence the wide-area network performance of an RDMA application: 1) the number and size of buffers, and 2) the number of RDMA operations that are posted, or in transit, at any given time. In the context of an RDMA communication channel, this corresponds to the message size, or size of the memory window in RDMA READ/WRITE operations, and the transmit queue depth, *tx-depth*, respectively. In general, managing fewer, larger buffers can result in less overhead for both the application and the wire protocol, and we developed a configurable ring buffer solution in support of our RDMA implementations that allowed us to experiment with various buffer/message sizes for the transfer of real data. The design of the application itself is tasked with allocating adequate buffers and posting enough RDMA operations based on the characteristics of the network path in order to saturate the given link capacity. Depending on the requirements of a particular application, either RDMA middleware libraries or direct integration of the InfiniBand Verbs API can be used. Although the details of these implementations are outside the scope of this paper, we note that both our tools and commonly available RDMA benchmarks allow for the explicit tuning of message sizes and transmit queue depths.

2.5 High-Performance Network Applications

The focus of this paper is to better understand network performance across a number of protocols and network scenarios. To that end, we chose not to optimize any particular data dissemination or workflow system that can take advantage of high-performance networks, but rather made use of existing file transfer applications and benchmarks that allowed us to better characterize data transfer behavior. We also developed our own benchmarking application for the purposes of our experimental evaluation.

RoCE performance results were collected using our own network benchmark called *xfer_test*, which allowed us to compare both TCP and RoCE transfers from the same application. In addition to the *zero-copy* techniques supported by RDMA protocols, we take advantage of the Linux kernel *zero-copy* splice support, as described above, in our *xfer_test* implementation. The benefit of this approach is highlighted in our 10 Gbps and 40 Gbps network benchmarking results.

A number of Sockets-based network benchmarking tools are also publicly available. For our 40 Gbps analysis, we made use of the *netperf* [4] tool since it supports accurate and high-throughput TCP and UDP tests. In addition, the *netperf* tool includes a TCP_SENDFILE test, which uses *splice()* to provide a *zero-copy* memory-to-memory TCP transfer benchmark.

The Globus Toolkit's GridFTP [16] distribution was used to provide performance numbers for a widely-used and well-supported transfer tool. Our previous work [20] developed an RDMA driver within the Globus XIO [19] framework on which GridFTP is built. However, due to overheads involved in enabling high-performance RDMA within the existing XIO implementation, we focus on our *xfer_test* transfer tool in the following evaluation. Related work [30, 31] has investigated extending XIO to make it more suitable for RDMA-style communication, and we plan to take advantage of these advances in future testing.

[4]netperf:, `http:/www.netperf.org/`

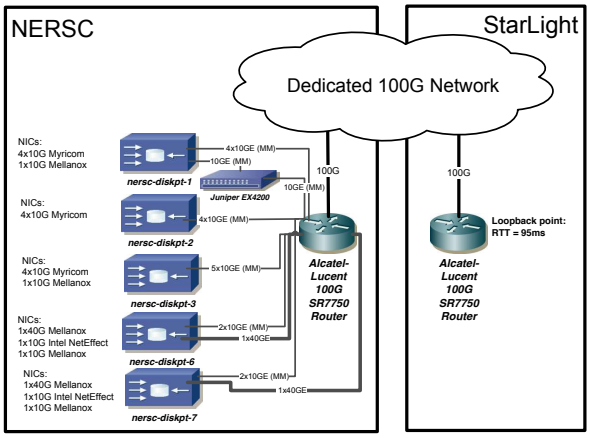


Figure 1: ESnet 100G Testbed Resources Used

3. EVALUATION

3.1 Overview

Our evaluation of network performance took place on two separate testbeds. First, we investigated and compared network performance between different protocols and applications between hosts on the ESnet 100G Testbed. Then, for a more detailed analysis of RoCE transfer performance, we made use of a smaller 10 Gbps testbed that contained a Spirent XGEM [3] network impairment device, allowing for finer-grained control of network conditions. Among the more immediate goals are to provide a better understanding, along with recommendations, for future use of RoCE in a number of network configurations. Unless otherwise noted, the results from each performance test is for a single flow between a client and server application.

3.2 ESnet 100G Testbed

Our performance analysis uses resources from ESnet's 100G Testbed[5], which includes a 100 Gbps wave connecting the National Energy Research Scientific Computing Center[6] (NERSC) in Oakland, CA to StarLight[7] in Chicago, IL.

The ESnet Testbed, a public testbed open to any researcher, is shown in Figure 1, includes high-speed hosts at both NERSC and StarLight. The results shown below were collected using two 40 Gbps capable hosts (*diskpt-6,7*) and two 10 Gbps hosts (*diskpt-1,3*) at NERSC. Each of the 10 Gbps hosts has two 6-core Intel processors and 64GB of system memory. These are all PCI Gen-2 hosts, which support a maximum of 27 Gbps IO per flow. Each 40 Gbps host has similar specifications but is based on the Intel Sandy Bridge with PCI Gen-3, supporting double the previous generation bus capacity.

