

Performance Evaluation of RDMA over IP: A Case Study with the Ammasso Gigabit Ethernet NIC*

Hyun-Wook Jin

Sundeep Narravula
Pavan Balaji

Gregory Brown
Dhabaleswar K. Panda

Karthikeyan Vaidyanathan

Department of Computer Science and Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210

{jinhy, narravul, browngre, vaidyana, balaji, panda}@cse.ohio-state.edu

Abstract

Remote Direct Memory Access (RDMA) has been proposed to overcome the limitations of traditional send/receive based communication protocols like sockets. The immense potential of RDMA to improve the communication performance while being extremely conservative on resource requirements has made RDMA the most sought after feature in current and next generation networks. Recently, there are many active efforts to enable RDMA over IP networks and fabrication of RDMA-enabled Ethernet NICs. However, the performance of RDMA over IP networks has not been quantitatively evaluated over WAN environments yet. In this paper, we evaluate the performance of RDMA over IP networks with the Ammasso Gigabit Ethernet NIC while emulating high delay WANs and varying load on remote node. We observe that RDMA is beneficial especially under heavy load conditions. More importantly, even with a high delay, RDMA can provide better communication progress and requires less CPU resources as compared to the traditional sockets over TCP/IP. Further, we show that RDMA can support high performance intra-cluster communication while providing a unified communication interface for inter- and intra-cluster communication. To the best of our knowledge, this is the first quantitative study of RDMA over IP on a WAN setup.

Keywords: *RDMA over IP, Gigabit Ethernet, WAN Emulation, Cluster of Clusters, and Grid*

* This research is supported in part by Department of Energy's Grant #DE-FC02-01ER25506, National Science Foundation's grants #CCR-0204429, #CCR-0311542, and CNS-0403342; and equipment donations from Ammasso, Inc. and Foundry Networks.

1. Introduction

Remote Direct Memory Access (RDMA) is emerging as the central feature in modern network interconnects. It has been proposed by the researchers to overcome the limitations of traditional communication protocols such as sockets over TCP/IP suite. Although there have been many studies aimed at the performance improvement of the sockets over TCP/IP [13, 6, 21, 27], inherent limitations of the sockets obstruct achieving high-performance communication on high-end servers. RDMA potentially improves the communication performance while being extremely conservative on resource utilization, and hence has become the most sought after feature in current and next generation networks. Interconnects like InfiniBand [3], Myrinet [5] and Quadrics [18] have long introduced RDMA in LAN environments.

RDMA over IP has been developed to extend the benefits of RDMA beyond the LAN environments and across the WAN/Internet. The RDMA consortium [7] has proposed the RDMA Protocol Verbs Specifications (RDMAVS 1.0) [10] to standardize the efforts. Several performance critical grid applications like GridFTP [19], GridRPC [16], etc. can benefit significantly by utilizing RDMA over IP. Further, web based systems like web-servers, data-centers, Internet proxies, etc. can also leverage the benefits of an IP based RDMA protocol for use over the WAN.

To enable existing and new WAN applications to effectively utilize RDMA, researchers have done qualitative studies [22, 8]. While the benefits of RDMA have been highlighted in LAN environments, there have been no comprehensive quantitative evaluations of RDMA over WAN environments [2, 17, 4, 26]. There also has been research on utilizing RDMA for IP network storage [15] but this

does not take into account the high delay characteristics of WAN. Hence it has become very critical to evaluate and analyze the performance of RDMA over IP in a WAN environment.

In this paper, we evaluate RDMA over IP in WAN environments with the Ammasso Gigabit Ethernet Network Interface Card (NIC) [1] in several aspects of performance such as (i) basic communication latency, (ii) computation and communication overlap, (iii) communication progress, (iv) CPU resource requirements, and (v) unification of communication interface. These performance metrics and features are known to be important to user applications. We especially focus on the impact of larger delays often experienced for WAN communications upon these important factors.

In order to emulate the WAN environment, we have constructed two different IP networks connected by a workstation-based router. In addition, we have implemented a delay generator named *degen* on the router, which adds a specified delay to the network at the kernel level, characterizing WAN.

