Performance Characterization of Large and Long Fibre Channel Arbitrated Loops

Thomas M. Ruwart
University of Minnesota
Laboratory for Computational Science and Engineering

Abstract

The bandwidth performance of a Fibre Channel Arbitrated Loop (FCAL) is roughly defined to be 100 MegaBytes (10^6 bytes) per second. Furthermore, FCAL is capable of a theoretical peak of 40,000 I/O operations (transactions) per second., These performance levels, however, are largely not realized by the applications that use Fibre Channel as an interface to disk subsystems. The bandwidth and transaction performance of an Arbitrated Loop is sensitive to both the number of devices on the loop as well as the physical length of the loop. This study focuses on the effects of these two factors on the observed performance of Fibre Channel Arbitrated Loop as the number of nodes is scaled from 2 to 97 devices and as the physical length of the loop is scaled from 50 meters to several kilometers in length. To summarize, this study shows that the performance decreases significantly for very long loops and explains how this can be partially avoided. Also, the loop propagation delay on loops with many devices has only a moderate affect on performance. Finally, the effects of length tend to dominate the effects of population for very long, highly populated loops.

Motivation

The factors associated with the performance of long distance and highly populated networks and shorter-distance, lightly populated I/O channels have been well studied. Fibre Channel allows for the connection of a relatively large number of devices to a single I/O channel. At the same time it has many characteristics of a low latency, high bandwidth network. Typically when Fibre Channel is implemented in a storage environment, there are relatively few disk drives on a relatively short arbitrated loop. The propagation delay imposed by the physical length of the loop is normally insignificant compared to the latencies imposed by the disk drives themselves (i.e. rotational and seek latencies). Furthermore, the small number of disk drives that populate the loop do not contribute any significant propagation delay overall.

The Fibre Channel architecture makes it possible to extend an arbitrated loop well beyond the "typical" physical length and population scales. This can be done in order to accommodate, for example, direct access to physically remote disk drives or a completely populated loop of disk devices. Such a system exists at the University of Minnesota, where a 128-processor Origin 2000 computer system at one facility (the Minnesota Supercomputer Institute) is attached directly to a high-performance, high-capacity disk storage subsystem located at another facility (the Laboratory for Computational Science and Engineering) at a distance of approximately 3.8 kilometers. Since bandwidth performance is critical in this application, the effects of the extended distance of the loop needed to be considered. The infrastructure installed to support this system turned out to be an ideal test bed to construct a highly populated 30 kilometer loop. This, along with generous equipment loans from Seagate Technology, MTI, Finisar, Ciprico, and Ancor Communications, made it possible to investigate the effects of distance and loop population on the observed performance of a Fibre Channel Arbitrated Loop storage subsystem.

Overview

This study focuses on the overall performance of a single Fibre Channel Arbitrated Loop as the number of *nodes* increases and as the physical length of the loop increases. Disk drives are used as the nodes on the loop for this study since it is relatively simple to connect large numbers of them to a single loop and they are inexpensive compared to the alternatives (host computers). Furthermore, the performance of FCAL storage subsystems is also of great interest so this study can serve two purposes.

The performance metrics used are bandwidth and transaction rate or I/O Operations per Second (IOPS). Two aspects of performance are considered in this paper. First, there is the performance as seen by a benchmark application doing I/O to a single disk drive. Second, there is the aggregate performance of the disk subsystem as a whole. Finally, there is the issue of Access Fairness on the loop. Access Fairness is necessary in order to prevent lower priority devices from being starved for access to the loop. The effectiveness of the FCAL Access Fairness algorithm is demonstrated in this study by comparing the performance metrics across the individual disk drives for consistency. In other words, showing that all disk drives get reasonably similar performance when competing for loop access on a congested loop.

The testing process used to determine the performance levels involves the use of a benchmark program that attempts to saturate the disk subsystem with I/O requests. As the number of disks on the loop increases, the *aggregate* performance of the disk subsystem will increase to a limit imposed by the host bus adapter and/or the FCAL itself. This aggregate performance should remain at this limit as the number of disks on the loop reaches the maximum. At the same time, the performance as seen by each of the benchmark threads will start out at the peak performance capability of each disk and decline as contention for the loop increases due to the increasing number of disks arbitrating for access. The questions to be addressed are exactly what do these performance curves look like and how does the FCAL access fairness algorithm affect performance on a very congested loop. It will also be interesting to note how the number of non-participating disks present on the loop affects performance. In other words, if only 12 disks are being used on a loop of 96 disks, how the performance is different than having the same 12 disks on an isolated loop all by themselves.

Similarly, as the physical length of the loop increases, the signal propagation time increases proportionally. This translates to longer loop tenancies since each loop tenancy requires several round trips around the loop in order to perform the necessary arbitrate, open, transfer data, report status, and close operations. The effect of a physically longer loop is certainly lower performance but the question is how fast does the performance degrade as the size increases. Also, how does the performance further degrade as the number of devices is increased on a long loop. Finally, what is the difference between reading data from a disk subsystem as opposed to writing data to a disk subsystem under these various conditions.

It is worth noting that the intent is not to find the peak performance of the loop, the FC host bus adapter, the computer running the benchmark, or the disk drives. Rather, the intent is to find how the performance *changes* as the loop gets larger, longer, and both. Furthermore, it is the performance as seen by the application, in this case the benchmark program, that is important both from an individual benchmark thread perspective and as an aggregate whole over all the threads.

Fibre Channel Arbitrated Loop

Fibre Channel Arbitrated Loop (FCAL) is an ANSI standard that defines a ring topology of the Fibre Channel standard. FCAL can be used as an interface between a relatively large number of disk drives and a relatively small number of host computer systems. Up to 126 devices can be present on a single loop at any given time. These "devices" include host computer system FCAL adapters as well as peripherals (i.e. disks, tapes, printers, …etc.).

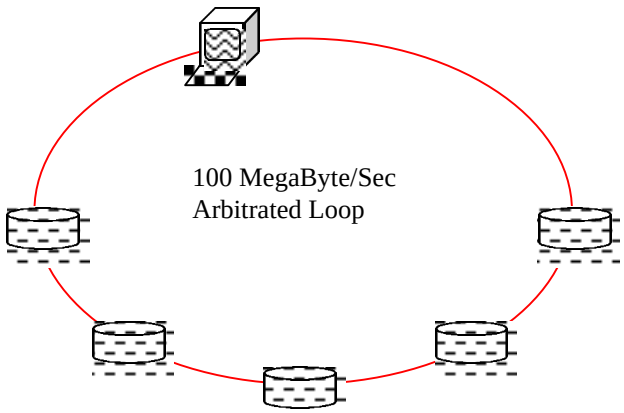


Figure 1. Example of an arbitrated loop configuration with one host computer and five disk drives.