To meet our goal of evaluating WAN protocol performance, the path between NERSC and StarLight was configured as a loop across the 100G testbed. Traffic originating from a configured host interface at NERSC would reach StarLight

[5]ESnet 100G Testbed http://www.es.net/testbed/
[6]National Energy Research Center http://www.nersc.gov
[7]StarLight: http://www.startap.net/starlight/

and then return to NERSC for a total RTT of 95ms. All of our ESnet 100G testbed tests were run along the loopback path unless where otherwise noted. The layer 2 circuit between sites is also divisible into dedicated bandwidth "circuits", which allowed us to investigate bottleneck conditions over the course of our testing.

All of the hosts ran a recent 3.5.7 Linux kernel. For RDMA testing with RoCE, we installed and configured OFED-3.5 with the supplied Ethernet drivers for the Mellanox ConnectX-3 and Intel NetEffect NICs. We performed standard host and NIC driver tuning to ensure the best performance for each of our benchmarks The Maximum Transmission Unit (MTU) on each installed NIC was set to 9000 bytes and Rx/Tx link layer flow control was enabled by default. For the RoCE tests, the effective MTU is limited to 2048 bytes as defined by the current RoCE specification. Both 10G and 40G Mellanox NICs were used for the RoCE tests, and each of our experiments involved memory-to-memory transfers to remove disk I/O as a potential bottleneck.

Note that the ESnet 100G testbed, while very high performance, is not a particularly realistic way to emulate a campus network. In this testbed, host interfaces are connected directly to high-end 100 Gbps Alcatel-Lucent Model SR 7750 border routers, which has a large amount of buffer space. This may mask some of the issues that would be seen with endpoints in a real campus network with less-capable devices in the path. However, we also note that no observable performance difference was detected between 10 Gbps flows that passed through an intermediate Juniper EX4200 Ethernet switch compared to the directly connected SR 7750.

3.3 10 Gbps Testing

We began with a set of tests to evaluate the performance and CPU requirements of different protocols for data transfers over 10 Gbps circuits.

We ran a series of tests to get baselines for the *xfer_test* and GridFTP applications when running single stream transfers over a 10G NIC with uncapped 100 Gbps WAN capacity. Each test was run for 120 seconds and the steady-state maximum performance achieved was recorded. The host monitoring tool *nmon* was used to collect CPU usage statistics for both the sender and receiver host over the duration of each test. The total CPU percentage is additive across the 12 cores present in a given system. Thus, 150% CPU usage denotes one core fully utilized and a second core 50% utilized, for example. The result of each 10Gbps test is summarized in Table 1.

With the more than capable hardware, the *xfer_test* tool had no trouble reaching close to line rate transfer speeds. Sending data with TCP Sockets is more efficient in the *xfer_test* TCP cases compared to the receiving side, but both sender and receiver are well below fully utilizing a single core. The benefits of the *splice()* system call are significant in reducing the overhead involved in sending data from user space over a TCP Socket. However, the clear winner in terms of transfer efficiency is *xfer_test* using RoCE, reaching 9.7 Gbps with only 1% CPU utilization on both the sender and receiver. Here we observed the RoCE test reaching 9.7 Gbps, which is approximately the maximum achievable "goodput" possible with a 2KB MTU when factoring in protocol overhead.

Table 1: ESnet 100G Testbed, 10 Gbps performance results..

Tool	Protocol	Gbps	Tx_CPU	Rx_CPU
xfer_test	TCP	9.9	45%	86%
xfer_test	TCP-splice	9.9	13%	85%
xfer_test	RoCE	9.7	1%	1%
gridftp	TCP	9.2	91%	88%
gridftp	RoCE	8.1	100%	150%

Table 2: ESnet 100G Testbed, 40 Gbps performance results.

Tool	Protocol	Gbps	Tx_CPU	Rx_CPU
netperf	TCP	17.9	100%	87%
netperf	TCP-sendfile	39.5	34%	94%
netperf	UDP	34.7	100%	95%
xfer_test	TCP	22	100%	91%
xfer_test	TCP-splice	39.5	43%	91%
xfer_test	RoCE	39.2	2%	1%
gridftp	TCP	13.3	100%	94%
gridftp	RoCE	13	100%	150%

Finally, single stream GridFTP tests were run for TCP and RDMA XIO drivers. Due to a combination of limited memory bandwidth on the 10 Gbps systems and GridFTP/XIO overheads, were were unable to achieve more than 9.2 Gbps application performance in the TCP case, and worse performance for the RoCE case. We are working with the GridFTP developers to optimize and improve single stream transfer performance for this application.

It is also worth mentioning that we evaluated the 10G Intel NetEffect NE020 iWARP NICs available in the testbed. Unfortunately, with a maximum TCP sender window of 256 KB, the NetEffect TCP engine was not suitable for use in WAN environments, capable of only saturating 10 Gbps paths where the RTT is <1ms. The NICs did, however, perform very well in a LAN setting, achieving 9.6 Gbps with only 1% CPU utilization at both the sender and receiver.

