CITI Technical Report 92–2

Evaluating the Network System Corporation Data Exchange Unit Performance under AIX/370

Charles J. Antonelli
`cja@citi.umich.edu`

ABSTRACT

This report presents the results of tests run to evaluate the performance of Network Systems Corporation's Data Exchange Unit (DXU) channel-to-FDDI interface.

June 9, 1992

Center for Information Technology Integration
University of Michigan
519 West William Street
Ann Arbor, MI 48103-4943

Evaluating the Network System Corporation Data Exchange Unit Performance under AIX/370

Charles J. Antonelli

June 9, 1992

1. Introduction

We have been providing AIX/370-based AFS file service on IBM mainframes (first a 3090-600E and later an ES/9000-720) for many months as a result of our efforts on Phase I of the IFS Project at the Center for Information Technology Integration (CITI). From the beginning, it has been painfully obvious that the most severe performance bottleneck of this service has been the connection between the mainframe and the campus network. This bottleneck is due primarily to two culprits: a slow channel-to-network converter and a slow network.

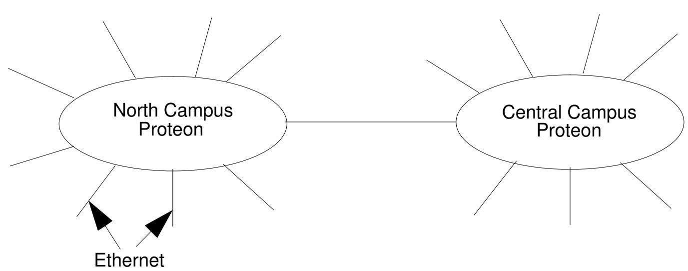

Figure 1. Campus Networking Infrastructure

Figure 1 shows a birds-eye view of the campus networking infrastructure. The backbone of this infrastructure consists of a pair of Proteon token rings operating at 80 Mb/s.[1] For efficient campus file service, it is essential that the connection between a mainframe-based file server and the campus backbone support the highest rate of data exchange possible. Unfortunately, that is currently not the case, as shown in Figure 2. The current connection consists of a 4.5 MB/s streaming FIPS (parallel) channel connected to a Bus-Tech Incorporated ELC2 running in native mode. The BTI ELC2 is connected to a portion of the Computing Center's production Ethernet, which is connected to the North Campus Proteon token ring. The two limiting factors in this arrangement are the BTI, which previous measurements have shown to support a maximum of 3 Mb/s [1], and the Ethernet with its peak rating of 10 Mb/s. What we need is a big, fat pipe from an ES/9000 channel to the campus backbone. FDDI, with its 100 Mb/s rated speed, could serve as such a pipe given the availability of suitable interfaces at each end. In 2Q91, we became aware of the Network Systems Cor-

1. In this paper, the abbreviation "b" is used to indicate bits; "B" means bytes.

poration (NSC) Data Exchange Unit as a suitable host-end interface. Accordingly, we obtained the use of an NSC DXU for evaluative purposes.

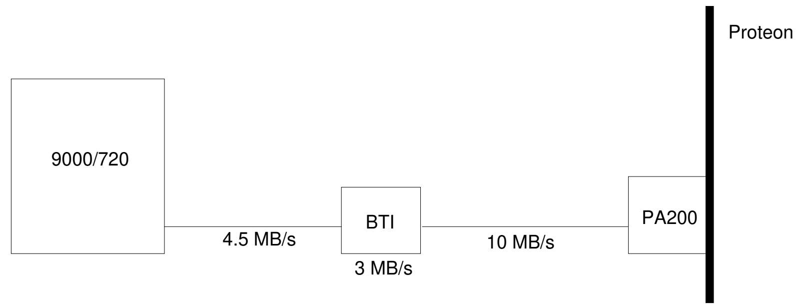

Figure 2. Current Topology

From an external point of view, one possible configuration of an NSC DXU connects multiple FIPS channels to an FDDI ring [2]. The internal architecture of the DXU consists of a high-speed bus, a shared memory and processor card (collectively called a \fInucleus\fP), two NB225 host interface cards for FIPS channels, one IP router card, and an NCDAS FDDI card.[2] Packets from the ES/9000 travel over the channel to an NB225, which writes the packets to the shared memory and notifies the IP router. The IP router then sends the packets on to the NCDAS and out onto the FDDI ring or to the other NB225 and to a second channel.