The Fibre Channel standard provides for efficient support of many communication protocols such as TCP/IP, UDP, and SCSI. This study is restricted to the SCSI protocol using a single host computer and a number of disk drives on a Fibre Channel Arbitrated Loop. Furthermore, there is only one host computer system on the loop thus no host-to-host communication. Therefore, no other communication protocol is needed.

The SCSI Fibre Channel Protocol (SCSI-FCP) for read and write operations consists of three to four basic "phases":

- Command
- Transfer Ready (Write operations only)
- Data transfer
- Status

Each phase occupies one or more loop "tenancies," each of which is preceded by an arbitration period. A loop tenancy is the time between winning arbitration and the subsequent loop close operation. During a loop tenancy the device that wins arbitration "owns" the loop in order to communicate with another device (e.g. the host bus adapter sending a SCSI command packet to a disk drive) and all other devices remain quiet until the loop ownership is relinquished and a new arbitration cycle begins.

The process of a single exchange of data between two devices on the loop requires several "trips" around the loop. For example, during loop arbitration, the arbitrating device (call it A) sends out an arbitration request packet that must circumnavigate the entire loop before winning the arbitration. Winning arbitration marks the beginning of the loop tenancy at which time device A issues an Open request to a target device (call it B) with which intends to communicate. This open request requires a single trip around the entire loop. After the open succeeds, device A sends one or more data packets

to device B. After the data transfer is complete, device A issues a Close operation to device B and device B issues a close operation to device A which also effectively requires a single trip time around the loop. At this time device A relinquishes control of the loop and the next arbitration cycle begins. This marks the end of the loop tenancy. For this simple transaction, several trips around the loop are required. It is therefore easy to see that as the loop gets physically longer, each trip around the loop takes longer (since the speed of light has not yet been changed by ANSI) which adds to the overhead of each loop tenancy.

Other sources of additional overhead are the non-participating devices themselves. Even though these devices do nothing more than receive and re-transmit data on the loop as it passes by, there is a propagation delay *through* each device that, for a large number of devices, can become significant. Again, the questions to be addressed are (1) how significant is the cumulative propagation delay through a large number of devices on the loop and (2) how does this delay manifest itself in the performance of the loop and individual devices on the loop.

Finally, the question of access fairness on a heavily populated loop is of principle concern. The FCAL access algorithm is designed such that priority is assigned on the basis of loop ID (the device number from 0 to 125). The higher the loop ID, the higher the priority. Thus, when two or more devices arbitrate at the same time, the device with the highest loop ID will win arbitration. Since there is a potential for high priority devices to starve lower priority, a fairness mechanism is implemented to insure that all devices on a loop can eventually gain access within a reasonable timeframe. The way this works is that a higher priority device "agrees" not to re-arbitrate for the loop until after all other lower priority devices have at least had the chance to win arbitration for the loop. Devices can choose to ignore the fairness mechanism and be unfair (such as a fabric loop port) but for the purposes of this study, all devices on the loop are fair devices.

The Performance Benchmark Program – *xdd*

The performance benchmark program used to generate the performance data is a program called *xdd*. This program has been under development at the University of Minnesota for the past several years and is designed to provide detailed, accurate, and reproducible results of the I/O performance of disk subsystems. The design of *xdd* and the many run-time options allow for tight control of specific I/O parameters used in testing. These include read/write operation, device lists, request size, access patterns, thread synchronization, time stamping and tracing, and run time. *Xdd* was chosen for this study because with this level of control, it is possible

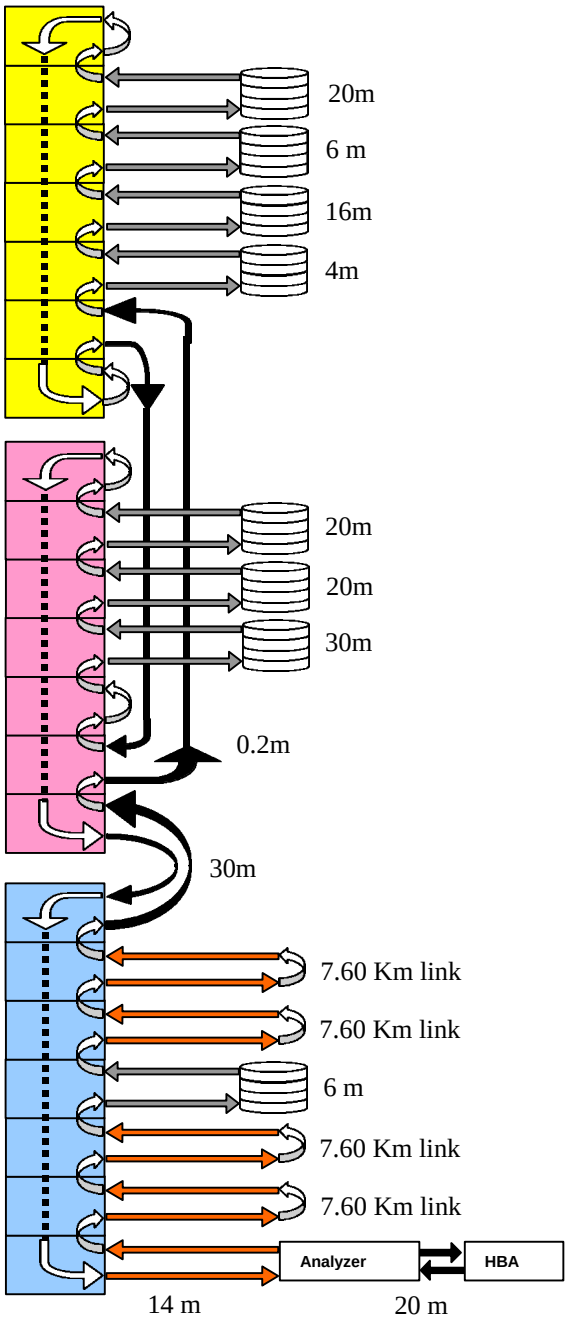


Figure 2. The Long Loop configuration consisting of three 7-port hubs, an analyzer, an HBA, and eight sets of 12 disks. Distances are for round-trips.

to change a single I/O parameter, either as an option to the program or an external parameter such as adding a disk, and observe the resulting effect.

Xdd reports results in terms of millions (10^6) of bytes transferred per second as well as the number of I/O operations per second. These are reported on a per-device basis and as an aggregate whole over all devices tested during a run. The detailed trace capability time stamps

every I/O operation and records this information in an ASCII readable file along with summary information for each device. All other run time options are reported in order to more accurately identify the parameters with which a particular run was performed.