3.4 40 Gbps Testing

Since all protocols we tested were able to saturate a 10G NIC, we next ran the same set of tests over a 40G NIC in order to see what the limitations are. Building on our previous 40 Gbps evaluation [27], for completeness and accuracy we repeated those benchmarks on the updated testbed and host configuration. We found the results to be unchanged, as shown in Table 2. Results for the *netperf* benchmark tool are included to compare with our own *xfer_test* tool and to provide an accurate UDP data point.

At 40 Gbps, we begin to see the effects that insufficient memory bandwidth and buffer copying have on traditional Socket-based transfer approaches. In each of our three TCP application tests, none were able to achieve more than 45% of the available link bandwidth. The best result was obtained by xfer test at 22 Gbps for a single stream. The limiting factor is the overhead involved in copying between kernel and user space buffers. A clear advantage of using the *zero-copy* calls in our benchmarks is the reduction in

memory copy overhead, allowing single stream TCP performance to saturate the 40G links with significantly reduced CPU utilization at the sending side (TCP-sendfile and TCP-splice cases). GridFTP is again limited by the overheads within the current XIO framework and achieves 13 Gbps for both TCP and RDMA cases.

The *xfer test* RoCE case is once again the clear winner in terms of achieving single-stream 40 Gbps performance with minimal system overhead. As in our 10 Gbps testing, reaching 40 Gbps speeds using RoCE involves ensuring the application has allocated and posted adequate buffers to match the network characteristics. At 95ms, the bandwidth-delay product (BDP)[8] of the path approaches 500MB. This necessitates that an application or middleware system with a real workload would need to allocate and manage a substantial buffer(s) of data to sustain 40 Gbps throughput.

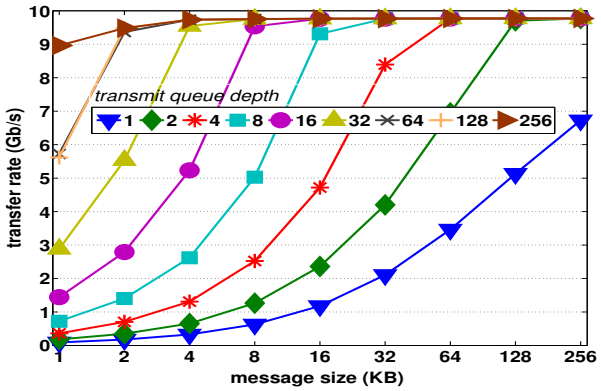
3.5 Overhead and Tuning Discussion

As is clearly shown in Tables 1 and 2, Socket-based applications incur costly overheads that limit their performance at faster network speeds. We have used the Linux *perf* [15] tool to perform system call profiling of our application during transfer tests. Profiling results indicate that a majority of CPU time is spent performing memory copies. Using Linux *zero-copy* support in the form of *splice()* or *sendfile()* calls reduces user space to kernel buffer copying significantly; however, context switching to kernel space remains the leading contributor to CPU load. Another issue is sending data via Socket descriptors requires send/recv loops that keep the application busy while it could be doing performing other tasks.

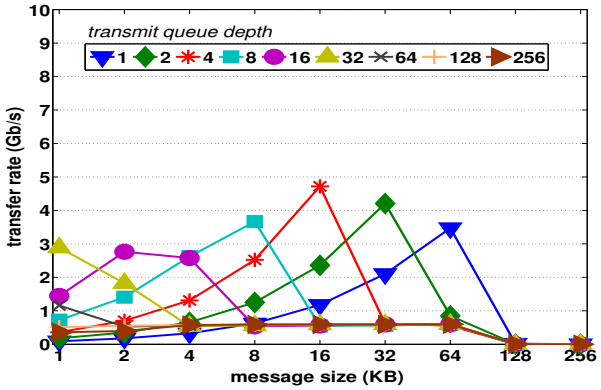
In contrast, RDMA protocols give high-performance transfer applications a mechanism to move large data sets with very little system overhead, thus enabling an efficient overlap between communication and computation for a number of data-intensive use cases. The application simply has to post operations and check the associated RDMA completion queue for events. At 40 Gbps, this bookkeeping requires a minimal 2% of a single core. Since these operations are non-blocking and asynchronous, the application is not involved in the active sending of data from memory buffers as is the case with Socket-based send/recv loops. Additionally, a key benefit with RDMA is the reduction of data cache loads issued by the CPU, thereby reducing memory bus contention and providing more bandwidth for disk I/O operations, for example. Results of our *perf* profiling have seen orders of magnitude reductions in the number of issued load instructions when comparing RDMA transfers tasks with their TCP equivalents.