Our experimental results clearly show that, even with a high delay, RDMA can provide better communication progress and less CPU resource requirements as compared to the traditional sockets over TCP/IP. We also observe that RDMA is beneficial especially under loaded conditions. We further show that RDMA can support high performance intra-cluster communication while providing a unified communication interface for inter- and intra-cluster communication. To the best of our knowledge, this is the first quantitative study of RDMA over IP on a WAN setup.

The rest of the paper is organized as follows: In Section 2, we describe an overview of RDMA and the Ammasso Gigabit Ethernet NIC. Section 3 briefly describes the design of our WAN emulator - *degen* - and the system configurations. Section 4 describes our experimental methodologies and performance evaluation results. Section 5 addresses a few relevant issues regarding RDMA over IP. Finally, we conclude the paper in Section 6.

2. Background

In this section, we briefly describe the necessary background information. We first look at Remote Direct Memory Access (RDMA), followed by an overview of the Ammasso Gigabit Ethernet NIC used.

2.1. Remote Direct Memory Access

Remote Direct Memory Access (RDMA) is a modern network feature that allows nodes to communicate without any involvement of remote node’s CPU. Two basic RDMA operations exist: (i) RDMA Write and (ii) RDMA Read.

RDMA write is used to transfer data to a remote node’s memory and RDMA read is used to get data from a remote node. In a typical RDMA operation, the initiator node posts a descriptor containing the details of the data-transfer, to the NIC and the NIC handles the actual data-transfer asynchronously. On the remote side, the NIC stores/fetches the data to/from the host memory without disturbing the CPU.

The usage of RDMA presents multi-fold benefits to enable application scalability. Since the sender and receiver CPU’s are not involved in RDMA data-transfers, the applications benefit from this additional computing capability. Further, the elimination of kernel context switches and multiple copies during the data-transfers provides significantly better utilization of memory bandwidth and CPU. Interconnects like InfiniBand [3], Myrinet [5] and Quadrics [18] have introduced RDMA in LAN environments. In addition, several protocols have been proposed to take advantage of RDMA operations on IP networks [20, 23, 14].

2.2. Overview of Ammasso Gigabit Ethernet NIC

The Ammasso Gigabit Ethernet NIC [1] provides an implementation of the RDMA over TCP/IP enabled NIC. Based on the RDMA Protocol Verbs (RDMAVS 1.0) [10] specified by the RDMA consortium, the RDMA interface of the Ammasso Gigabit Ethernet NIC provides low latency and high bandwidth on Gigabit Ethernet network. As shown in Figure 1, Ammasso Gigabit Ethernet NIC supports the legacy sockets interface and the Cluster Core Interface Language (CCIL) interface.

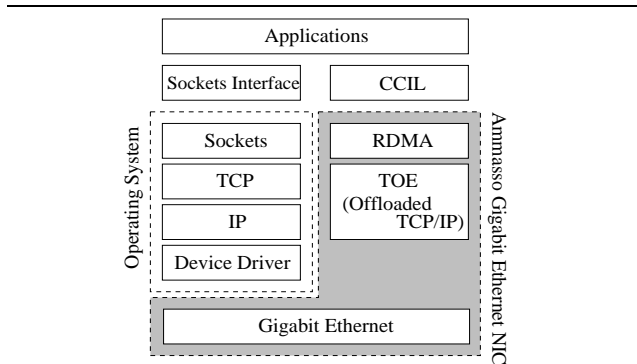


Figure 1. Protocol Stacks on Ammasso Gigabit Ethernet NIC

The CCIL interface is an implementation of the Verbs layer to utilize RDMA over IP. The CCIL interface uses the RDMA layer and offloaded TCP/IP on the NIC to transmit

the data. On the other hand, the sockets interface still sends and receives the data through the traditional TCP/IP implemented in the operating system kernel. The CCIL interface enables zero-copy and kernel-bypass data transmission.

