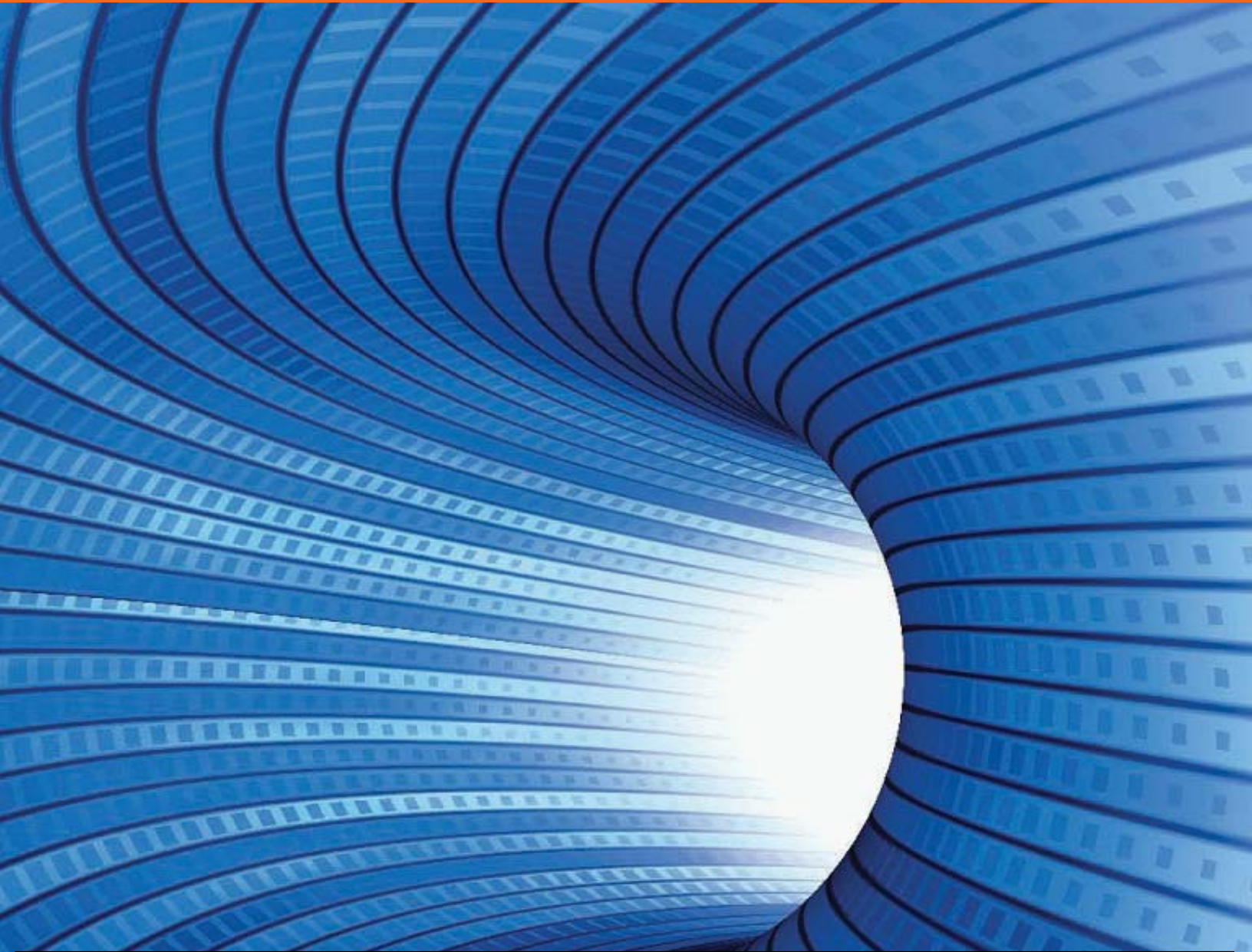


**Center for Performance
Evaluation of Cluster
Networking Technologies**



dice
PROGRAM



real testing | real data | real results



Signature Page, Release Acknowledgement

By signing below, the DICE collaborators certify that the contents presented in this document are accurate to the best of their knowledge.

A handwritten signature in blue ink that reads "Dhalbaleswar Panda".

Dr. Dhalbaleswar Panda
The Ohio State University

A handwritten signature in blue ink that reads "Tracey Wilson".

Tracey Wilson, FRB Chair

A handwritten signature in blue ink that reads "Al Stutz".