The basic testing procedure was to run *xdd* on a set of disks with a given set of parameters. Each time *xdd* is run it runs the same test three times (three "passes") in order to smooth out any variations in the results. When *xdd* is started, it is given, among other parameters, a list of devices to test. *Xdd* spawns one thread for each of the devices under test. The threads perform the necessary initialization tasks (allocating buffers, locking memory, opening the target device, …etc) and proceed to a starting point. The threads wait at the starting point until all the threads have completed their initialization and reached the starting point as well at which time they are all released. Each thread performs I/O to its target device until the requested number of operations has completed at which time each thread collects and calculates its results and returns to the starting point to await the beginning of the next pass. Passes are synchronized at the start of each pass but there is no synchronization between threads during each pass. Three passes are run for each test after which *xdd* reports the "combined" average of all the passes of all the threads. Per-thread pass results and averages are also reported as the test proceeds.

The Configuration

A sizeable system was required to perform the benchmarking, based simply on the fact that 96 disk drives needed to be kept as busy as possible for extended periods of time driving the loop as close to saturation as possible. This required multiple fast processors, a great deal of main memory, a fast and efficient host bus adapter, and a large number of disk drives among other necessary items. The basic configuration consisted of the following components:

- A single Silicon Graphics ONYX2 (Origin 2000 class) computer system with the following configuration:
  - 8 - 195MHz R10000 processors (four nodes)
  - 2 GB main memory
  - 1 SGI XIO Fibre Channel Host Bus Adapter (Adaptec Emerald based)
  - IRIX 6.5 OS
- 96 Seagate Barracuda 9 Half Height Fibre Channel disk drives (ST19171WC running FB39 firmware)
- 8 MTI 2700 12-bay Fibre Channel disk enclosures
- 3 Vixel Rapport 1000 Fibre Channel hubs
- Finisar DB-9 copper and long wave optical GBICs (transceiver modules)
- AMP DB-9/DB-9 Fibre Channel cables
- Finisar GLA3100 Fibre Channel Analyzer
- Methode DB-9/NOFC Optical Media Interface Adapters

A single large system was chosen in order to provide for a single reference clock that is used to time stamp each and every I/O operation. It was also critical to have a large main memory on the machine to accommodate all the I/O buffers for the benchmark programs. Consider that when testing 96 devices, there are 96 separate benchmark threads, each with its own I/O buffer. If the each I/O buffer is 4 MegaBytes, a total of 384 MegaBytes of main memory is required. In addition to the I/O buffers are the time stamping trace buffers which can likewise be relatively large (on the order of 1MB each). All the testing was performed on this single SGI computer using the single host bus adapter. The I/O capacity of the Origin 2000 computer is well beyond the 100 MB/sec required for this test. Furthermore, the focus is on the relative performance of multiple benchmark runs rather than the performance of any single benchmark run.

The disk drives are all identical in model (Seagate ST19171FC) and firmware level (FB39). The disk enclosures provided by MTI are 12-bay Fibre Channel enclosures (model 2700) using DB-9 style copper Fibre Channel connections.

The Fibre Channel physical standard (FC-PH0) accommodates different physical media for the actual transmission of data. There are two basic types – copper and fiber-optic cables. The copper cables are designed to operate within the confines of a computer room and are limited to 25 meters between nodes. The optical fiber however comes in two varieties: multi-mode and single-mode. The multi-mode fiber using short wavelength lasers is designed to span distances of up to 500 meters between any two nodes. The single-mode fiber using long wavelength lasers is designed to operate over distances of up to 10 kilometers between nodes. The copper cables used here are unequalized DB-9-to-DB-9 AMP cables of various lengths. The optical cables are all 62.5 micron multi-mode cables of various lengths. Media Interface Adapters (MIA's) are used on the MTI enclosures to connect to the optical cables from the hubs. Finally, the Vixel Rapport 1000 hubs used the Finisar DB-9 copper and long wave optical GigaBit Interface Converters (GBICs) as well as IBM short-wave optical GBICs. The Rapport 1000 hubs are "passive" hubs and therefore require the use of the optical components in order to extend the loop to eight disk enclosures.

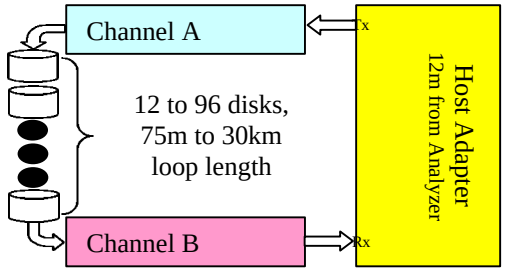


Figure 3. Finisar GLA-3100 Analyzer logical configuration.

The actual configuration used for both the long and large loop testing is shown in Figure 2. The configuration used a 10 meter copper cable from the HBA to the Finisar GLA3100 Fibre Channel Analyzer which in turn connected to the first of three Vixel hubs using a 7m optical cable. Events *from* the host bus adapter are recorded on Channel A of the analyzer. The signal from the host bus adapter is then routed to a second cable that attaches the analyzer to the first of three Vixel Rapport 1000 Fibre Channel hubs that connects to the disks and the extended single-mode fibers. Events entering the host bus adapter are recorded on Channel B. The time for a signal to travel from Channel A to Channel B depends on the cumulative length of the cable and the number of devices on the loop. The time for a signal to travel from Channel B to Channel A depends simply the length of the cable attaching the analyzer to the host bus adapter and the propagation delay of the adapter. In this configuration, the host adapter-to-analyzer cable length is 10 meters. Therefore, the transmission time between Channel A or B and the host bus adapter is approximately 40 nanoseconds.

The single-mode fiber occupied four ports on the first hub and were inserted into the loop on an as needed basis depending on the desired loop length. A time-domain analyzer was used to measure the actual length of the fiber path from the LCSE to the end-point at the Minnesota Supercomputer Institute (MSI). The single-mode fiber ran from the hub through three other communication facilities around campus before it eventually reached the communication closet at MSI at a measured distance of 3.77 kilometers. The fiber was then looped back using a simple SC-style optical cable junction guide.

The first enclosure of 12 disk drives is plugged into the first hub between the long fibers as shown in Figure 2. The second hub contains 36 disks (3 enclosures) and the third hub attach to the remaining 48 disks (4 enclosures). Testing the large loop consists of configurations of 12, 24, 48, and 96 disks. Using the hubs in this configuration it is easy to reconfigure the number of disks by simply plugging and unplugging hubs.