3. Experimental WAN Setup

Our performance evaluations have been performed on the experimental system shown in Figure 2, where we have two different IP networks and they are connected through a workstation-based router emulating WAN. The end nodes are SuperMicro SUPER P4DL6 nodes - each has dual Intel Xeon 2.4GHz processors with a 512KB L2 cache and an Ammasso 1100 Gigabit Ethernet NIC. The router node is a SuperMicro SUPER X5DL8-GG workstation with dual Intel Xeon 3.0GHz processors, 512KB L2 cache, and 2GB of main memory. The router node is connected to IP networks A and B with Broadcom BCM5703 and Intel PRO/1000 Gigabit Ethernet NICs, respectively. All nodes use Linux kernel version 2.4.20. The switches used for each IP network are Foundry FastIron Edge X448 Switch and Netgear GS524T Gigabit Switch, respectively.

To reflect the characteristics of high latency in the WAN environment, we have implemented a delay generator named *degen*. It delays the forwarding of each packet on the router by a given value after the corresponding routing decision has taken place as shown in Figure 2. We use the `netfilter` hooks provided by Linux 2.4 kernel to implement *degen*, which can be dynamically inserted to the chain of packet processing by using a run-time loadable kernel module.

Each packet is time-stamped when it reaches the router by the kernel. *Degen* uses this time stamp to delay the packet appropriately. In our design, we avoid adding cascading delays for consecutive packets that arrive in a burst mode by pulling these packets out of the network and maintaining them in a queue. These packets are reinjected into the network at an appropriate time as needed.

4. Evaluation of RDMA over IP Networks

In this section, we compare RDMA with traditional TCP/IP sockets on WAN environments with respect to (i) latency, (ii) computation and communication overlap, (iii) communication progress, (iv) CPU resource requirements, and (v) unification of communication interface.

4.1. Basic Communication Latency

The basic communication latency is one of the most important performance metrics. In this section, we carry out the latency test in a standard ping-pong fashion to report

one-way latency. The client sends a message and waits for a reply message of the same size from the server. The time for this is recorded by the client and it is divided by two to find out one-way latency. In the case of RDMA, we use RDMA write to transmit the data.

Figure 3(a) shows the results of latency without the delay by *degen*. We can see that the latencies of RDMA and sockets are almost the same regardless of the message size even without the delay. It is because the latency added by the default experimental setup described in Section 3 is relatively large compared to the overhead on the end nodes. Although RDMA achieves a zero-copy data transmission, since the MTU size is only 1500 Bytes, we cannot expect a large benefit with respect to the latency. In the case of messages larger than the MTU size, TCP constructs several segments so that each segment can fit into the MTU sized IP fragment. Hence the transmission of the segments are pipelined and we can obtain the benefit of the zero-copy only for the first segment in a high delay environment. However, if the network delay is smaller than the processing overhead of the end node, the zero-copy transmission helps to deliver low latency.

Figure 3(b) shows the latency varying the network delay with 1KB message. As we can see, RDMA and sockets report almost the same latency. This reveals that the basic communication latency is not an important metric to distinguish RDMA from the sockets in a high delay WAN environment because the overheads on the end nodes is too small compared with the network delay in the order of milliseconds.

4.2. Computation and Communication Overlap

In this section, we evaluate how well the process is able to overlap computation with communication. In our test shown in Figure 4(a), the client performs a computation loop that does a dummy work for a given time in between ping (i.e., sending) and pong (i.e., receiving) operations of the latency test described in Section 4.1. We evaluate the computation and communication overlap ratio with $(Computation_Time)/(Total_Time)$ on the client side. Thus a value closer to 1 represents a better computation and communication overlap. It is to be noted that our sockets latency test is using non-blocking sockets to maximize the computation and communication overlapping.

Figure 5(a) shows the overlap ratios with varying computation time and without network delay. The message size used is 1KB. As we can see, RDMA can achieve better overlap even with smaller computation time compared to the sockets. This is because RDMA provides asynchronous communication interface. In addition, we do not need to utilize CPU resources to get the data from remote node because the remote node uses RDMA write. On the other