Finally, we note that achieving the 40 Gbps performance results in the previous sections required a number of host and NIC tuning tasks. An issue that affects all high-performance applications, regardless of network protocol, is evolving architectural changes in modern off-the-shelf systems, in particular the introduction of Non-Uniform Memory Architecture (NUMA) systems such as the Intel Sandy Bridge 40G hosts. As we discovered, naively installing a 40G NIC in such a system and starting transfer tests will result in inconsistent

[8]The bandwidth delay product of a network path is typically calcuated as $BDP = RTT_{seconds} * Rate_{Bps}$



(a) 10 Gbps clean links



(b) 5 Gbps bottleneck

Figure 2: Effect of bottleneck link on RoCE performance. No added latency.

and often abysmal performance depending on the default CPU scheduler and IRQ balancing mechanisms present on the host operating system. Key settings to achieve repeatable and consistent results include: (i) setting IRQ affinity so that a single core handles all interrupts from the NIC, (ii) binding application sender and receiver threads to specific cores to avoid CPU migration, and (iii) binding memory to the same NUMA node as the thread and interrupt handling to avoid crossing memory boundaries and ensure memory locality for our application buffers [9]. As operating systems gain better support for consumer-based NUMA platforms, we expect that intelligent binding and interrupt handling will eventually become a less burdensome task. As of this writing, ensuring proper tuning of 40G NICs is still a manual, yet critical, step that is left up to the application developer and user of the system.

3.6 Empirical RoCE Performance Analysis

The final set of tests we ran were designed to allow for a more detailed understanding of how RoCE transfers behave over varying conditions. This includes multiple competing RoCE flows in the same circuit, multiple TCP and RoCE flows in the same circuit, on both low latency and high latency paths. It also includes tests with a 5 Gbps bottleneck on a 10 Gbps path. Our previous work also tested RoCE competing with UDP flows, and RoCE on a lossy path, so those results are not included here. Our aim is to collect empirical data that can guide the deployment of RoCE on real networks.

[9]http://fasterdata.es.net/host-tuning/interrupt-binding/

3.6.1 10G Testbed with XGEM

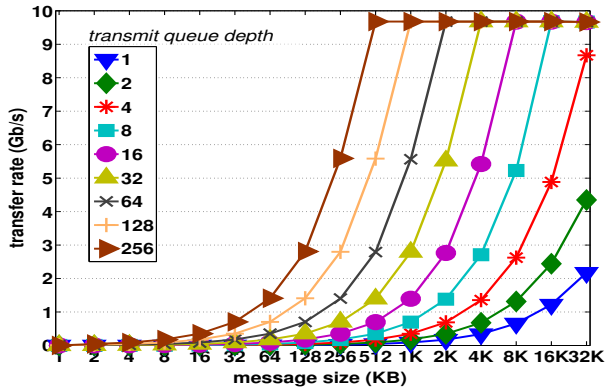
Distinct from the ESnet 100G Testbed, the systems used for these experiments are compute nodes with 8-core Intel Xeon CPUs and 189 GB of RAM at Indiana University. Each ran the same Linux 3.5.7 kernel with OFED-3.5 and CUBIC as the default congestion control algorithm. Appropriate host and NIC driver tuning was also applied. Each node contains a dual-port Mellanox ConnectX-3 10G NIC connected to a 10G top-of-rack switch supporting data center bridging standards. In addition, the Indiana University testbed makes use use of a Spirent XGEM [3] network impairment device, also connected to the switch. The 10G switch allowed us to direct traffic between any pair of hosts simultaneously across the XGEM, giving us the ability to evaluate the performance of competing RoCE and TCP transfers while emulating WAN latencies and bottleneck characteristics. We could also enable and disable Rx/Tx link layer flow control on the appropriate switch ports for a number of our experiments.

The following results represent a collection of tests run with no added latency, emulating a data center network, as well as the same tests run with 95ms added latency to emulate the loopback circuit on the ESnet 100G Testbed. Each experiment consisted of a repeatable sequence of RoCE transfers sweeping the message size and transmit queue depth parameter space over powers of two.

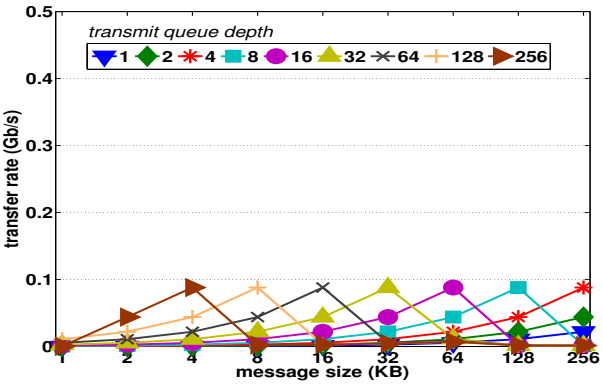
3.6.2 RoCE and Network Bottlenecks

Our first test verifies the RoCE performance over 10G links with no added latency, Figure 2(a). With a measured RTT of 0.204ms, we calculate a BDP of ≈ 256 KB, which matches our testing results. The performance chart illustrates the importance for the application or RDMA middleware to post enough RDMA operations to fill the network pipe. Without any policing along the path, we can send very large messages (up to 1 GB for RoCE) and the NIC will automatically pace to the given link speed.