Time Scales

The time scales on which events occur provides a useful frame of reference when putting events into perspective. Three events of interest are Loop Trip Time, Node Delay (particularly for Large loops), and data transfer times (see Figure 4). The loop trip time is simply the time for a signal to traverse the entire length of the loop with a minimum number of devices. The loop trip time ranges from approximately 30 nanoseconds on a 6-meter loop to 150 microseconds on a 30 kilometer loop. Similarly, the data transfer time for 1024 bytes of data ranges from 10 to 160 microseconds as the loop increases in length from 50 meters to 30 kilometers. A 4-megabyte transfer can take from 21 to 186 milliseconds depending on loop length. However, the importance of Figure 4 is to demonstrate the *relative* effects of these different configuration variables.

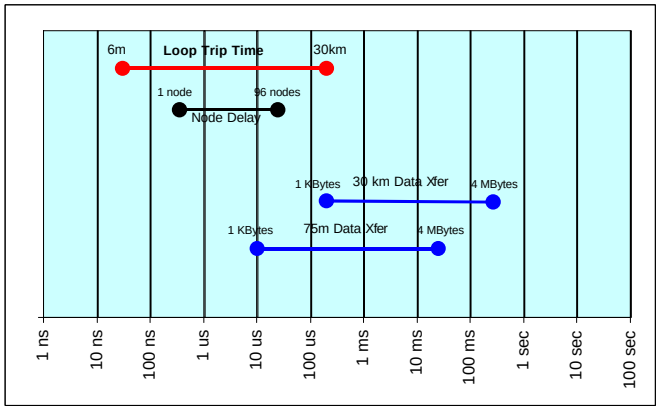


Figure 4. Time scales of events on a Fibre Channel Arbitrated Loop.

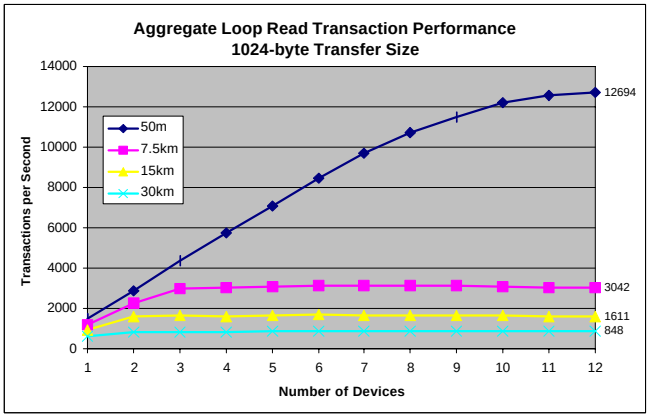
Baseline Performance

The Seagate Barracuda 9 Fibre Channel disk drive used in this study can sustain a bandwidth of 11.2 MB/sec on the outer cylinders and has the ability to handle nearly 1500 I/O operations per second by utilizing the read-ahead and write-behind caching on purely sequential transfers. The xdd benchmark threads do not queue I/O operations on the disks via command tag queuing and therefore do not realize the true peak bandwidth of these disks. For the purposes of this study, only the outer cylinders are used (within the first 2000 MB) in order to sustain a peak bandwidth of 11.2 MB/sec as well as the 1500 transactions per second.

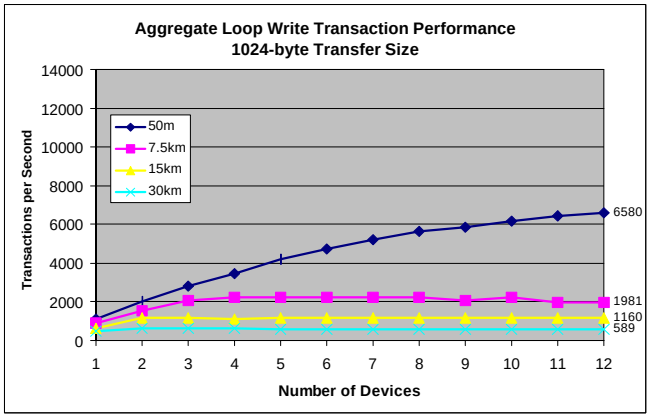
Long Loops

It is well within the scope of the Fiber Channel architecture to construct a physically very long loop. In theory, a loop could be constructed with 10 km of single-mode fiber between each of the 126 devices on a loop.

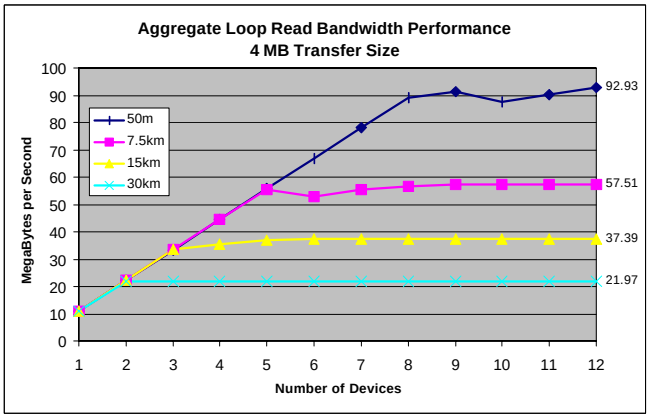
This loop would measure physically 1260 km or more than 750 miles. At this length, a single trip around the loop would be on the order of 6.3 milliseconds assuming the signal propagation through the fiber optic cable is 60% the speed of light or 5 nanoseconds per meter. A single 1024-byte SCSI read operation to a disk drive, assuming no disk latencies, would take 56.7 milliseconds (18.9 ms for each of three phases). This translates to a transaction rate of approximately 17.6 I/O operations per second which is significantly lower than the nearly 1500 I/O operations per second the drive is capable of. From simple estimates it is apparent that as the loop gets longer, the performance decreases due to speed-of-light limitations.



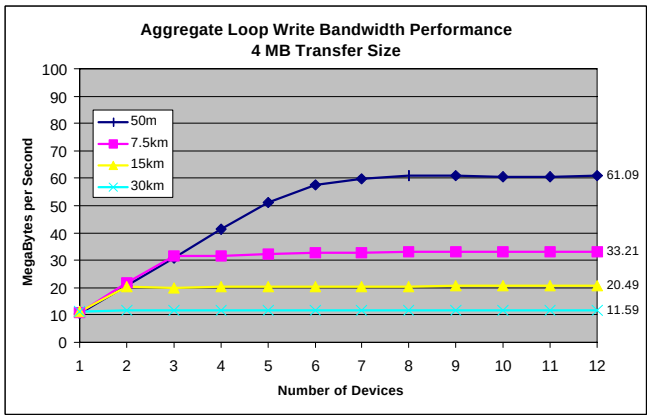
Graph 1. Aggregate loop performance for 1Kbyte simultaneous read transfers from 1 to 12 disks.



