SLAC-PUB-6567
August 1994
(M)

Adventures in the Evolution of a High-Bandwidth Network for Central Servers [†]

Karl L. Swartz, Les Cottrell, and Marty Dart
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94309

ABSTRACT

In a small network, clients and servers may all be connected to a single Ethernet without significant performance concerns. As the number of clients on a network grows, the necessity of splitting the network into multiple sub-networks, each with a manageable number of clients, becomes clear.

Less obvious is what to do with the servers. Group file servers on subnets and multi-homed servers offer only partial solutions—many other types of servers do not lend themselves to a decentralized model, and tend to collect on another, well-connected but overloaded Ethernet. The higher speed of FDDI seems to offer an easy solution, but in practice both expense and interoperability problems render FDDI a poor choice. Ethernet switches appear to permit cheaper and more reliable networking to the servers while providing an aggregate network bandwidth greater than a simple Ethernet.

This paper studies the evolution of the server networks at SLAC. Difficulties encountered in the deployment of FDDI are described, as are the tools and techniques used to characterize the traffic patterns on the server network. Performance of Ethernet, FDDI, and switched Ethernet networks is analyzed, as are reliability and maintainability issues for these alternatives. The motivations for re-designing the SLAC general server network to use a switched Ethernet instead of FDDI are described, as are the reasons for choosing FDDI for the farm and firewall networks at SLAC. Guidelines are developed which may help in making this choice for other networks.

Introduction

In a small network, clients and servers may all be connected to a single Ethernet. This simple approach provides fast and relatively reliable communications between clients and servers. Unfortunately, it does not scale well—performance may suffer as the addition of more hosts (and thus more traffic) brings on network congestion, reliability may suffer as the result of there being more pieces in the Ethernet that could fail in a manner which impacts the entire network, or simple physical limitations may be reached. Splitting the network into multiple subnets works for the clients, but what to do with the servers may be far from obvious.

Keeping servers close to clients

Ideally, one would like to preserve the simplicity inherent in having clients reach their servers over a single Ethernet. Forcing traffic to traverse multiple networks adds delay and introduces additional opportunities for failure, especially if the routers become congested. NFS in particular is very sensitive to congested intermediate routers and responds to the situation in a most ungraceful manner[1, 2].

Many vendors would like everyone simply to buy workgroup servers and distribute them amongst the client networks. This can work quite nicely if groups within your organization are neatly compartmentalized and you can afford to buy servers for each of them, but substantial interactivity between groups will reduce the effectiveness of this solution. Institutional databases take this to the extreme, yet they are commonplace.

Presented at the USENIX Association's Eighth Large Installation System Adminstration Conference (LISA VIII), San Diego, California, September 19-23, 1993.

[†] This work supported by the United States Department of Energy under contract number DE-AC03-76SF00515.

Inherently centralized services

SLAC is an experimental physics laboratory, and the physics data is at the core of our computing. Today that means a few terabytes of data in four tape silos, with a new experiment and hundreds of terabytes of data looming ominously on the horizon. With various groups within the laboratory working together to collect and study such large amounts of data from a single experiment, departmental servers are not viable. A similar situation exists in many other organizations—airlines and their reservation databases are a striking example, though most any organization probably has examples.

Supercomputers pose a similar problem; the only difference is a change in perspective, with computing cycles instead of data as the shared, central resource. More common are mainframes, which represent a mix of shared data and computing cycles.

Even in a more enlightened world, where such dinosaurs have been banished to Hollywood, centralized services will persist. Firewall gateways to the Internet and NetNews (which really is just another big database) come to mind, as do mail routers, even if departmental mail servers handle part of the load. The problem isn't likely to go away.

Multi-homed servers

Connecting a few large servers to multiple networks—multi-homing them—appears to provide a reasonable compromise. Auspex servers are designed with this in mind, and Sun, for example, seems to encourage using their larger servers this way. SLAC has implemented multi-homing on a limited basis[1], but, like any compromise, this solution is not perfect. The added complexity is perhaps the worst problem—even after investing a great deal of effort, multi-homing causes confusion amongst users and administrators, while certain applications simply don't work on multi-homed hosts.

Availability of I/O slots in the servers and the cost of additional Ethernet interfaces places further constraints on widespread multi-homing of servers. While connecting a few big servers to a few busy networks helped, a lot of smaller servers talking to a lot of quieter networks still created a tremendous load. Individual server-network pairs could not justify additional direct connections, but in aggregate, the server network was still very congested.

A bigger pipe

The need for higher bandwidth amongst the central servers and core routers suggested a switch to something faster than Ethernet. Conventional wisdom suggested a switch to FDDI[3], with bandwidth at least an order of magnitude greater than Ethernet. (100 megabit Ethernet hadn't entered the scene yet.) Interfaces were expensive and availability spotty, but there seemed to be a strong movement towards FDDI and we felt the situation would improve by the time we needed a substantial investment in FDDI.