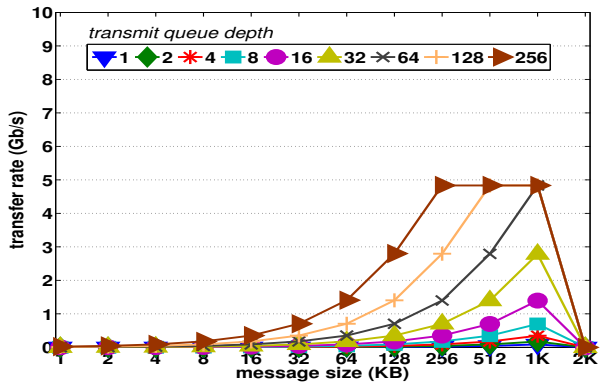
The picture changes dramatically when we introduce a 5 Gbps hard policer using the XGEM, creating a bottleneck between sender and receiver hosts. The policer, or "Line Bandwidth Control", in the XGEM is configured with the maximum allowed 1MB buffer, which is obviously much larger than the BDP in this case. There is also an interesting dynamic given the very small RTT in these tests. The BDP over a 5 Gbps path with 0.204ms should be half of the original 10 Gbps path, ≈ 128 KB. However, we see in Figure 2(b) that the bottleneck link is getting close to saturated at around 65 KB "in flight" from the test application. This is due to the fact that the BDP is bound by the 5 Gbps bottleneck link to the receiver since the policer effectively splits the small RTT between hosts. The empirically determined BDP is closer to ≈ 75 KB. The take away from this result is that with a larger-than-BDP buffer, the policer is able to absorb bursts from the faster edge link, and as long as the sender bursts messages totaling less than the BDP of the path, the bottleneck can be saturated without rate limiting. If the burst is greater than the BDP of the path, the policer buffer will fill as the network pipe is not long enough to contain the burst and the RoCE transfer will eventually stall as frames are dropped.



(a) 10 Gbps clean links



(b) 5 Gbps bottleneck



(c) 5 Gbps bottleneck with rate limiting

Figure 3: Effect of bottleneck link on RoCE performance. With 95ms RTT.

In Figure 3, we repeat the above experiments but this time with the XGEM also introducing 95ms RTT. With clean links (no policer), we increase our maximum message size to 32 MB in order to saturate the high-BDP (≈ 119 MB) path as shown in Figure 3(a). In Figure 3(b), we introduce the 5Gbps bottleneck again and see a much different picture than the low-latency case above. Here, the 1MB policer buffer becomes the limiting factor in how much data the transfer application can send. Note the reduced transfer rate scale in the figure. The application can burst up to the buffer size, or 1 MB total messages, in each test before the policer will drop frames and stall the RoCE transfer. Because of the 95ms RTT link on the receiver side of the policer, this limits

the achievable transfer rate to ≈ 100 Mbps for all queue depths.

Again, when we rate limit the application, achieving the bottleneck rate is straightforward as can be seen in Figure 3(c). Once enough messages are "in flight" to reach the target rate, the application simply waits before sending additional messages. One caveat is that the sender is limited to messages that are bound by the available buffer space in the policer. Intuitively, sending a message greater than 1 MB will simply overrun the buffer preventing the message from being completely received. We note that pacing in RDMA applications is at the granularity of IB messages, not at the link layer directly.

3.6.3 The Importance of Flow Control

An open question regarding the use of RoCE in the WAN has been its ability to share available bandwidth with competing traffic. Since the InfiniBand transport has no congestion control, one would intuitively expect that congestion due to other traffic along the path would negatively impact RoCE transfers if some mechanism did not prevent the NIC from sending at the available line rate. In fact, this is the observed behavior along shared paths when competing flows are introduced, and a key motivator for dynamic circuit services that can provide dedicated end-to-end bandwidth for RoCE flows.

At the same time, lossless Ethernet works in data centers in the form of converged Ethernet, and the RoCE implementation makes use of link layer flow control to avoid overrunning congested network segments. We can take advantage of the flow control mechanism to allow competing flows to share WAN paths, and it happens that RoCE is willing to allow a first-hop switch to provide pushback when competing traffic is present and causing contention for available network bandwidth. This pushback mechanism, known as a PAUSE frame, is essentially an Ethernet, link layer, signal to an attached network element that informs it to stop sending for a given time interval. In this manner, a number of simultaneous flows can be supported with effectively equal bandwidth sharing at an aggregation switch. What follows are the results of our evaluation that tests the ability of RoCE to share WAN paths while competing with both RoCE and TCP traffic.