Graph 2. Aggregate loop performance for 1Kbyte simultaneous write transfers from 1 to 12 disks.



Graph 3. Aggregate loop performance for 4Mbyte simultaneous read transfers from 1 to 12 disks.



Graph 4. Aggregate loop performance for 4Mbyte simultaneous write transfers from 1 to 12 disks.

Therefore, an arbitrated loop gets "longer" as the cumulative length of the physical cabling increases. The length of the long arbitrated loop used for this study is approximately 30 kilometers. This was achieved by looping back four pairs of 9-micron single-mode fiber between the LCSE and the Minnesota Supercomputer Institute. Testing of this configuration placed a single box of 12 disk drives equidistant from the host bus adapter at approximately 15.5 kilometers. Using the four pairs of fiber it was possible to construct loops with lengths of approximately 7.5 km, 15km, and 30 km.

A set of xdd benchmarks were run on configurations from 1 to 12 disks that spanned the request size range from 1K to 4096K bytes per request. These tests were repeated on loops varying in size from approximately 50 meters, 7.5km, 15km, and 30km. There are several effects on performance that are described using the following graphs. First, the performance as a measure of the total aggregate transaction rate and bandwidth achievable on the loop as a function of the number of active devices on the loop. This is different for read operations (transferring data from the disk to the host computer) and write operations (transferring data from the host to the disk) (see Graphs 1-4). It should be noted that all read operations are effectively *from* the disk drive cache and all write operations are *to* the disk drive cache. Therefore, there are few delays due to rotational and/or seek latencies.

From these graphs it is plain to see that the length of the loop has a significant effect on the aggregate

performance of the loop. The performance deltas are different depending on type of I/O being performed. Read Transaction I/O, measured as the number of 1024-byte read operations per second, drops from 12700 to 3000 IOPS (76%) as the loop length increases from 50 meters to 7500 meters. At the same time, the Read Bandwidth performance only drops from 93 MB/sec to 58 MB/sec (38%). In each case though, the aggregate performance of the loop declines significantly from 50 meters to 30 kilometers.

Another interesting effect of length on the loop is the Read versus Write performance. The write performance is roughly half the corresponding read performance for both transaction and bandwidth (Graphs 1 & 3 compared to graphs 2 & 4). There are two reasons for this. First, the write operations performed in this test have four phases (Command, Transfer Ready, Data, and Status) as opposed to three for a read operation (Command, Data, and Status). The extra phase requires an extra loop tenancy that adds three loop trip times to the overall command processing time. For very long loops, three extra loop trip times become a significant source of additional overhead when compared to a three-phase read command.

The second reason write operations are slower than read operations on long loops has to do with the buffer-to-buffer credit (BB_CREDIT) management. For read operations, two buffers are used in the host bus adapter to receive incoming data from the disk drive. On write operations, the disk uses only one buffer to receive data from the host bus adapter. The data receiver sends the data source a Receiver Ready (R_RDY) signal for each buffer that it has available for receiving data. Upon receipt of an R_RDY, the data source will send a single frame with up to 2048 bytes of payload data. There is a one-to-one correspondence of R_RDYs to 2048-byte data frames. Hence, it takes an entire loop trip time (the time for a signal to travel around the entire loop) to complete an R_RDY/Data Transfer cycle. Since during read operations the host bus adapter sends two R_RDYs and receives two buffers at a time from the disk drive, it will get roughly twice as much data per unit time as a write operation where only one buffer is being used. This is normally not a problem for short loops because the loop trip time is significantly less than the data transfer time and the extra overhead induced by the single buffer is not noticed. However, when the loop gets very long, the loop trip time becomes a significant portion of the overall data transfer time.

For example, a 50 meter loop has a loop trip time of approximately 0.2 microseconds and a 30 kilometer loop has a loop trip time of roughly 150 microseconds. The time to transfer a single 2048-byte data frame at 1.0625 gigabits per second is approximately 20 microseconds. On the 50-meter loop, the loop trip time is 1% of the data transfer time and is not a significant source of overhead. On the 30-kilometer loop however, the loop trip time is 150 microseconds or 750% of the data transfer time. Analysis of the traces from the Finisar analyzer shows that on a 30-kilometer loop a read operation using two buffers (BB_CREDIT=2) takes an average of 180.4 microseconds for 4096 bytes (2 frames). A write operation, using one buffer (BB_CREDIT=1) takes an average of 171.2 microseconds for only 2048 bytes (1 frame) or half the amount of data during roughly the same amount of time as the read operation. Hence the observed performance disparity.

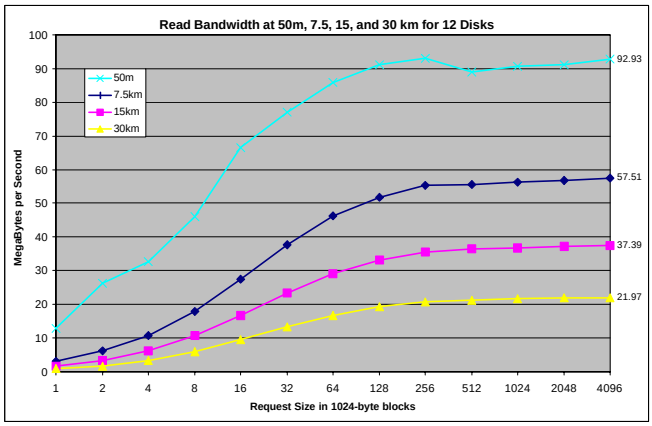
This demonstrates that the number of available buffers on the receiving device has a significant impact on the bandwidth performance of long loops. Therefore, in order to increase the bandwidth performance on long loops it is necessary to increase the number of receiving buffers. A simple estimate of the optimal number of buffers is:

$$2 + ((\text{Loop trip time}) / (\text{Data frame transfer time}))$$

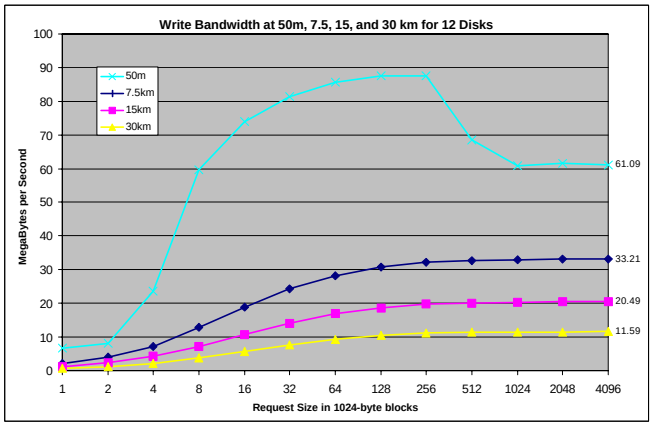
The loop trip time is the time for a signal to traverse the loop and the Data Frame transfer time is the time required to transfer a single data frame. In the current 1Gbit/sec Fibre Channel standard, a data frame is 2048 bytes and the corresponding Data Frame transfer time is approximately 20 microseconds. Thus, given a 30-kilometer loop with a loop trip time of 150 microseconds, the optimal number of buffers would be $2+(150/20) \approx 10$. This would insure that the sender would always receive at least one R_RDY before it has completed sending the 10^{th} buffer and therefore would be able to stream data out continuously. This assumes that the receiving device can absorb data (i.e. write it to the disk or cache) as fast as it can receive it from the sender. Given this number of data frame buffers, a 4 MB transfer on a 30 kilometer loop would operate at maximum speed.