2. Test Procedure

Ideally, we would have liked to connect a couple of dozen AFS client machines via FDDI through the DXU to AIX/370, and start serving files *en masse*. We couldn't do that; we didn't have enough RISC System/6000 machines (the only platform available to us capable of being fitted with FDDI interface cards) or enough interface cards. We therefore set up a private FDDI ring with two RISC System/6000 machines (**maggie**, a model 520, and **esa**, a model 320) running AIX/V3 Release 3.0.3, each fitted with a pair of FDDI cards we borrowed from the IBM Kingston Programming Lab. The two NB225 cards were connected over 4.5 MB streaming FIPS channels to two AIX/370 guests (**bart** and **lisa**); a third AIX/370 guest (**krusty**) participated in one of the tests. All guests were running under VM/XA 2.1 on the B side of the ES/9000; we used a GA 1.1-based AIX/370 kernel built with Sig Handelman's TCP/IP improvements and a driver for the DXU obtained from AIX/370's Non-Integrated Function Library. We were constrained to run along with an MVS guest providing library database access that used around 20% of the B side processor complex. An IBM/RT workstation (**max**) running AOS 4 without FDDI capability participated in the baseline tests. In addition, we utilized a Proteon P4200 interface that connected our private FDDI ring to the north campus

2. A second DXU, obtained for testing by the Systems Group and configured with a nucleus, a NCDAS card and several Ethernet cards, did not participate in this evaluation.

Proteon ring. All machines were also able to communicate over the Computing Center development Ethernet. The complete experimental testbed is shown in Figure 3.

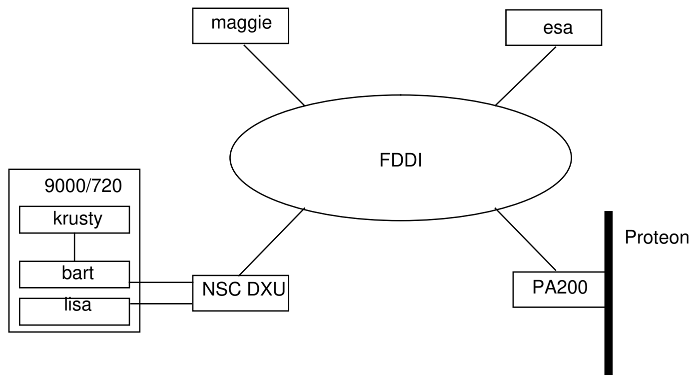

Figure 3. Experimental Testbed

Using this setup, we ran the following benchmarks: baseline performance, in which the performance of the IP software on AIX/370 was measured; TCP DXU performance, in which File Transfer Protocol (FTP) performance was evaluated; RX DXU performance, in which we measured the speed with which the Rx protocol (the AFS client/server communications protocol) could transfer bytes through the DXU; AFS DXU performance, in which we measured how well the testbed was able to serve files; and client load performance, in which we measured how well the testbed was able to serve files to multiple clients.

Instead of arranging for system time and running these tests on an otherwise idle system, we were interested in obtaining numbers more indicative of actual use; that is, we wanted estimates of performance while the system was doing other things besides serving files.

We also ran more TCP/IP-based tests, in particular tests using the Internet File Transfer Protocol (FTP), than in previous performance measurement sessions. For AFS file service, Rx performance (which is based on UDP/IP) is paramount. On the other hand, for comparing results to those obtained by others [3,4], it is important to obtain FTP performance figures.

2.1 Baseline Performance

These tests did not involve the DXU hardware at all; instead, we were interested in measuring the maximum throughput supported by the the various kernel IP layers. This was done by performing loopback tests, in which a machine would pump packets through its loopback interface and back to itself; these tests then measure the theoretical maximum throughput of the underlying TCP/IP software.[3] Three TCP/IP loopback tests were run on **lisa**, **maggie**, and **max**. All tests involved one party sending a 1 megabyte file[4] via FTP to the other, who threw it away to `/dev/null`. The tests

3. The interaction with the loopback device was not taken into consideration.

4. Actually, it was `/vmunix`, approximately 1.3 MB.

were each repeated five times until all caches were warm, and five subsequent test results were averaged and rounded to two significant figures. The results are shown in Table 1.

Table 1. Baseline Measurements

Test	Performance
AIX/370 loopback FTP	4100 KB/s
RISC System/6000 Model 520 loopback FTP	1100
RT loopback FTP	1500

These data indicate that AIX/370 TCP/IP code can sustain about 4 MB/s when not hamstrung by network interface hardware. Somewhat surprisingly, they also show that the PC/RT's are faster than the RISC System/6000 machines when it comes to the TCP/IP code.