Al Stutz, Governance Board Chair

Project Title
Table of Contents

1	Project Executive Summary.....	4
2	Introduction.....	4
3	Project Overview/Description.....	5
4	Project Goals.....	5
5	Benchmarks and Evaluations.....	5
6	Evaluation Results.....	7
7	Analysis of Deviations from Predictions.....	9
8	Conclusions.....	10
9	Proposed Next Steps.....	10

1 Project Executive Summary

The federal government and vendors have been struggling with measuring and validating the performance of various interconnection technologies and methods for use in cluster supercomputers in a standardized manner. The objective of this project has been to design, develop and validate a set of standard measurement tools for evaluating cluster networking and I/O technologies. During Phase I of this grant (first year), a set of MPI-level benchmarks have been designed. These benchmarks have been evaluated on the DICE testbed: ASC cluster with two different networking technologies: InfiniBand SDR and Quadrics. Future plans include using these benchmarks on other DICE platforms: such as the Avetec cluster with InfiniBand DDR and 10 GigE. The results included in this report (interim Phase I) for the ASC cluster demonstrate the strengths of these benchmarks to compare performance (strengths and weaknesses) of various networking technologies and MPI libraries for clusters. As these benchmarks are being designed, developed and tested, they are also being made available to the community through OSU MVAPICH project web site as open-source benchmarks. These are also included in the OSU MVAPICH and MVAPICH2 distributions. The performance evaluation results of these benchmarks are also being made available. During the first year of this project, these benchmarks have had wide usage and acceptance in the HPC community. Current Open Fabrics Enterprise Distribution (OFED), the main organization behind open-source InfiniBand and 10GigE/iWARP development, include these benchmarks in its distribution. A large number of InfiniBand and iWARP vendors and several Linux distributors also include these benchmarks (known in the community as 'OSU Benchmarks') in their distributions. Future plans on this project include expanding these benchmarks for next generation multi-core and multi-threaded systems as well as I/O environments. It is anticipated that the benchmarks developed through this project will lead to the creation of a complete tool set and the creation of a National facility to evaluate all current and future cluster networking and I/O technologies and produce fair and unbiased reports. The project will significantly help in reducing cost for the agencies and the vendors to evaluate next generation clusters.

2 Introduction

Modern day supercomputers are built with commodity clusters of Shared Memory Multiprocessors (SMPs) or multi-core architectures, all interconnected with high-performance networks (such as InfiniBand, Myrinet, Quadrics or 1/10 GigE). Some of these networks are used for both inter-processor communication and I/O. Sometimes, different networking technologies are also used for networking and I/O. The cluster networking and I/O technologies play a crucial role in providing high-performance to end applications. However, determining the *right* networking and I/O technology which provides optimal performance for a given class of applications is not a trivial task since each application has differing computation and I/O characteristics. In order to enable accurate and fair comparison of end application performance, experimental evaluation must be carried out on homogenous platforms where the CPU, memory subsystems and I/O devices are common for each one of the networking and I/O technologies being considered. In addition to hardware, there are several software layers that provide the compile and runtime support to

the application. Each of these software layers, ranging from low level device drivers to higher level layers like MPI, address unique design challenges and provide innovative solutions. Several key metrics on which application performance is dependent are: overlap of computation and communication, scalability, parallel speedup and interaction of I/O and computation. Usually, these metrics are enmeshed with the design of these software layers and architecture of the networking and I/O technology. Thus, a careful and integrated study of the various layers involved with each technology is required for understanding the characteristics of a cluster interconnect and the kind of benefits it can provide to end applications in a given systems environment.

3 Project Overview/Description

The goals of the project (for the first year) were to design, develop and evaluate a set of MPI-level benchmarks which can demonstrate the strengths and weaknesses of the underlying networks and MPI libraries. The scope includes point-to-point communication (two-sided and one-sided) and collective operations.

The DICE sites involved in the project are: ASC cluster (with InfiniBand SDR and Quadrics) and AVETEC cluster (InfiniBand DDR and 10GigE). During the period of this project, ASC cluster was reconfigured to work with these two interconnects. For InfiniBand-SDR, MVAPICH library was used. For Quadrics, the proprietary library from Quadrics was used.

For the AVETEC cluster, we will run experiments there in the near future as a few setup issues are resolved. In this interim report we present numbers only from the ASC cluster and will update this report with the results from the AVETEC cluster as they become available.

4 Project Goals

The focus of this project has been to study the networking aspects (such as data movement and impact of networking protocols) in HPC clusters and how modern networking technologies deliver different performance to the applications using the MPI standard. The benchmarks designed as a part of the project uses the standard MPI-1 and MPI-2 semantics. It aims to explore the impact of networking technologies and to carry out comparative performance evaluation of different networking technologies for clusters.

5 Benchmarks and Evaluations

As part of our project and evaluation, we design and develop a variety of benchmarks designed to target individual system components as well as the overall network. Some of these tests have been released as the OSU Benchmarks at the PECNIT website. Other benchmarks will be released at this address in the near future:

<http://nowlab.cse.ohio-state.edu/projects/dice/>

The first class of benchmarks is designed for MPI-1.2 compliant libraries. Because full MPI-2 features are not available for all interconnects, it is necessary to make this subset as complete as possible. Additionally, most current applications are written to MPI-1.2, making this performance a high priority.