If this example is extend to faster Fibre Channel speeds (2-4 Gbit/sec) the buffering will have a more significant impact. This is because the Data Frame transfer time decreases with higher Fibre Channel speeds but the travel time does not. On 2Gbit Fibre Channel, the number of buffers increases to $2+(150 / 10) \approx 17$ because now it takes only 10 microseconds to transfer a 2048-byte data frame but the travel time remains constant at 150 microseconds for a 30-kilometer loop.

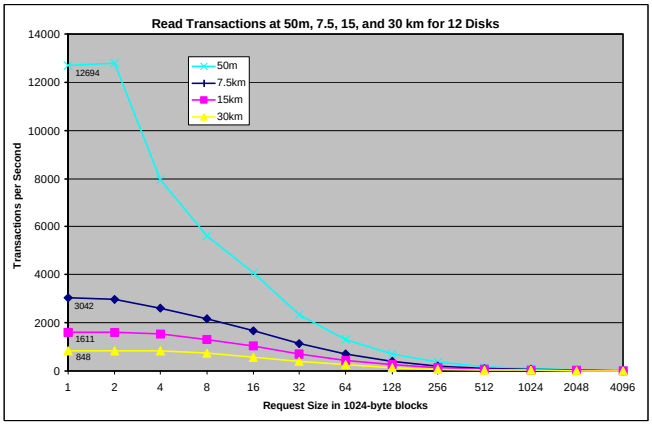
The aggregate performance of a loop of any length can also be viewed as a function of the data transfer size. Graphs 5-8 show the aggregate bandwidth and transaction performance curves of 12 disks being accessed simultaneously on a loop as a function of the request size used. The aggregate bandwidth increases to the loop maximum which is largely governed by the physical length of the loop. The longer the loop length, the lower the achievable peak bandwidth. An interesting effect for read and write operations is that the knee of the



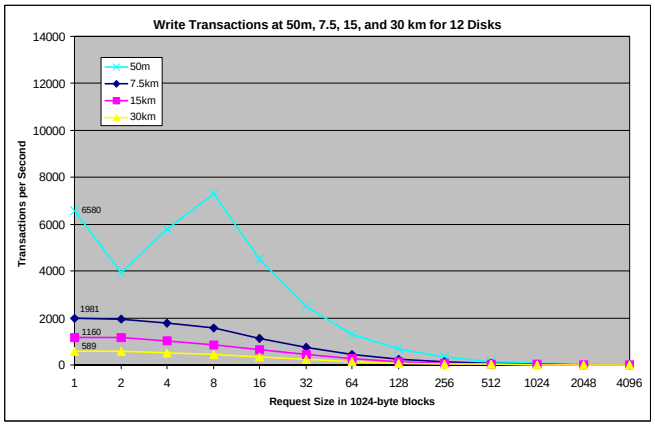
Graph 5. Aggregate loop read bandwidth as a function of Transfer Size using 12 disks.



Graph 6. Aggregate loop write bandwidth as a function of Transfer Size using 12 disks.



Graph 7. Aggregate loop read transaction performance as a function of Transfer Size using 12 disks.



Graph 8. Aggregate loop write transaction performance as a function of Transfer Size using 12 disks.

performance curves occur at the same request size for each loop length. In other words, 256Kbytes is an optimum request size for read and write operations independent of the loop length. The dip at the tail end of the write bandwidth curve in graph 6 is an artifact that is still being investigated.

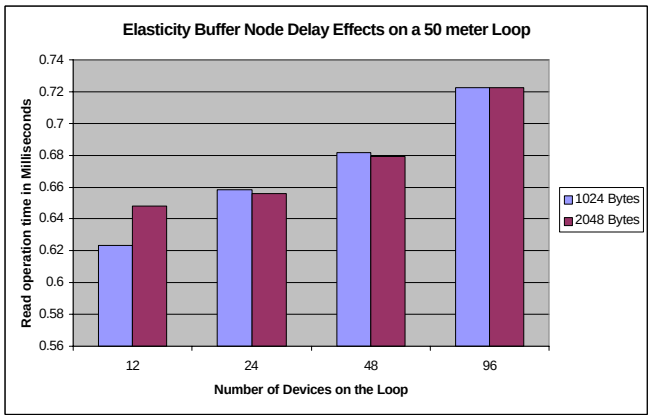
Large Loops

A loop gets "larger" as the number of devices physically connected to the loop increases. In this study, the number of disk devices is increased from 1 to 96 for a total of 97 nodes including the host bus adapter. Performance measurements are taken with 12, 24, 48, and 96 disks on the loop. In order to get very detailed information about events on the loop, analyzer traces are taken for many of the tests.

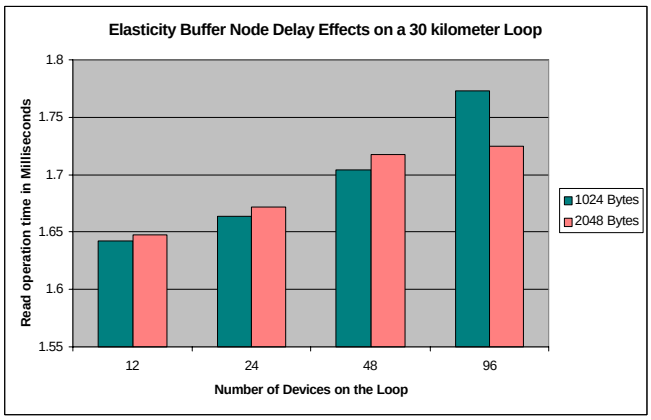
There are three points of interest covered in this part of the study. First, how the performance of a single disk affected by the presence of other non-participating devices on the loop. Second, what happens to the aggregate performance of the loop as the congestion increases. And finally, what happens to the performance of the individual benchmark threads as the congestion increases.