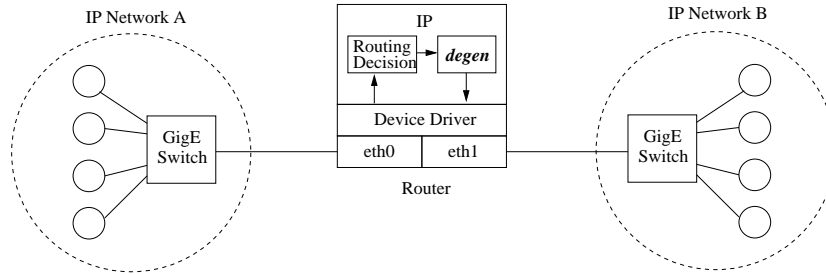


Figure 2. Experimental WAN Setup

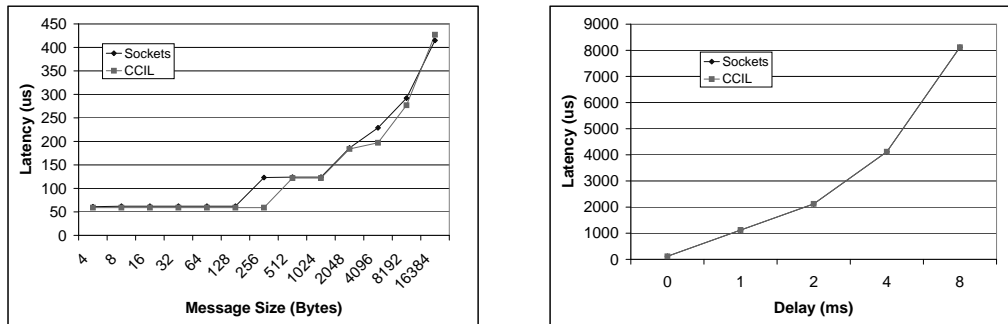


Figure 3. Communication Latency: (a) Varying Message Size and (b) Varying Network Delay

hand, in the case of the sockets, some of the receiving functions of the sockets (e. g., kernel buffer to user buffer copies) are performed in the context of the application process. As a result, these receiving functions cannot be overlapped with actual application processing while some other functions (e. g., interrupt handling and bottom half) can run in parallel with the application processing by another CPU in a SMP system. Further, the offloaded TCP/IP of the RDMA case increases the chance of overlapping between packet processing overhead and computation overhead.

Figure 5(b) shows the overlap ratio values of RDMA and sockets for varying network delay, where we have fixed the computation time to 242ms and message size to 1KB. It can be observed that the difference between RDMA and sockets reduces with large network delays. It is mainly because the network delay is the dominant overhead and it can be overlapped with computation time regardless of RDMA or sockets. Since the packet processing overhead of end nodes is not a critical overhead anymore on a high delay WAN, its overlapping with other overheads does not affect much to the overlap ratio. However, still we can see that RDMA can provide better overlap than sockets for delays in order of a few milliseconds.

4.3. Communication Progress

In many distributed systems, we often observe the communication pattern that a node requests some data to a remote node and it returns the data. This operation can be implemented with either by using a pair of send and receive calls or by using RDMA read. Moreover, the remote node can be heavily loaded because of burst requests on the data or CPU intensive computations. To compare the performance of RDMA read with the traditional sockets in this scenario, we simulate the load on the remote node by adding a dummy loop running for a given time. We measure the latency to get 1KB of data from remote node as shown in Figure 4(b).

Figure 6(a) shows the data fetching latency varying the load on the remote node, where the load is represented as the response delay. Since RDMA read does not require any involvement of remote process for data transmission, it can read data from remote memory without any impact from the load on the target. It is to be noted that the sockets interface is not able to deliver good performance as the load increases. It is because the overall communication progress of the sockets highly depends on that of both sides (sender and receiver).