- *Ping-Pong Latency* (osu_latency): In this benchmark, the sending process issues a message of a selected size to the receiver, who responds with a message of the same size. This proceeds for many iterations and the round-trip time divided by two is reported.
- *Uni-Directional Bandwidth* (osu_bw): A window of 64 messages is sent from the sender to the receiver, which responds with an acknowledgement.
- *Bi-Directional Bandwidth* (osu_bibw): Each of the two communicating processes sends a window of 64 messages to the other. This process is repeated for many iterations and the computed bandwidth is reported.
- *n-Hop Latency* (osu_nhop): This benchmark determines the latency cost for each peer in the system. While traditional latency benchmarks measure only the latency of a selected neighbor, leading to the lowest-possible latency result, this benchmark calculates the latency for each peer. This takes into account the realization that many clusters are built from multiple layers of switches and determines the cost for communication to peers other than the nearest neighbor.
- *Displaced Ring Communication* (osu_drc): In designing this benchmark, we wish to evaluate the effects of contention and hot-spots across the network by using concurrent operations. To perform this evaluation, each process sends data to its "right" neighbor and receives from the "left" neighbor. During the first iteration of the benchmark, each neighbor is a distance of one rank away. Progressive iterations are 'iteration number' distance number of ranks away. In this manner there is one communication ring in iteration 1, two communication rings in iteration 2, and 'n' communication rings in iteration 'n.' During this benchmark, all processes are sending and receiving, stressing the network and host.

The second class of benchmarks is designed to evaluate the features found in MPI-2 compliant libraries. These focus on evaluating the one-sided performance characteristics found in that specification. These are likely to be of increasing importance as next-generation parallel applications are written.

- *One-Sided Put Latency* (osu_put_latency): The origin process calls MPI_Put (ping) to directly place a message of certain data size in the receiver window. The receiver (target process) calls MPI_Win_wait to make sure the message has been received. Then the receiver initiates a MPI_Put (pong) of the same data size to the sender which is now waiting on a synchronization call. Several iterations of this test is carried out and the average put latency is reported.

- *One-Sided Get Latency* (`osu_get_latency`): The origin process calls `MPI_Get` (ping) to directly fetch a message of certain data size from the target process window to its local window. It waits on a synchronization call (`MPI_Win_complete`) for local completion. After the synchronization call the target and origin process are switched for the 'pong' message. Several iterations of this test are carried out and the average get latency is reported.
- *One-Sided Put Uni-Directional Bandwidth* (`osu_put_bw`): The put bandwidth benchmark is carried out by the origin process calling a fixed number of back-to-back put operations and then waiting on a synchronization call (`MPI_Win_complete`) for completion. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time and the number of bytes sent by the origin process.
- *One-Sided Get Uni-Directional Bandwidth* (`osu_get_bw`): The get bandwidth benchmark is carried out by origin process calling a fixed number of back-to-back get operations and then waiting on a synchronization call (`MPI_Win_complete`) for completion. This process is repeated for several iterations and the bandwidth is calculated based on the elapsed time and the number of bytes sent by the origin process.
- *One-Sided Put Bi-Directional Bandwidth* (`osu_put_bibw`): The put bi-directional bandwidth benchmark is similar to the bandwidth test, except that both processes send out a fixed number of back-to-back put messages and wait for the completion. This test measures the maximum sustainable aggregate bandwidth by two nodes.
- *One-Sided Get Bi-Directional Bandwidth* (`osu_get_bibw`): The get bi-directional bandwidth test is similar to that of the Put Bi-Directional Bandwidth benchmark. Instead of using `MPI_Put` operations, however, `MPI_Get` is used.

6 Evaluation Results

Evaluation Platforms:

Evaluation using the designed benchmarks was performed on two systems in the DICE environment at the Aeronautical Systems Center (ASC) Major Shared Resource Center (MSRC) at Wright Patterson Air Force Base, OH. Additional future evaluation will be performed on clusters at AVETEC.

The two evaluated platforms are:

- *ASC – InfiniBand SDR (ASC-IB-SDR)*: These nodes are connected with Mellanox InfiniBand Single Data Rate (SDR) interconnects. These run at a data rate of up to 8

Gb/sec. The software stack installed is OpenFabrics Enterprise Edition (OFED) 1.1. The MPI library evaluated was MVAPICH2 from The Ohio State University.

- *ASC – Quadrics (ASC-Quadrics)*: In this cluster all nodes are connected with the Quadrics Elan interconnect. The QsNetII software stack was installed and the MPI library used for evaluation was Quadrics MPI.

Two-Sided Evaluation:

We first compare the basic two-sided benchmarks on each of the systems to assess the basic performance that is often quoted for systems by vendors. The one-way latency reported by our benchmark shows that the ASC-Quadrics system is able to achieve 2.03usec 0 byte latency and 2.40usec for 4 bytes (Figure 1). The ASC-IB-SDR system gives a higher latency of 4.82usec for 0 bytes and 4.90usec for 4 bytes. This confirms earlier evaluations that showed very low latency for Quadrics platforms.

Next we evaluate the throughput characteristics of each of our platforms. The ASC-Quadrics system peaks at a uni-directional bandwidth of 900MB/sec and reaches the N/2 bandwidth of 450MB/sec between 1KB and 2KB (Figure 2). The ASC-IB-SDR system reaches only 768MB/sec and the N/2 bandwidth is also reached between 1KB and 2KB message sizes. For the bi-directional bandwidth benchmark the ASC-IB-SDR shows higher performance (Figure 3). Each platform attains a peak of 900MB/sec, limited by the PCI-X bus, however from the graph it can be observed that the ASC-IB-SDR configuration achieves higher bandwidth for the small to medium sized messages.