The overall effect of a highly populated loop depends heavily on the amount of data being transmitted. To test this effect a single disk is accessed using 128 read and write options of 1024-bytes, 2048-bytes, and 4 Megabytes. A set of access tests are run for loop populations of 12, 24, 48, and 96 disks. The additional disks are not active in the sense that they are not accessed but do have a presence on the loop. All I/O operations are time stamped and analyzer traces are taken for each access test.

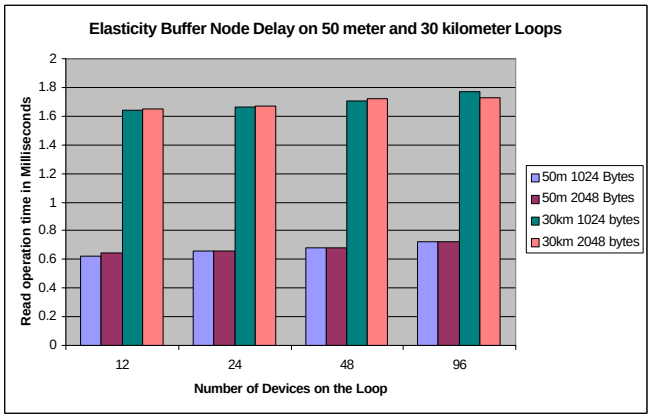
For short transfers the effect was measurable but still relatively small. For large transfers the effect was not significant. Graph 9 shows the shortest recorded time for each I/O operation of a single disk for read operation of 1024 bytes and 2048 bytes. The I/O time increases steadily as the number of non-active devices are added to the loop. This graph shows two trends. First is the most obvious trend that as devices are added, the I/O time increases. Secondly, as the data transfer size increases, the effect of additional devices becomes negligible.



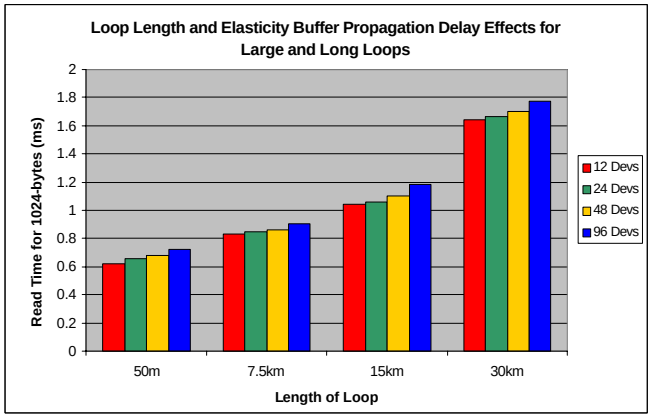
Graph 9. Effect of node propagation delay on read operations on a 50-meter loop.



Graph 10. Effect of node propagation delay on read operations on a 30-kilometer loop.



Graph 11. Comparison of the effect of node propagation delay on read operations on 50-meter and 30-kilometer loops.



Graph 12. Comparison of the effect of node propagation delay on read operations as a function of loop length.

The measured increase in command operation time amounts to 99 microseconds for 1024-byte read operations when an additional 84 devices are added to the loop. Each of the three phases in a read command (Command, Data, and Status) takes three loop tenancies to complete for a total of nine loop tenancies for a single read command. The increase per loop tenancy is approximately 19 microseconds (84 times the elasticity buffer delay per node of 226 nanoseconds). Therefore, the *expected* increase in the overall command time is 9 times 19 microseconds or 171 microseconds. The observed value of 99 microseconds is only about half that. So where did the remaining time go?

By using the analyzer traces it is possible to partially dissect each read operation into its three phases and further into each loop tenancy. Table 1 shows the time for each phase (Command, Data, and Status) as well as the Idle time between phases on a 50 meter loop populated with 12 and 96 devices. The Delta row is the amount of increase/decrease in time for each phase as the population changes from 12 to 96 devices. A single loop tenancy is measured at approximately 20 microseconds when all 96 disk drives are attached to the loop. This is consistent with the theoretical propagation delay that is calculated by multiplying the number of devices on the loop by 226 nanoseconds of propagation delay per node.

With the addition of 84 devices, the increase in each loop *tenancy* is approximately 19 microseconds. The Command phase (three loop tenancies) shows an increase of 52 microseconds which is consistent with the theoretical increase of 57 microseconds (19 times 3). There is no increase in the idle time between the Command and Data phases as would be expected. The Data phase only increases 34 microseconds instead of the expected 57. Upon closer inspection however, the missing time most likely was absorbed in the Idle time between the Data and Status phase which shows up as a decrease in time. The Status phase likewise only increases 34 microseconds and again, that additional time is most likely absorbed in the inter-command idle time. (Using a single 2-channel analyzer as described in Figure 2 it is difficult to capture events that occur on both sides of the sending and receiving devices. To accomplish this a 4-channel analyzer would be required.)

Table 1. Phase timing for 12 and 96 devices on a loop.

# Devs	Cmd µsec	idle µsec	Data µsec	idle µsec	Status µsec	Next Cmd µsec	Total Time µsec	IOPs
12	10	220	18	211	8	200	667	1499
96	62	220	52	177	42	200	753	1328
Delta	**52**	**0**	**34**	**-34**	**34**	**0**	**86**	**89%**

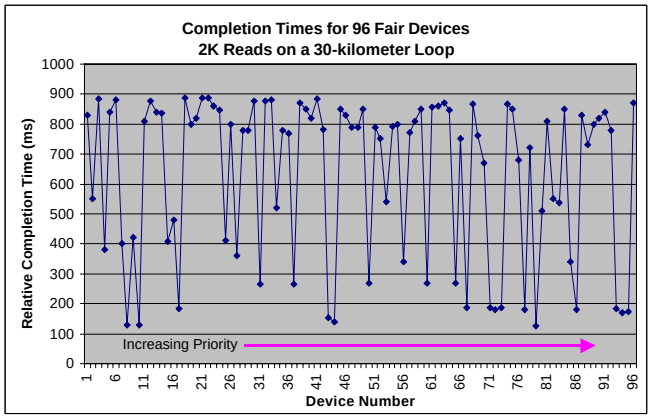
Therefore, the net effect of a loop populated with 96 devices is quantitatively about 11-13% for small (1024-byte) read requests. The effect decreases as the request size gets larger. For 4MB transfers, the effect was less than 0.1%.