In Figure 4, we demonstrate the ability of a single RoCE flow to compete with another RoCE flow and four parallel TCP flows, with and without flow control enabled. No additional latency was emulated in this experiment. Each test involved starting one or more flows across the XGEM between a pair of hosts, then starting a single RoCE flow between another pair of hosts that traverse the same shared path. Figures 4(a) and 4(c) show that RoCE with flow control enabled on all active switch ports will result in a nearly equal distribution of available bandwidth to active senders when competing with both RoCE and parallel TCP flows, with the parallel TCP flows taking a slight edge. With flow control disabled on the switch, achievable RoCE performance varies greatly between tests. In Figure 4(b), two RoCE flows compete in a haphazard manner, the introduced RoCE flow sometimes exceeding the already established flow depending on the message size and queue depth. Figure 4(d) indicates that a newly introduced RoCE flow cannot significantly outpace the rate of four already established TCP flows.

We repeated our experiment at 95ms RTT and the results are shown in Figure 5. Again, with flow control enabled (Figure 5(a)) RoCE is able to share the available bandwidth with the existing flow, now along the high-latency path. With flow control disabled (Figure 5(b)), we see a pattern similar to that of the TCP case with no latency. The newly introduced RoCE flow is unable to perform well with an established flow already filling network buffers over the high-latency path. The 95ms RTT case for four parallel TCP flows exhibits nearly the same performance characteristics as the single competing RoCE flow case and is omitted for conciseness.

In both the latency and non-latency cases, we see a pattern of certain message sizes having more success; for example, 64 KB and 4 MB in the no-latency and added latency cases, respectively. We speculate that this behavior is an artifact of the buffering that exists within the 10G switch and the XGEM device. In general, however, the behavior of RoCE flows is erratic when competing with other flows in the absence of flow control. A limitation of this scheme is that PAUSE frames do not typically propagate across network segments. Thus, if the congested segment is not the first hop from the sender, a PAUSE frame is never received and the RoCE transfer is likely to stall or otherwise exhibit unexpected behavior.

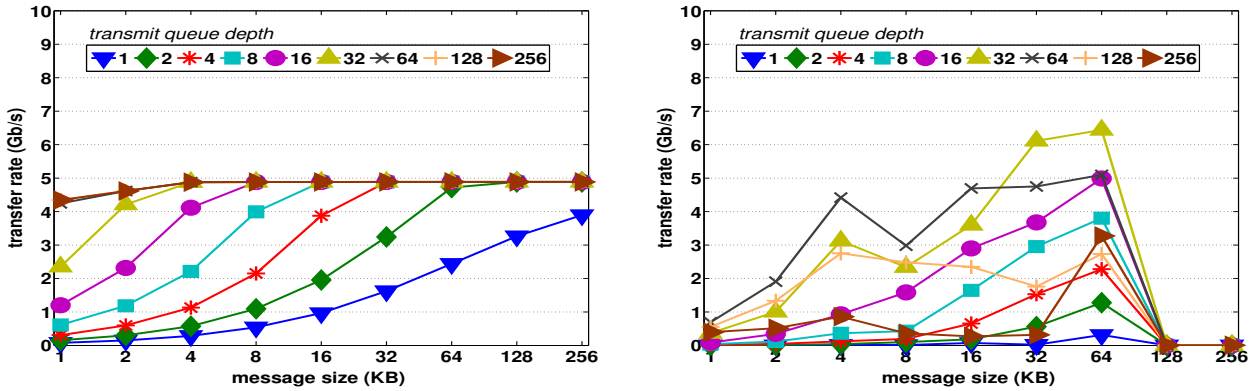
4. RELATED WORK

Being a recently proposed standard, there has been relatively little previous research in analyzing and characterizing RoCE performance over existing Ethernet infrastructure. A number of other RDMA and *zero-copy* protocols not involving InfiniBand (IB) have been proposed to run over Ethernet. These include technologies such as Intel's Direct Ethernet Transport (DET) [8] and approaches that use iWARP-enabled NICs [13]. Compared to RoCE and Infiniband, DET does not provide full OS-bypass functionality with limited hardware support, while iWARP remains bound to the limitations of TCP/IP over high-latency paths.

At the same time, there have been active efforts involved with extending IB fabrics over WANs [24] and comparisons of IB to existing 10G Ethernet in high-latency transfer scenarios [25]. These evaluations rely on IB extension devices which limit WAN performance to approximately 8 Gbs, whereas our approach shows that RoCE can easily saturate existing 10G networks. Other related work has investigated RDMA-capable storage protocols over WANs [29] and explored system-level benefits of RDMA interfaces over 10G networks [9]. More recently there has been work to make a file transfer tool based on RDMA verbs [26]. This open source tool, called *rftp*, is now available. A performance study of the RDMA API calls and how to use them has been done as well [21].