One-Sided Evaluation

Next we perform our evaluation of the one-sided features that were added in the MPI-2 specification. We evaluate the put, get and accumulate operations for both latency and throughput, as applicable.

We first evaluate the latency of the `MPI_Put`, `MPI_Get` and `MPI_Accumulate` operations. From Figure 4 we observe that `MPI_Put` performance for both platforms is higher than that of the two-sided calls. For ASC-IB-SDR the performance is 8.11 usec for 4 bytes and 29.85 usec for the ASC-Quadrics platform at the same message size. Only at 16KB message size does ASC-IB-SDR rise above the 4 byte message latency of ASC-Quadrics. The same trends from the put operations continue for `MPI_Get` latency (Figure 5). The 4 byte latency on the ASC-IB-SDR platform is 16.72 usec, compared to 32.53 usec for the ASC-Quadrics platform. Note that zero byte latency is significantly lower than 1 byte transfers for both platforms – these calls do not require data transfer, so this effectively is measuring the synchronization time required. `MPI_Accumulate` also shows a similar performance gap (Figure 6); ASC-IB-SDR shows a 4 byte latency of only 8.17 usec, compared to 40.61 usec for ASC-Quadrics. Also of interest is that at 4KB the latency for ASC-Quadrics jumps to 146 usec, over tripling the 2KB latency.

Next we evaluate the throughput for `MPI_Put` and `MPI_Get` operations. From the graphs, it is apparent that for put operations, the throughput of ASC-Quadrics is higher than that of ASC-IB-SDR (Figure 7). As a percentage of available bandwidth, however, ASC-IB-SDR is able to achieve the same maximum throughput as the two-sided bandwidth benchmark. ASC-Quadrics, although having a higher absolute number, gives a slightly lower value than the two-sided benchmark. The bi-directional throughput (Figure 8) for put operations is similar to that of unidirectional in that ASC-Quadrics is able to maintain higher performance for most message sizes. For 16KB messages and above both platforms attain a maximum of 900 MB/sec. For `MPI_Get` operations ASC-IB-SDR shows higher performance than ASC-Quadrics for all message sizes under 32KB (Figure 9). As expected from the network speeds, ASC-Quadrics has higher performance for message sizes of 32KB and above. For the bidirectional get throughput ASC-Quadrics shows higher performance (Figure 10), however, the difference between each of the platforms is minimal for most message sizes. The largest difference occurs at 1KB, with a gap of 100MB/sec between the two platforms.

Overall Network Evaluation

In this section we evaluate our two benchmarks that are designed to encompass the entire network, rather than our previous results, which measured the network performance when a single pair of processes is communicating.

In the first benchmark, we evaluate the latency difference between different nodes in the network. Large switches are made up of multiple switch blocks, so this benchmark tests the additional latency cost for crossing switch blocks. On larger clusters, multiple switches are also deployed, so this benchmark will show the total difference across all nodes. Since the clusters being tested here are smaller, the difference is muted. From the N-Hop graph (Figure 11) we observe that the different switch blocks are clearly visible. Quadrics shows little difference in this evaluation.

Second, we evaluate the effect of contention in the network on bandwidth using the Displaced Ring Communication pattern described earlier. The results of this evaluation shown in Figure 12 show little difference in throughput for 1MB message sizes. The network in both cases is able to sustain the maximum throughput of the host.

7 Analysis of Deviations from Predictions

The results of our evaluation have been mostly according to our expectations, however, some interesting data points were observed through the benchmarks we have designed:

- *One-sided performance of the Quadrics platform was in most cases significantly worse than that of two-sided.*

Without knowledge of the Quadrics implementation, it appears that the implementation used to implement one-sided operations uses a thread-based event approach. This would explain the large latency values for even small messages. The

large jump in MPI_Accumulate latency at 4KB for Quadrics is still rather puzzling. We plan to further investigate these issues.

- *Displaced Ring Communication (DRC) showed no congestion issues associated with communicating with more “distant” peers on the fabric.*

When running this benchmark on other machines, we have observed significant congestion on InfiniBand interconnects. Our results on the two platforms evaluated in this report show little difference, however. Since the PCI-X bus is limiting the bandwidth to 900 MB/sec and the DRC benchmark has a bi-directional communication pattern, even if the network fabric is running at 60% efficiency, it would not be detectable on these systems. This shows a need to evaluate on PCI-Express systems in the future – as is planned.

- *N-Hop Latency was not as clear as expected for ASC-IB-SDR.*

In this case, it seems that the ports for the ASC-IB-SDR platform were not connected sequentially as is customary. If host 0 connects to port 0, host 1 to port 1, etc., the tiers should be clearly visible. Instead, the ports seem to have been configured in another way. This shows a need to be careful with system wire-up as it may have a difference for applications that do near neighbor communication.