Figure 6(b) shows the data fetching latency varying the network delay for a fixed response load of 16ms. With in-

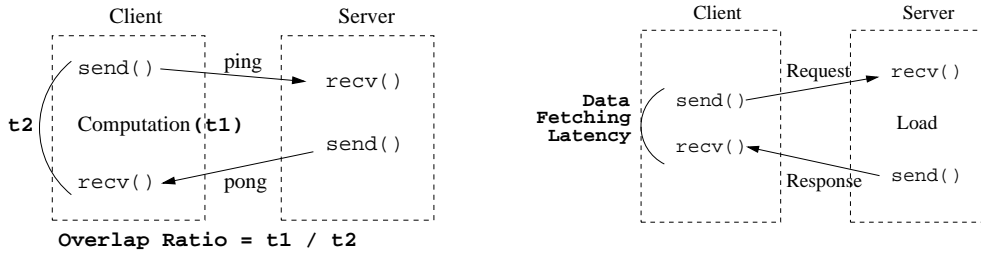


Figure 4. Pseudo Code for Benchmarks: (a) Overlap Test and (b) Communication Progress Test

crease in the network delay, both of the interfaces (sockets and RDMA) perform similarly because the network delay tends to dominate the performance costs more than the load on the remote node. However, it can still be seen that RDMA is more tolerant of the network delay achieving better communication progress.

4.4. CPU Resource Requirements

To measure the effect of CPU resource requirements for communication on the application performance, in this experiment, we run an application on a server node that performs basic mathematical computations while 40 clients continuously send data to this server. We report the total execution time of the application under this scenario.

Figure 7(a) shows the application execution time varying the message size with no added network delay. As we can see, the execution time with background sockets communications is very high for all message sizes while it does not have any significant performance impact with background RDMA communication.

Figure 7(b) shows the execution time of the application varying the network delay with 16KB message size. We can observe that the background sockets communication significantly degrades the application performance even on high delay WANs. The reason is that, in the case of sockets, the remote CPU is involved in the communication effort, with packet processing and interrupt handling at the kernel level and receive request posting at the application level. This results in stealing of the CPU resource from the computing application. However, RDMA can place the data to the remote memory without any CPU requirement on the remote node. Hence we can see in the figure that the application execution time is not affected by RDMA and constant for all network delays. This reveals that RDMA has a strong potential of saving the CPU resource on the server side even on a high delay WAN environment.

4.5. Unification of Communication Interface

In addition to direct performance metrics detailed in the previous sections, WAN and LAN interoperability is a very

important feature of RDMA over IP. Scenarios in which several inter-cluster and intra-cluster nodes communicate with each other need common communication interfaces for the job. Traditionally, the sockets over TCP/IP has been the main interface with this feature. However, with RDMA over IP, this interoperability can be achieved and it can be achieved with all the benefits described in the previous sections. Further, RDMA over IP performs significantly better than sockets for within LAN communications.

Figure 8 shows the latency of CCIL and Sockets communications within a LAN. We see that the small message latency differs by almost 50% with RDMA being better. Hence, RDMA over IP benefits these multi-cluster applications with better communication both over the WAN as well as in the LAN. It is to be noted that the benefit of zero-copy with RDMA is not significant even in the LAN as we have discussed in Section 4.1. However, 10 Gigabit Ethernet [11] is expected to provide very low propagation delay within LAN and show the benefit of zero-copy on the communication latency with large messages.

5. Discussion

In the previous section, we have compared the basic RDMA with sockets in which we do not deal with the address exchange phase for the RDMA communication. It is, however, required to exchange the memory information of the remote node before the RDMA communication can take place. Some applications or middleware perform the address exchange at the initialization phase and use the same set of buffers for all subsequent communications [17, 26, 4, 25, 12]; thereby, the cost of address exchange in data communication is avoided.

On the other hand, other applications or middleware may need to exchange the remote buffer information for each RDMA communication operation. This can be implemented with a rendezvous protocol as shown in Figure 9 [24]. Figure 9(a) shows RDMA write based rendezvous protocol, where the sending process first sends a control message (RNDZ_START) to the receiver. The receiver replies to the sender using another control message

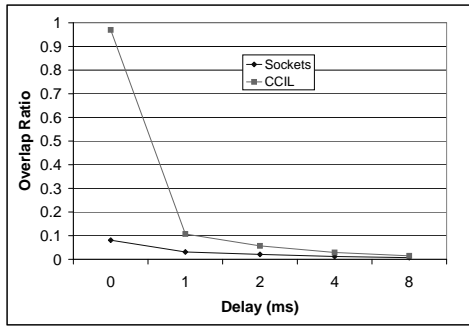
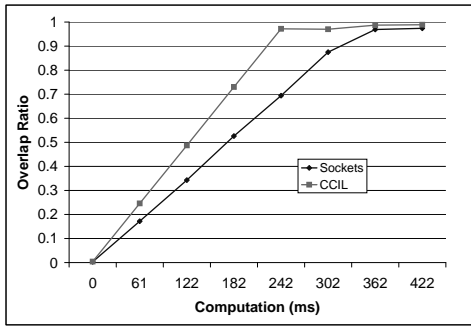