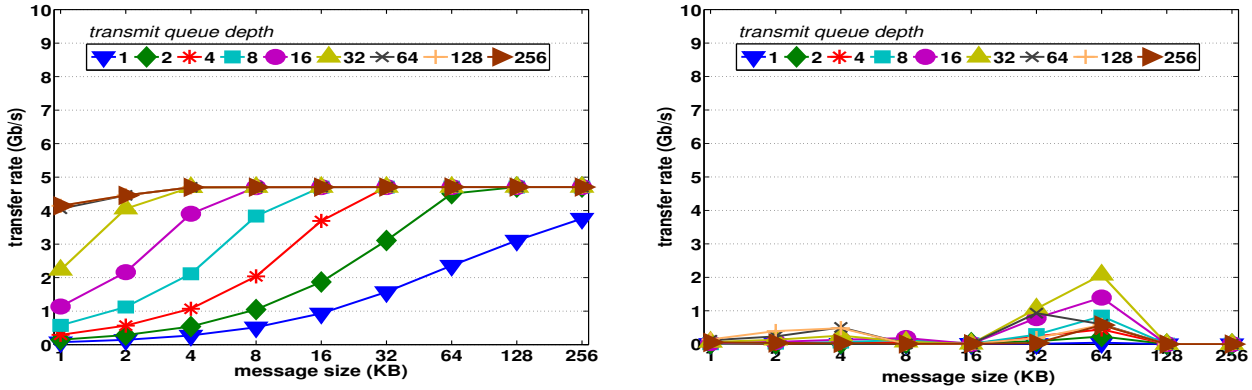
5. CONCLUSION

While RDMA over the WAN works well in our testing, there are a number of considerations in the use of wide area RDMA. As we have observed, RoCE is highly efficient, but sensitive. Our work explores and quantifies the particulars, but at high level these results are exactly what one would



(a) RoCE x1 with flow control

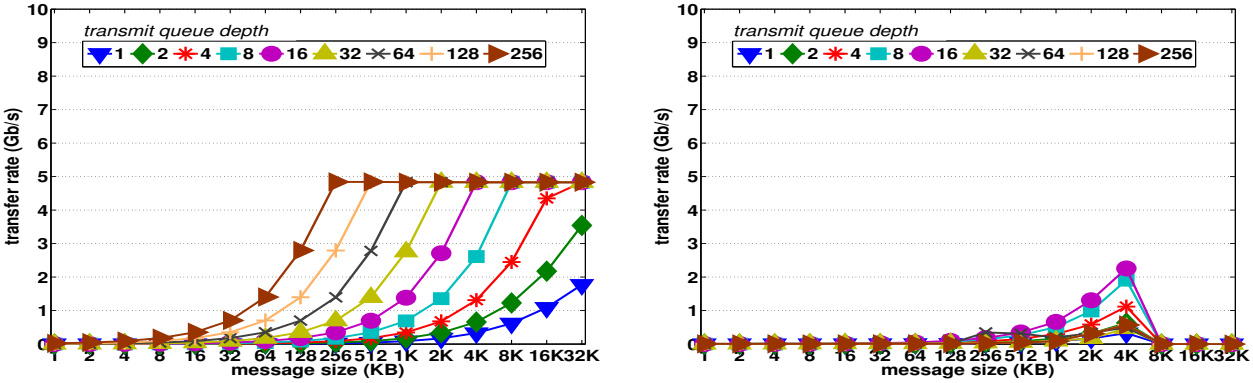
(b) RoCE x1 without flow control



(c) TCP x4 with flow control

(d) TCP x4 without flow control

Figure 4: Effect of competing traffic on RoCE performance, with and without flow control. No added latency.



(a) RoCE x1 with flow control

(b) RoCE x1 without flow control

Figure 5: Effect of competing traffic on RoCE performance, with and without flow control. With 95ms RTT.

expect. A general purpose entity can perform in a wider range of circumstances, but with less efficiency than a highly specialized one.

Loss has a significant impact on RoCE. One quantifiable observation is that it is important that the first hop switch or router have enough buffering. Recommended router buffers are $B = RTT \times C$ where C is the data rate of the link [28]. However, for single flows using RoCE without rate limiting, the device needs a full bandwidth-delay product worth of buffering if a bottleneck segment exists in the network.

Finally, we have observed that while RoCE can share a circuit with multiple flows and back off gracefully if the bottleneck is the first device in the path, if the bottleneck is several hops away, this creates a problem. Either there must be some out of band mechanism to provide feedback to the sender to throttle, or a mechanism to dynamically increase the circuit bandwidth to eliminate the bottleneck.

6. ACKNOWLEDGMENTS

This research used resources of the ESnet Testbed, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-05CH11231.

7. REFERENCES

[1] Aspera fasp. http://asperasoft.com/technology/transport/fasp/.

[2] Connectx-2 EN with RDMA over Ethernet (RoCE). http://www.mellanox.com/related-docs/prod_software/ConnectX-2_RDMA_RoCE.pdf.

[3] Gem: Network impairment emulator. http://www.spirent.com/Solutions-Directory/Impairments_GEM.

[4] Infiniband trade association. http://www.infinibandta.org.

[5] Ofed: Open fabrics enterprise distribution. https://openfabrics.org/ofed-for-linux-ofed-for-windows/ofed-overview.html.

[6] Results from nasa high end computing (hec) wan file transfer experiments/demonstrations super computing 2012 (sc12). http://science.gsfc.nasa.gov/606.1/docs/SC12.pdf.