Long Large Loops

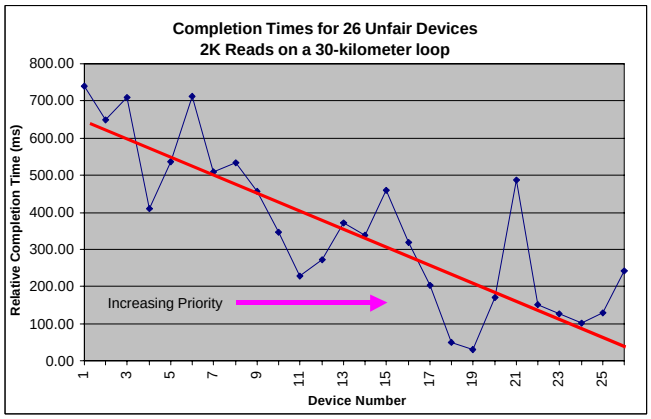
The testing of a single large long loop is accomplished by populating the long loop with the same 96 devices used in the large loop configuration. A loop of 30 kilometers is incrementally populated with disks and benchmarks are run similarly to the testing of the large loop. The loop trip time for long loops (7.5 kilometers and above) quickly became the dominant factor in the overall delay (See graphs 9-12). From the previous discussion on Large Loops, the increase in propagation delay due to large loop populations is on the order of 20 microseconds per loop tenancy. Similarly, from the discussion on the effects of long loop lengths, the increase in propagation delay is approximately 35 to 150 microseconds for loop lengths of 7.5 to 30 kilometers respectively. Together, the propagation delay for a single trip around the loop amounts to approximately 170 microseconds – a value that has been verified with the analyzer traces.

Access Fairness

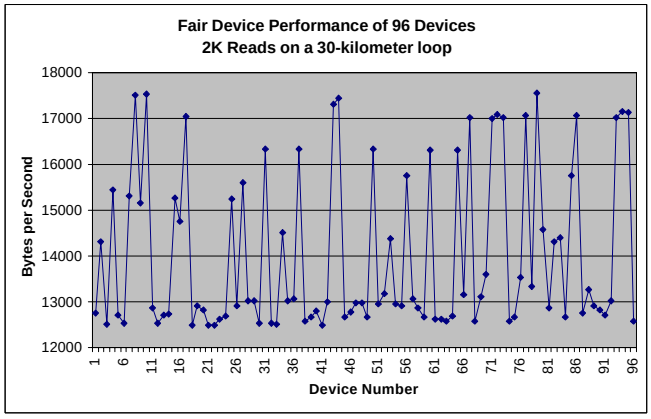
The access fairness worked as advertised and can be seen by the results from the tests that were performed simultaneously accessing 96 devices on the loop. Graph 13 plots the completion times of each of the 96 devices on a 30-kilometer loop in increasing priority. If the devices



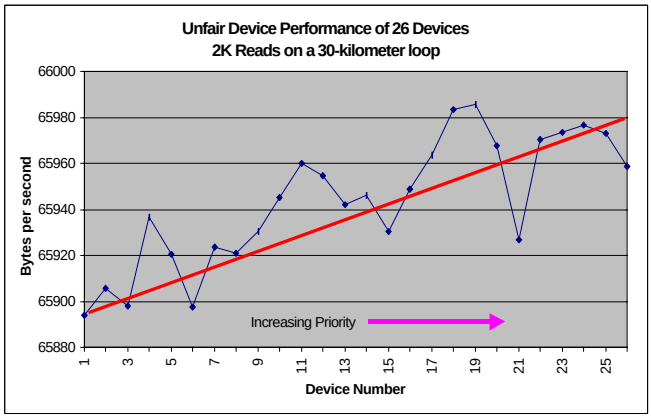
Graph 13. The effect of unfair device behavior on the order of completion for read operations across 26 devices on a 30-kilometer loop.



Graph 14. The effect of unfair device behavior on the order of completion for read operations across 26 devices on a 30-kilometer loop.



Graph 15. The effect of fair device behavior on the bandwidth performance of read operations across 26 devices on a 30-kilometer loop.



Graph 16. The effect of unfair device behavior on the bandwidth performance of read operations across 26 devices on a 30-kilometer loop.

were “unfair” the plot would show the higher priority devices completing before lower priority devices. Graph 14 shows the unfair behavior of 26 disks on a shorter 30-kilometer loop. The unfair devices are older Barracuda 9 disks that are running a very old version of firmware that did not have the fairness algorithm implemented. It is clear from Graph 14 that the higher priority devices get preferential access to the loop since they finish long before the lower priority (lower address) devices.

Conclusions

Long loops see a significant drop in aggregate performance, particularly for write operations on this configuration. This was primarily due to the number of buffers available on the disk drive to receive incoming data. The transaction performance is significantly affected for both reads and writes more than the bandwidth performance when viewed as a percentage of the peak. In short, extended loop lengths (greater than 5 kilometers) begin to show appreciable performance degradation.

As the number of devices on the loop increases, the propagation delay through the devices introduces a small but noticeable performance degradation. The degradation is more noticeable in transaction rate than in bandwidth.

As the loop congestion grows, the performance of each thread degrades uniformly such that each the average performance is the same over all threads. This is due in part to the loop access fairness mechanism. It should be noted that the access fairness algorithm does not guarantee equal performance among all devices on the loop rather it guarantees that each device will have an access window within which it can win arbitration for the loop and perform its function. The net effect, however, seems to indicate an even distribution of performance for all devices on a heavily congested loop.

The factor that contributes the most performance loss is the length of the loop. In terms of scale, the length of the loop can contribute 100 to 1000 times more propagation delay than the elasticity buffers in the devices themselves.

Future Work

Based on experiences from this project, researchers at the LCSE are working on ways to better gather and visualize performance data for large and complex storage area network configurations. This project alone generated nearly 10GB of compressed performance data much of which still needs to be analyzed. The LCSE is also currently working with Ancor Communications and Brocade Communication Systems evaluating their respective Fibre Channel switch products. This research is focused on what happens to the performance of large disk subsystems attached to these switches as the cross-sectional bandwidth and cross-sectional transaction rates are increased to the point of overwhelming the capability of the switch.

Acknowledgements

This work was performed at the University of Minnesota Laboratory for Computational Science and Engineering with support from the National Science Foundation and the Department of Energy under grants NSF/ACI 96-19019, DE/B347714, and NSF/CDA-9502979. This work was also partially supported by Minnesota Supercomputer Institute at the University of Minnesota. Other contributors and supporters include Seagate Technology, Inc., MTI, Vixel Corporation, AMP, Inc., Finisar Corporation, Ciprico, Inc. and Silicon Graphics, Inc.

References

Benner, Alan F. *Fibre Channel – Gigabit Communications and I/O for Computer Networks,* McGraw-Hill Series on Computers, 1966

Gary R. Stephens and Jan V. Dedek. *Fibre Channel – The Basics*, Ancot Corporation

Robert W. Kembel. *The Fibre Channel Consultant – Arbitrated Loop*, Connectivity Solutions