2.2 TCP DXU Performance

These tests were designed to measure TCP/IP performance through the DXU. The following tests were run: FTP transfer from AIX/370 to RISC System/6000; simultaneous FTP transfers from AIX/370 to two RISC System/6000's; FTP transfer between one AIX/370 guest and another; FTP transfer in both directions simultaneously between two AIX/370 guests; a repeat of the unidirectional transfer with the DXU driver's MTU increased from 4144 (the default) to 32767 bytes; and a repeat of the uni- and bi-directional tests with an 11 MB file and an 4144 MTU. In the latter five tests, packets travel between one NB225, the IP router, and the other NB225 without ever passing onto the FDDI ring. The results are shown in Table 2.

Table 2. TCP DXU Measurements

Test	Performance
One RISC System/6000	510 KB/s
Two RISC System/6000s	510 (each)
Two AIX/370 guests	1400
Bidirectional AIX/370	1000
Two AIX/370 guests, 32767 MTU	1400
Two AIX/370 guests, 11 MB	1400
Bidirectional AIX/370, 11 MB	940

Unfortunately, the Kingston FDDI cards can only sustain 510 KB/s. Given this low data rate, it is obvious that the DXU is capable of saturating two RISC System/6000 FDDI interfaces. Coupled with the small number of client machines, this low data rate made it impossible to examine the performance of the DXU at or near saturation.

In both tested cases, bi-directional performance is better than twice the unidirectional performance, indicating there is some overlapping going on, probably TCP piggybacking acknowledgement packets on data traffic headed in the opposite direction.

As expected, increasing the driver's MTU did not improve performance since TCP/IP's buffer sizes were left at 4 K. This test was done mainly to compare with Rx at the larger MTU.

2.3 Rx DXU Performance

One test was run to determine Rx performance over the DXU: 1000 Rx writes each of sizes varying from 1.5 K to 96 K bytes were sent between the two AIX/370 guests through the DXU. The device driver MTU was varied between 4144 and 32767 bytes. The test results are shown in Table 3.

Table 3. Rx DXU Measurements

	4144 MTU		32767 MTU	
Buffer Size	Elapsed	Throughput	Elapsed	Throughput
1.5 K x 1000	5.7 s	260 KB/s	6.3 s	240 KB/s
3K	7.3	410	7.2	420
4K	7.3	550	8.0	500
6K	9.9	610	10.1	590
12K	15.5	770	16.4	730
24K	25.3	950	27.4	880
48K	48.1	1000	51.1	940
96K	86.4	1100	94.2	1000

Here we clearly can see the throughput increasing with the size of the Rx write operation. In fact, curves plotted from these data show the throughput rising asymptotically. Unfortunately, at buffer sizes much above 96 K the test hangs irretrievably. Interestingly, Rx performance was slightly poorer with the larger MTU; the cause has not yet been determined. Viewed in combination with the result of the previous test, increasing the DXU driver's MTU has little positive effect.

It would have been instructive to repeat this test on the pair of RISC System/6000's. Unfortunately, the receiving FDDI card on these machines routinely crashed during the test attempts.

2.4 AFS DXU Performance

This test starts two simultaneous client requests (one request per RISC System/6000) to read the same 1 MB file; each client flushes the file from its cache before making the request. The elapsed time required to deliver the file to each client is recorded. This test was designed to parallel the clientload test of previous work [1], but here we are using FDDI and we are limited to two clients. Each test was run ten times with the results of the last five runs averaged. The results as shown in Table 4.

Table 4. AFS DXU Measurements

	1 MB Transfer		10 MB Transfer	
Client	Elapsed	Throughput	Elapsed	Throughput
maggie	3s	330 KB/s	38	270 KB/s
LISA CPU Util		-		50%
LISA IO/s		-		720
maggie	4 s	250 KB/s	44 s	230 KB/s
esa	4	250	44	230
Totals		500		460
FS CPU Util		-		49%
LISA CPU Util		-		80%
LISA IO/s		-		1200

The FS CPU entry records the CPU utilization of the file server process, expressed as a percentage of the total CPU resources available to **lisa**. The LISA CPU and LISA IO/s record **lisa's** CPU utilization and number of IO operations per second as obtained by a VM utility (RTMSF).

From these results we observe that throughput is well below the FTP limits measured earlier (250-330 KB/s vs. 510 KB/s for FTP).

Further, we see that two clients don't quite force twice as much throughput out of the server; the ratio is closer to 1.7. The guest CPU and guest IO/s ratios (1.6 and 1.7, respectively) are both very close to the overall throughput ratio, so it is not clear from these data which is the culprit. However, during one of the runs of the two client test, **lisa's** CPU utilization dropped to 50% due to an increase in CPU demand by the MVS guest. The overall performance did not vary due to this decrease in CPU utilization; therefore we must conclude that server performance is being limited by the I/O throughput rate and not by CPU starvation.