Figure 5. Overlap Ratio: (a) Varying Computation Time and (b) Varying Network Delay

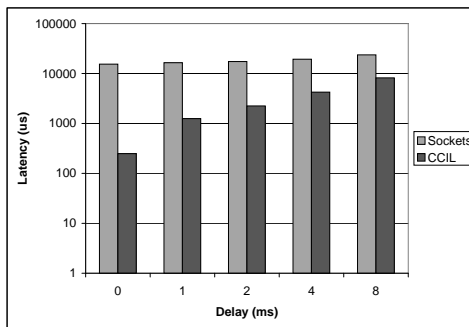
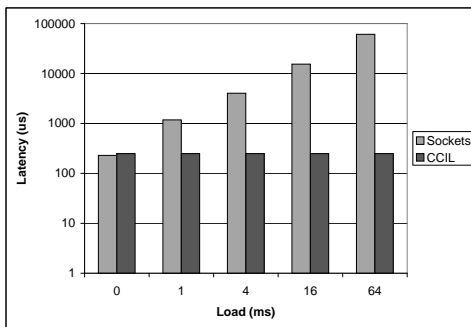


Figure 6. Communication Progress: Data Fetching Latency (a) Varying Load and (b) Varying Network Delay

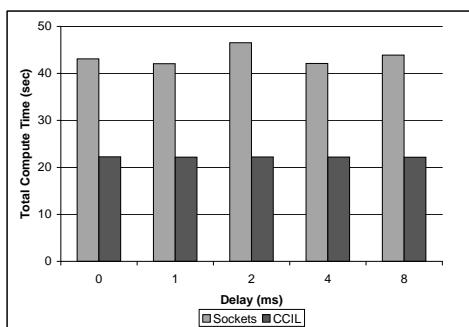
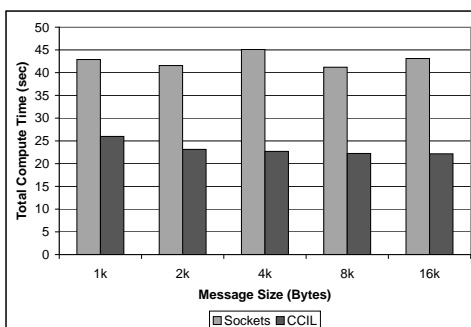


Figure 7. Impact on Application Execution Time: (a) Varying Message Size and (b) Varying Network Delay

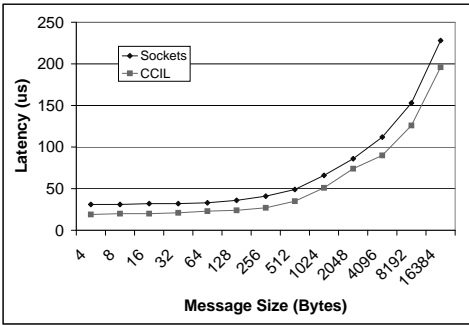


Figure 8. Communication Latency within a LAN with varying Message Sizes

(RNDZ_REPLY). This reply message contains the required information for the receiving application’s buffer. The sending process then sends a message directly to the remote buffer using RDMA write followed by another control message (FIN) which indicates to the receiver that the message has been placed in the specified application buffer. Figure 9(b) shows RDMA read based rendezvous protocol. The sending process sends the RNDZ_START message, which contains the address for sending application’s buffer. Upon its discovery, the receiving process issues RDMA read. When it is done, the receiver informs the sending process by a FIN message.

Since the rendezvous protocol avoids a copy operation on the data, it can provide a low latency with large message sizes on a high-speed link. However, on a high delay link, due to the control messages required for address exchange, the communication latency would be higher if the address exchange is required for each communication operation. Thus applications need to consider the trade off between copy overhead and address exchange overhead to decide between RDMA and sockets in a high delay WAN.