Building an FDDI network for the central servers would of course mean there would be at least one router hop between the servers and the Ethernet-based clients. In part, we hoped to minimize the risk by giving every Ethernet a direct connection to the FDDI ring, keeping client-server communications to at most one router hop, and by using fast routers[1] that should easily be able to keep up with an FDDI and a handful of Ethernets. For the common 8 kB NFS reads, the greater maximum transfer unit (MTU) of FDDI would also mean the router would only see two packet fragments instead of six, reducing the vulnerability of a packet to fragment loss.

Various features of FDDI promised further reliability benefits. The ability to "heal" the ring by wrapping back upon encountering a failed node was particularly attractive, as was the ability to further cordon off problems by isolating servers behind wiring concentrators.[2]

Overall, we felt that FDDI represented an improvement in reliability despite the added router hop. We still had the option of directly connecting servers (i.e. multi-homing them) to networks for which highly reliable connectivity was paramount.

Experience with FDDI

After several years, our experience with FDDI has been less idyllic than we had hoped for. Prices have come down somewhat, but FDDI interfaces and other devices are still expensive. Implementations have been buggy and have exhibited various interoperability problems. Identifying and solving problems has been hampered by inadequate diagnostic and monitoring tools as well as the ignorance of vendors and ourselves. When the networking is working well, it's not uncommon to find that other software is not prepared to take advantage of the faster speeds.

We began with a ring composed of three devices, shown in figure 1: a Cisco AGS+ router, a Sun SPARCserver-390 with a Sun FDDI/DX interface, and a DEC wiring concentrator. A VAX-9000/410 (running VMS and Multinet) was connected via the concentrator. Despite the mixture of vendors, things worked pretty well, which was a good thing since we had no way to diagnose problems.

The next step was to add a SPARCserver-2 and an RS/6000-340. Unlike Sun's VME bus FDDI/DX interface, the SBus FDDI/S card only implements a single attached station (SAS), so it had to be connected to the DEC wiring concentrator. The RS/6000

---

[1] Each FDDI router at SLAC is a Cisco AGS+.

[2] FDDI offers a variety of redundancy and fault isolation features. See [3] for a good, introductory discussion of FDDI's features.

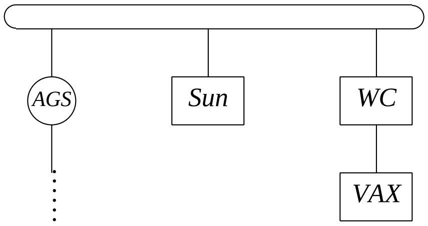


Figure 1. Initial SLAC FDDI network

had IBM's optional second board which allowed a dual attached (DAS) connection, i.e., directly to the ring, but we had decided that we only wanted routers and concentrators on the ring, so it was connected to the concentrator—the FDDI specification allows DAS devices to be connected in a SAS configuration. (We later connected the SPARC 390 this way too.)

Big Blue blues

The RS/6000 is where we encountered our first trouble. Always eager to annoy, AIX came with FDDI support, but it didn't work—a special microcode update was necessary. We obtained that and installed it, but the FDDI still didn't work. While checking the cabling, it was noted that the fiber was plugged into the second card of the IBM adapter. This adapter is composed of one MicroChannel card which implements the bus connections and a SAS interface, while a second, optional, MicroChannel card adds the DAS capability. Despite claims that it shouldn't matter, it was found that moving the fiber to the main adapter card allowed the connection to work. This problem was to become quite familiar, not only as technicians miscabled RS/6000s (about half the time), but also in some more perplexing ways, to which we'll return.

A channel connection between an RS/6000 and our mainframe, an IBM ES/9000 running VM/CMS, was the next source of trouble. While not really an FDDI problem, it was a direct result of FDDI, and dramatically illustrates the costs of FDDI. Specifically, the VM system is still a critical part of SLAC's computing environment, and a prime candidate for an FDDI connection. Unfortunately, IBM wanted $80,000 for a 370 FDDI adapter! We figured we could build one with an RS/6000, routing packets between FDDI and the mainframe channel for well under half that price. We could, and did, but it's always been cantankerous.

While the FDDI support in AIX still exhibited a number of problems, they were minor enough that we felt we could work around them until fixes arrived from Austin. Things seemed stable enough for us to put our main NFS fileserver, an RS/6000-970, onto the FDDI ring, and later our two Oracle servers. Despite the dramatically higher load, things seemed to be going well—for a while. Then, we started noticing occasional NFS hangs. They seemed to occur during periods of high load, and to last for about 90 seconds. Eventually, we managed to correlate them to an FDDI adapter error on the RS/6000-970, and found examples of the error in the logs of other RS/6000s. IBM didn't seem to know what was going on, but they did proffer a blizzard of new system patches.