8 Conclusions

During Phase I of this project, we have designed, developed and evaluated a set of MPI-level benchmarks to evaluate MPI libraries and the underlying networking technologies for clusters. The benchmarks, targeting both one-sided and two-sided operations, are able to bring out strengths and weaknesses of the MPI libraries and underlying networking technologies. During the last year, these benchmarks have received wide acceptance in the HPC community. As indicated in the Executive summary, these benchmarks are already being used by OFED, InfiniBand and iWARP vendors and by Linux distributors.

Future plan includes expanding these benchmarks to reflect multi-core and multi-threaded characteristics of emerging computing platforms and also expanding the benchmark suite to include I/O-related benchmarks.

9 Proposed Next Steps

During Phase I, as indicated earlier, we have not been able to carry out experiments on the AVETEC cluster yet. Once this cluster is available, we will perform additional evaluations on this cluster and evaluate the benchmarks for InfiniBand DDR and 10GigE interconnects and update this report. Phase II (year 2), of this project will focus on expanding the benchmarks for multi-core and multi-threaded computing platforms. Phase III (year 3) will focus on I/O-related benchmarks.

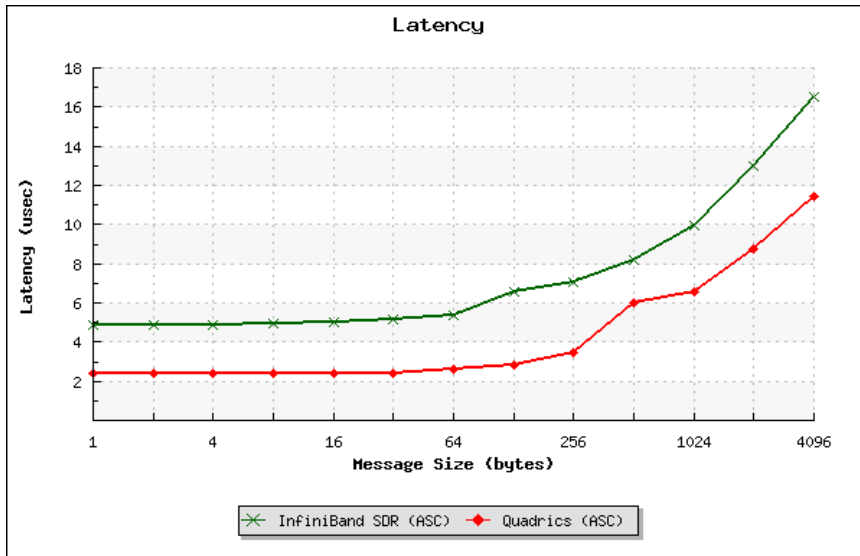


Figure 1. Ping-Pong latency comparison

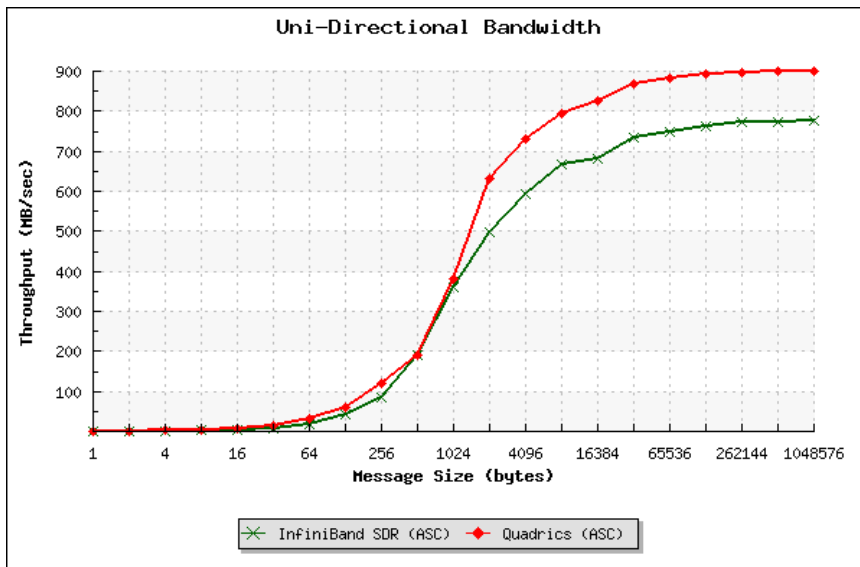


Figure 2. Uni-Directional Bandwidth Comparison

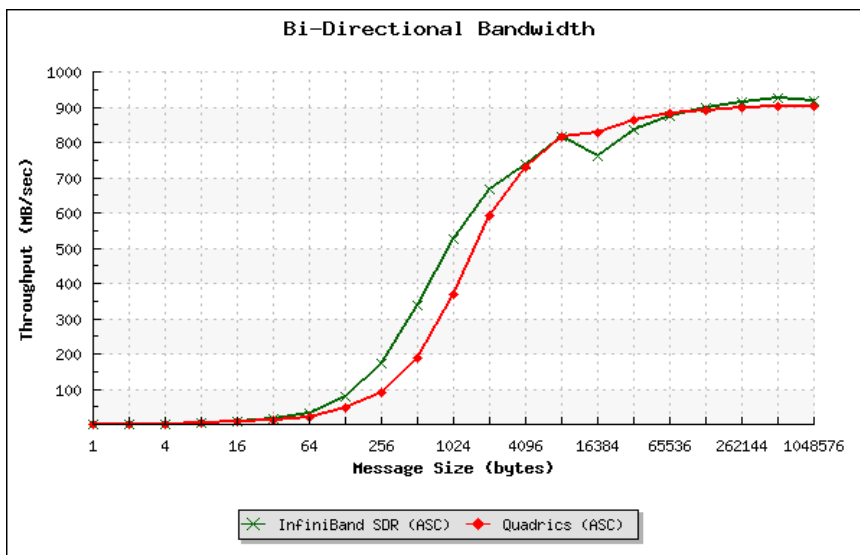


Figure 3. Bi-Directional Bandwidth comparison

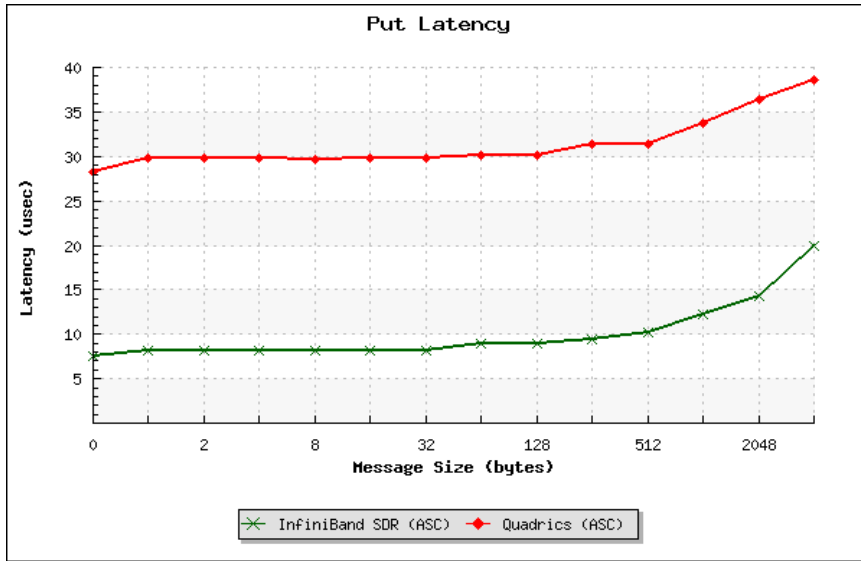


Figure 4. Put Latency comparison

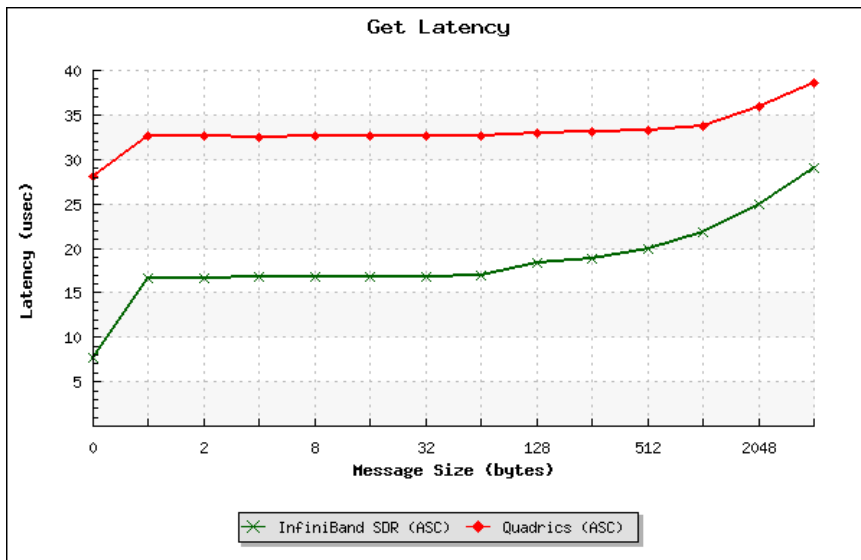


Figure 5. Get Latency comparison

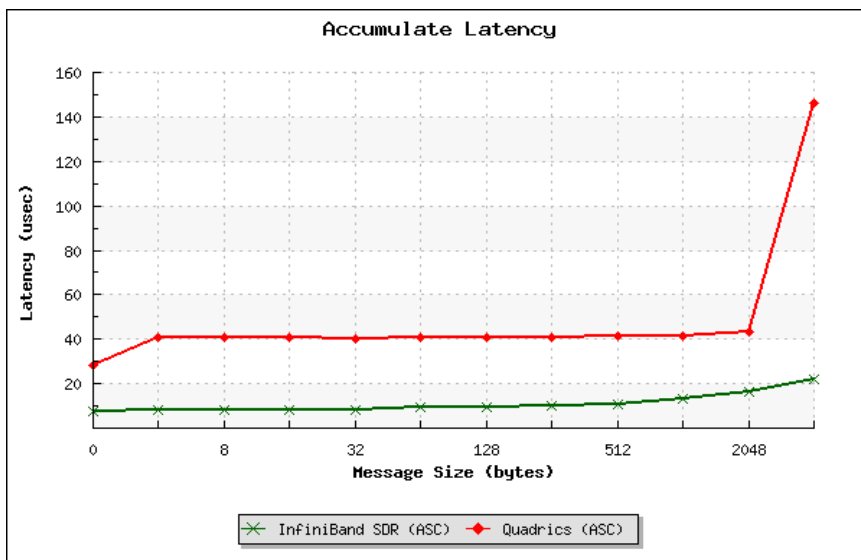


Figure 6. Accumulate Latency comparison

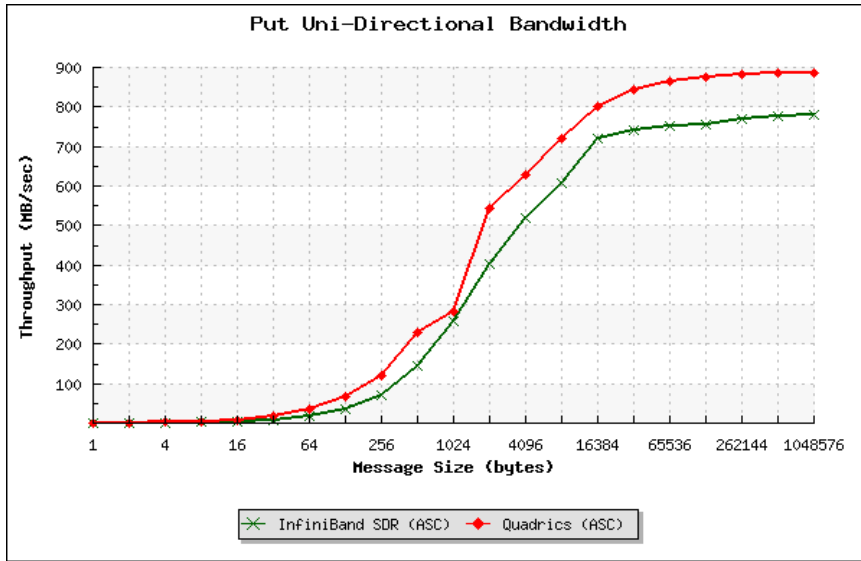


Figure 7. Put Uni-Directional Bandwidth comparison

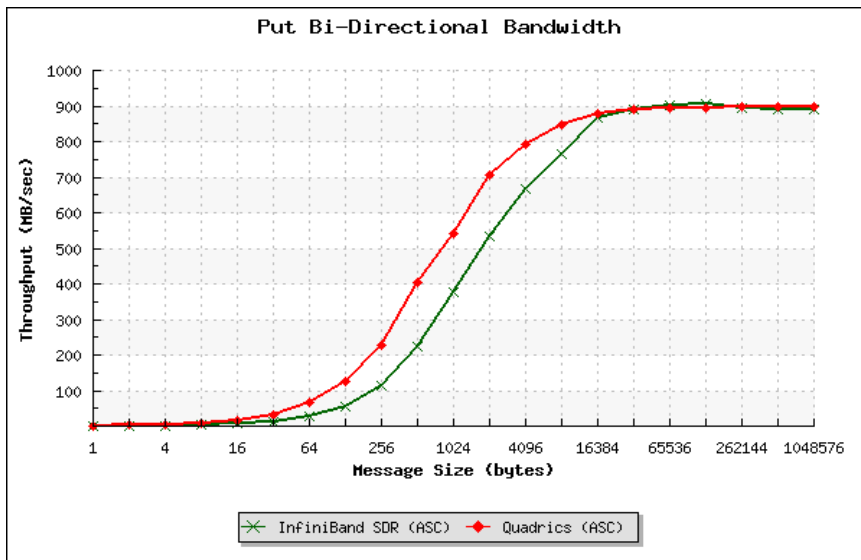


Figure 8. Put Bi-Directional Bandwidth comparison.