It is also to be noted that some middleware are still using the rendezvous protocol although they are implemented over the sockets [9]. The main reason of this is that in many cases a copy operation is needed between user buffer and middleware buffer and the rendezvous protocol can remove this copy operation. In this case, the communication latency may not show much difference between RDMA and sockets because we need control messages for both. However, RDMA can still achieve most of the benefits described in the previous section.

6. Conclusions and Future Work

To resolve the limitations of existing communication semantics of send and receive based protocols like sockets, RDMA has been proposed in modern high-speed LAN in-

terconnects. Moreover, recently there are many active efforts to enable RDMA over IP networks. However, its performance has not been quantitatively evaluated over WAN environments. In this paper, we have evaluated the performance of RDMA in WAN environments with the Ammasso Gigabit Ethernet NIC.

We have emulated the high delay characteristics of WAN environments with a workstation-based router with the help of our kernel module *degen*. Our experimental results have revealed that, even with a high delay, RDMA can achieve better communication progress and can save CPU resources as compared to the traditional sockets over TCP/IP. This is mainly due to the fact that RDMA does not require involvement of remote side CPU. The Offloaded TCP/IP further leverages the benefit of RDMA over IP. We have presented the potential benefits of RDMA over IP networks through comprehensive performance evaluations.

We intend to continue working in this direction. We plan to improve the delay generator (*degen*) to reflect other characteristics of WANs and evaluate the performance of RDMA over IP with more applications. We also plan to design and evaluate RDMA-aware middleware for widely distributed systems over WAN.

References

- [1] Ammasso, inc. <http://www.ammasso.com>.
- [2] MPI over InfiniBand Project. <http://nowlab.cis.ohio-state.edu/projects/mpi-iba/>.
- [3] Infiniband Trade Association. <http://www.infinibandta.org>.
- [4] P. Balaji, K. Vaidyanathan, S. Naravula, K. Savitha, H.-W. Jin, and D. K. Panda. Exploiting Remote Memory Operations to Design Efficient Reconfiguration for Shared Data-Centers. In *Workshop on Remote Direct Memory Access (RDMA): Applications, Implementations, and Technologies (RAIT)*, San Diego, CA, Sep 2004.
- [5] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W. K. Su. Myrinet: A Gigabit-per-Second Local Area Network. *IEEE-Micro*, 15(1):29–36, February 1995.
- [6] J. Chase, A. Gallatin, and K. Yocum. End-System Optimizations for High-Speed TCP. *IEEE Communications Magazine*, 39(4):68–75, April 2001.
- [7] RDMA Consortium. Architectural Specifications for RDMA over TCP/IP. <http://www.rdmaconsortium.org/>.
- [8] W. Feng, G. Hurwitz, H. Newman, S. Ravot, L. Cottrell, O. Martin, F. Coccetti, C. Jin, D. Wei, and S. Low. Optimizing 10-Gigabit Ethernet for Networks of Workstations, Clusters and Grids: A Case Study. In *SC2003: High-Performance Networking and Computing Conference*, November 2003.
- [9] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. High-performance, portable implementation of the MPI Message Passing Interface Standard. *Parallel Computing*, 22(6):789–828, 1996.

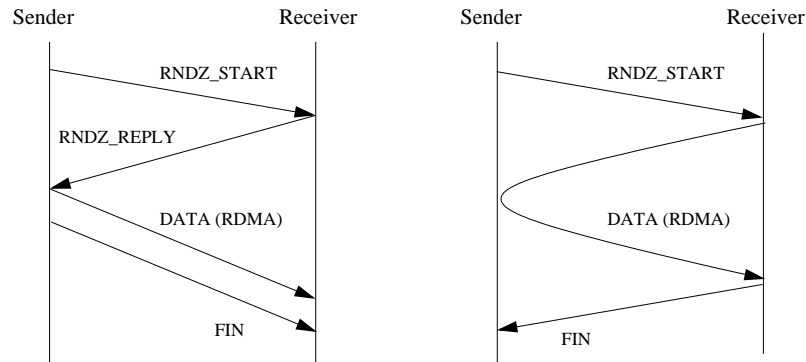


Figure 9. Rendezvous Protocols for Address Exchange: (a) RDMA Write based and (b) RDMA Read based Rendezvous Protocols