[7] Infiniband. architecture specification release 1.2.1 annex a16: Roce. Infiniband Trade Association, 2010.

[8] Intel direct ethernet transport (det). http://software.intel.com/en-us/articles/intel-direct-ethernet-transport, 2010.

[9] P. Balaji. Sockets vs rdma interface over 10-gigabit networks: An in-depth analysis of the memory traffic bottleneck. In *In RAIT workshop .04*, page 2004, 2004.

[10] C. Barakat, E. Altman, and W. Dabbous. On tcp performance in a heterogeneous network: A survey. *IEEE Communications Magazine*, 38:40–46, 2000.

[11] A. Barczyk, A. Mughal, and et. al. Efficient lhc data distribution across 100gbps networks. In *Proceedings of the 2012 ACM/IEEE international Conference For High Performance Computing, Networking, Storage and Analysis, SC Companion*, Nov 2012.

[12] Biological and Environmental Research Network Requirements. Office of Biological and Environmental Research (BER), DOE Office of Science and the Energy Sciences Network, 2012.

[13] D. Dalessandro and P. Wyckoff. A performance analysis of the ammasso rdma enabled ethernet adapter and its iwarp api. In *In Proceedings of the IEEE Cluster 2005 Conference, RAIT Workshop*, 2005.

[14] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski. The science dmz: A network design pattern for data-intensive science. In *IEEE/ACM Annual SuperComputing Conference (SC13)*, Denver CO, USA, 2013.

[15] A. C. de Melo. The new linux 'perf' tools. http://vger.kernel.org/~acme/perf/lk2010-perf-paper.pdf, 2010.

[16] GridFTP. http://www.globus.org/datagrid/gridftp.html.

[17] Y. Gu and R. Grossman. Udt: Udp-based data transfer for high-speed wide area networks. *Computer Networks (Elsevier) Volume 51, Issue 7*, 2007.

[18] V. Jacobsen, R. Braden, and D. Borman. Tcp extensions for high performance. RFC 1323, May 1992.

[19] R. Kettimuthu, W. Liu, J. M. Link, and J. Bresnahan. A gridftp transport driver for globus xio. In *PDPTA*, pages 843–849, 2008.

[20] E. Kissel and M. Swany. Evaluating high performance data transfer with rdma-based protocols in wide-area networks. In *Proceedings of the 14th IEEE International Conference on High Performance Computing and Communications (HPCC-12)*, 2012.

[21] P. MacArthur and R. D. Russell. A performance study to guide rdma programming decisions. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS)*, June 2012.

[22] S. Molnár, B. Sonkoly, and T. A. Trinh. A comprehensive tcp fairness analysis in high speed networks. *Comput. Commun.*, 32(13-14):1460–1484, 2009.

[23] I. Monga, C. Guok, W. E. Johnston, and B. Tierney. Hybrid networks: Lessons learned and future challenges based on esnet4 experience. In *IEEE Communications Magazine*, May 2011.

[24] S. Narravula, H. Subramoni, P. Lai, R. Noronha, and D. K. Panda. Performance of hpc middleware over infiniband wan. In *Proceedings of the 2008 37th International Conference on Parallel Processing*, ICPP '08, pages 304–311, Washington, DC, USA, 2008. IEEE Computer Society.

[25] N. S. V. Rao, W. Yu, W. R. Wing, S. W. Poole, and J. S. Vetter. Wide-area performance profiling of 10gige and infiniband technologies. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, SC '08, pages 14:1–14:12, Piscataway, NJ, USA, 2008. IEEE Press.

[26] Y. Ren, T. Li, D. Yu, S. Jin, T. Robertazzi, B. Tierney, and E. Pouyoul. Protocols for wide-area data-intensive applications: Design and performance issues. In *Proceedings of the 2012 ACM/IEEE international Conference For High Performance Computing, Networking, Storage and Analysis*, Nov 2012.

[27] B. Tierney, E. Kissel, M. Swany, and E. Pouyoul. Efficient data transfer protocols for big data. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–9, 2012.

[28] C. Villamizar and C. Song. High performance tcp in ansnet. *SIGCOMM Comput. Commun. Rev.*, 24(5):45–60, Oct. 1994.

[29] W. Yu, N. Rao, P. Wyckoff, and J. Vetter. Performance of rdma-capable storage protocols on wide-area network. In *3rd Petascale Data Storage Workshop PSDW 2008*, 2008.

[30] W. Yu, Y. Tian, J. Vetter, N. S. Rao, T. Liu, and G. Shainer. Exio: Enabling globus on rdma networks - a case study with infiniband. Technical Report AU-CSSE-PASL/10-TR01, PASL, Dept. of CDDSSE, Auburn University, 2010.

[31] W. Yu, Y. Tian, and J. S. Vetter. Efficient zero-copy noncontiguous i/o for globus on infiniband. In *Proceedings of the 2010 39th International Conference on Parallel Processing Workshops*, ICPPW '10, pages 362–368, Washington, DC, USA, 2010. IEEE Computer Society.