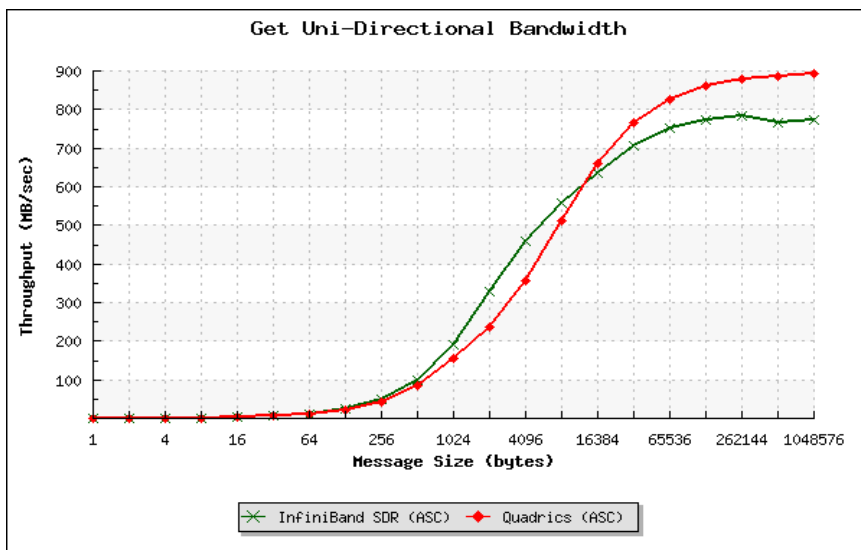


Figure 9. Get Uni-Directional Bandwidth comparison

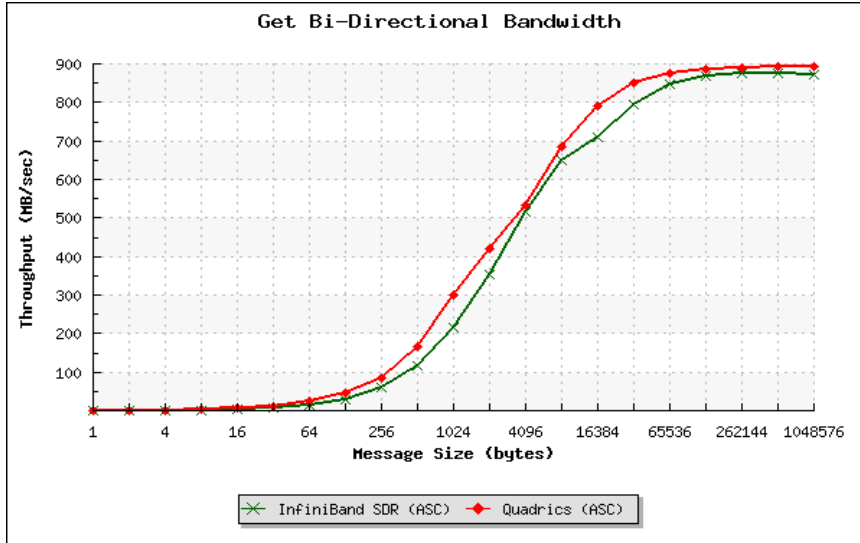


Figure 10. Get Bi-Directional Bandwidth comparison

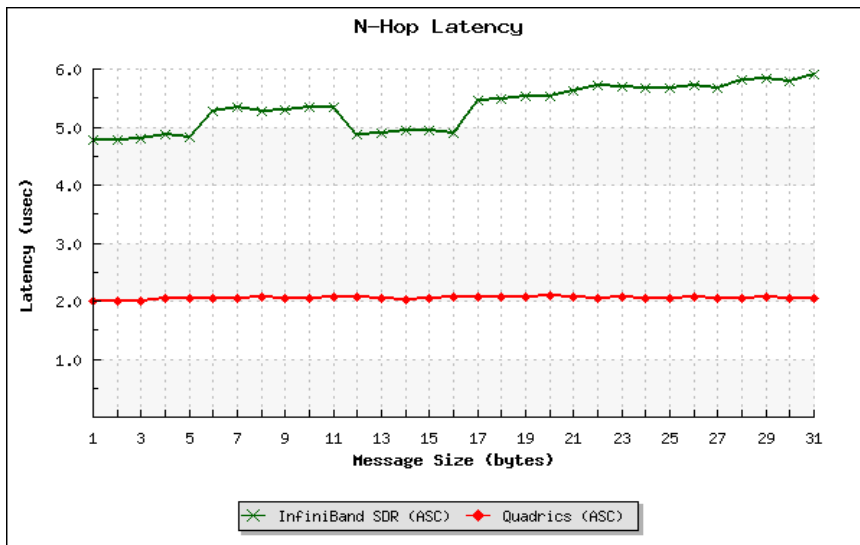


Figure 11. N-Hop Latency

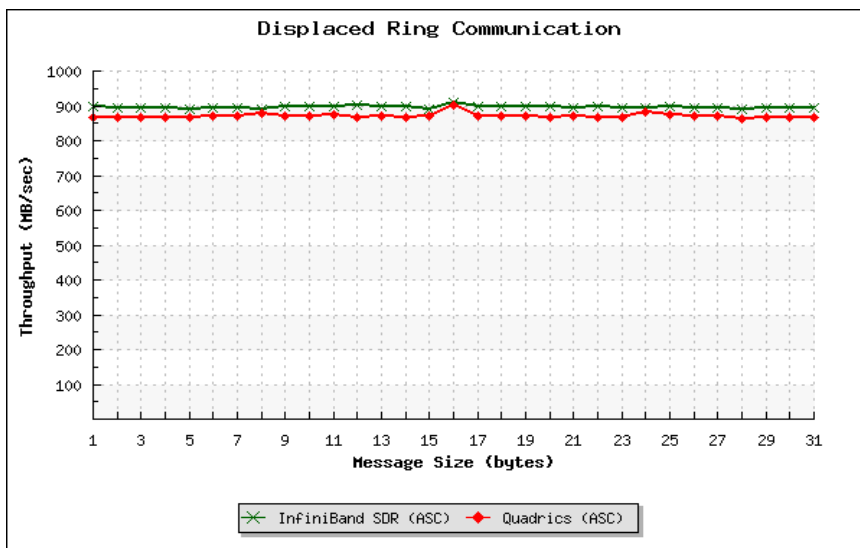


Figure 12. Displaced Ring Communication