- [10] J. Hilland, P. Culley, J. Pinkerton, and R. Recio. RDMA Protocol Verbs Specification (Version 1.0). Technical report, RDMA Consortium, April 2003.
- [11] IEEE. IEEE Std 802.3ae-2002, Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation, August 2002.
- [12] W. Jiang, J. Liu, H.-W. Jin, D.K. Panda, W. Gropp, and R. Thakur. High Performance MPI-2 One-Sided Communication over InfiniBand. In *IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 04)*, April 2004.
- [13] H.-W. Jin, P. Balaji, C. Yoo, J.-Y. Choi, and D.K. Panda. Exploiting NIC Architectural Support for Enhancing IP based Protocols on High Performance Networks. *Journal of Parallel and Distributed Computing, Special Issue on Design and Performance of Networks for Super-, Cluster-, and Grid-Computing*, 2005. accepted for publication.
- [14] M. Ko, J. Hufferd, M. Chadalapaka, U. Elzur, H. Shah, and P. Thaler. iSCSI Extensions for RDMA Specification (Version 1.0). Technical report, RDMA Consortium, July 2003.
- [15] K. Magoutis, S. Addetia, A. Fedorova, and M. Seltzer. Making the Most out of Direct Access Network-Attached Storage. In *Second USENIX Conference on File and Storage Technologies (FAST'03)*, March 2003.
- [16] H. Nakada, S. Matsuoka, K. Seymour, J. Dongarra, C. Lee, and H. Casanova. A GridRPC Model and API. Technical report, GridRPC Working Group, November 2003.
- [17] S. Narravula, P. Balaji, K. Vaidyanathan, S. Krishnamoorthy, J. Wu, and D. K. Panda. Supporting Strong Coherency for Active Caches in Multi-Tier Data-Centers over InfiniBand. In *SAN*, 2004.
- [18] F. Petrini, W. C. Feng, A. Hoisie, S. Coll, and E. Frachtenberg. The Quadrics Network (QsNet): High-Performance Clustering Technology. In *Hot Interconnects*, 2001.
- [19] Globus Project. GridFTP: Universal Data Transfer for the Grid. Technical report, The University of Chicago and The University of Southern California, September 2000.
- [20] R. Recio, P. Culley, D. Garcia, and J. Hilland. An RDMA Protocol Specification (Version 1.0). Technical report, RDMA Consortium, October 2002.
- [21] G. Regnier, D. Minturn, G. McAlpine, V. A. Saletore, and A. Foong. ETA: Experience with an Intel Xeon Processor as a Packet Processing Engine. *IEEE Micro*, 24(1):24–31, January/February 2004.
- [22] A. Romanow. An Overview of RDMA over IP. In *First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003)*, February 2003.
- [23] H. Shah, J. Pinkerton, R. Recio, and P. Culley. Direct Data Placement over Reliable Transports. Internet Draft, February 2005.
- [24] S. Sur, H.-W. Jin, L. Chai, and D.K. Panda. RDMA Read Based Rendezvous Protocol for MPI over InfiniBand: Design Alternatives and Benefits. Technical Report OSU-CISRC-5/05-TR30, The Ohio State University, May 2005.
- [25] S. Sur, H.-W. Jin, and D.K. Panda. Efficient and Scalable All-to-All Exchange for InfiniBand-based Clusters. In *International Conference on Parallel Processing (ICPP-04)*, August 2004.
- [26] K. Vaidyanathan, H.-W. Jin, S. Narravula, and D. K. Panda. Accurate Load Monitoring for Cluster-based Web Data-Centers over RDMA-enabled Networks. Technical report, The Ohio State University, 2005.
- [27] E. Yeh, H. Chao, V. Mannem, J. Gervais, and B. Booth. Introduction to TCP/IP Offload Engine (TOE). 10 Gigabit Ethernet Alliance, May 2002.