Again, it would be instructive to add several more clients to the FDDI ring and repeat this client-load test to stress the DXU more fully.

2.5 Clientload Benchmark

This benchmark is essentially a repeat of the clientload test previously done [1]. The clientload test starts fifteen simultaneous client requests (one request per Ethernet-connected PC/RT workstation) to read the same 1 MB file; each client flushes the file from its cache before making the request. The elapsed time required to deliver the file to each client is recorded. This test is repeated twice per hour throughout the night and the results are averaged. This test was conducted in order to observe the effect, at an Ethernet-connected workstation, of the faster DXU interface to the AIX/370 server. What we hope to observe is not an improvement at any given workstation, but a higher aggregate performance because data is presumably entering the campus token ring at a higher rate. The results are shown in Table 5.

Table 5. Clientload Benchmark

Client	Elapsed
barnone	147 s
cauchy	138
doom	152
eagle	146
ebbtide	149
eh	146
emptys	146
fleabag	141
flim	148
jackpot	148
nunn	147
pinhead	149
pumper	104
rioja	141
virgo	137
average	142 s

It must be pointed out that the packets generally arrive at the same Ethernet subnet 141.211.168 and contend for bandwidth there (**pumper** alone resided on a different topologically closer subnet). A better version of this test would scatter the receiving workstations around on different subnets, but this was not possible given the scope of the test and the number of readily available workstations on other subnets. However, consider that receiving 14 concurrent copies of a 1 MB file in 142 s requires a throughput of 99 KB/s, well below the point at which an Ethernet begins to degrade due to excessive collisions.

These results are very close to those previously obtained, using a Bus-Tech Incorporated ELC2 as the channel connection [1]. In that study we obtained numbers ranging from 97 s to 144 s, depending on the machine configuration and loading. Because this test was the equivalent of a loaded test, we must compare our result with the highest previous figure. Even allowing for increased network and server loading in our current test, we observe no appreciable performance improvement at the end-user workstation. Clearly, the network connection between the DXU and the end-user workstations is causing this degradation, since the local FDDI ring tests elicited good performance from the DXU.

3. Conclusions

Based on these results, it is clear that the NSC DXU was easily able to handle the throughput rates imposed on it by our somewhat limited performance tests. In point of fact, it is indeed a big, fat pipe attached to the end of a FIPS channel. The problem, as evidenced by Table 5, is that other bottlenecks at the end of this pipe prevent any performance improvement from being seen at client workstations, within the limited scope of this evaluation. What the DXU does, in fact, is move the bottleneck elsewhere.

Therefore, we cannot recommend purchasing an NSC DXU for IFS file service at this time, since a slower channel interface, the BTI ELC2, can provide less but acceptable client throughput at a much lower cost.

However, when the downstream bottlenecks have been analyzed and removed (clearly, the performance of packet traffic between the test FDDI ring and the campus Proteon rings bears closer examination) it would certainly be appropriate to purchase one or more NSC DXU's, for it would then be possible to exploit the increased performance offered by this channel interface.

4. Acknowledgements

Brian Cashman and Bert Rossi at the University of Michigan provided assistance in connecting the FDDI ring to the campus backbone networks. Chuck Lever at the University of Michigan lent his expertise during the early part of this effort. Binh Hua of IBM KPL and Mark Bell of NSC provided technical assistance during the evaluation. Julian Thomas of IBM KPL provided a pair of FDDI cards when no one else could. Dave Cherkus of IBM DSD pointed out the existence of the DXU driver in the NIFL library. Thanks also to the reviewers for their insightful comments. Finally, thanks to Kathy Rozek of NSC for her support of this evaluation, for arranging an extension of the initial evaluation period, and for her patience in awaiting the completion of this report.

References

1. Charles J. Antonelli, Steve Burling, and Lee Pearson, "Measuring IFS AIX/370 File Server Performance in LPAR Mode," CITI Technical Report 91-2, Center for Information Technology Integration, Ann Arbor, MI (October 1990).

2. Network Systems Corporation, "Data Exchange (DX) Systems Description Guide," Publication NS1167 (1990).

3. Information Technology Services, *FTP Performance of AIX/370 and VM TCP/IP*, Rensselaer Polytechnic Institute, Troy, NY (December 1990).

4. A.T. Salinas, "AIX/370 LAN Perfornance," GG66-3180-00, Washington Systems Center, Gaithersburg, MD (August 1990).