All of our RS/6000s with FDDI had been ordered with the second card to provide DAS capability, and all of them were connected SAS-fashion via a concentrator. Once again, that extraneous card came to mind as a possible culprit. A check revealed that most of the RS/6000s were miscabled—but now they worked, at least most of the time! We had switched to a Cabletron wiring concentrator somewhere along the way, and apparently it wasn't as fussy about cabling as the DEC concentrator had been. Still, that extra card was a suspect, so we called in IBM field service to have the board removed from the NFS server. Upon arriving, the technician refused to remove the board, claiming the extra board provided additional reliability. He'd obviously heard about FDDI's ability to wrap around failed devices, which only works for dual-attached stations, but he didn't understand enough to appreciate IBM's own recommendation that hosts not be directly attached[4].

Eventually, we did get the extra card removed from one machine, but the problem persisted. An upgrade to AIX 3.2.5 appeared to produce a reliable FDDI connection for our RS/6000s at last, but not without one last round with the extra cards—while nobody could prove any problem with SAS-cabled DAS systems running older versions of AIX, IBM Austin knew for a fact that this configuration would not work with AIX 3.2.5. We finally removed all of them once and for all and installed 3.2.5. Only one bug remained, and while confusing, it was harmless if you were aware of it. After a mere 18 months, we had what seemed to be production-quality networking.

At least that's what we thought. Then we suffered another 90 second outage of the file server, again correlated to an FDDI adapter error but not traceable to any other event on the network. While the problem is much less obtrusive now than it was previously, due to the lower frequency of occurance, it remains unresolved.

Sun brings darkness to the fiber

IBM wasn't the only vendor to bring grief to our FDDI effort. A new SPARCserver-10, again with Sun's FDDI/S adapter, was installed, its FDDI interface was configured, and all was well—for a minute or two. Then the wiring concentrator started having convulsions, and the entire ring crashed. Disabling the Sun's FDDI interface restored the ring, though at least once a wiring concentrator crashed hard enough to require cycling its power. Replacing the FDDI/S adapter was ineffective. A patch was obtained from Sun which addressed a problem with frequent ring state transitions on the FDDI/S adapter, but the only effect seemed to be to reduce error messages on the console. (We later discovered that that is all the patch was intended to do!) The SPARC-2 had never had these problems, but for other reasons was no longer on the FDDI, and a lot had changed since it had been. Puzzled, and needing to get other work done, we temporarily shelved the problem.

When we came back to it a few months later, we started from scratch. SunOS was re-loaded from CD ROM and the FDDI/S software was installed. Every SPARC 10 and FDDI patch we could find was applied. We scheduled an outage, checked and re-checked all the cables, and finally enabled the FDDI interface once again. It worked. The network map stayed green, the concentrator hummed along peacefully, and everything did exactly what it was supposed to do, even after a week had gone by.

In reviewing what had changed in the past few months, we found that our Cisco routers (there were several on the ring by now) had received a microcode update that fixed a hyper-sensitivity to ring state transitions—exactly the situation which that first Sun patch was supposed to have addressed. We subsequently found a review of SBus FDDI adapters which documented the problem of frequent ring resets with the Sun FDDI/S adapter[5]. It appears that the Sun adapter bug had been aggravating the microcode bug in the Cisco routers, which then not only crashed themselves but also took out the Cabletron wiring concentrators—something which isn't supposed to happen.

What went wrong?!

In spite of work dating back an entire decade, FDDI clearly is not mature yet. High costs have undoubtedly inhibited widespread deployment of FDDI, and our mixture of products from at least half-a-dozen vendors is probably a greater interoperability challenge than we would have liked. Considering the number of substantial interoperability problems with Ethernet, even after more than two decades of use in far more diverse environments[6, 7], it shouldn't come as much of a surprise that the more esoteric (and far more complex) FDDI still has a lot of bugs to be uncovered.

Monitoring and troubleshooting FDDI

Debugging FDDI problems and monitoring the health of the network has also proven to be problematic due to a lack of experience, compounded by inadequate tools. The case of the IBM technician who knew only the sparsest details of FDDI is by no means an isolated case. The SPARC-10 case demonstrated that sufficiently in-depth experience with FDDI within SLAC was equally lacking.

value	alert if exceeds
CRC errors and alignment errors	1 in 10k packets
total utilization on a network	10% for the day
broadcast rate	300 per second
(shorts+collisions)/good_packets	10%
packet losses from ping tests	1% in a day

2a. Ethermeters (RMON data)

value	alert if exceeds
CRC errors and alignment errors	1 in 10k packets
buffer, controller overflows	0

2b. bridges

value	alert if exceeds
total interface input errors	1 in 10k packets
collision rates	10% of packets
CRC errors and alignment errors	1 in 10k packets
buffer, controller overflows	0
in/out queue drops and discards	0
ignored packets	0
interface ping packet losses	1%

2c. routers

Figure 2. SNMP data and alert thresholds

The acquisition of a Tekelek FDDI analyzer helped with debugging to some degree, and with testing new equipment. It's mainly oriented towards the hardware, though, and the lack of a device which understands the higher-level protocols has been a handicap. (Network General's Sniffer now has an FDDI option which brings this capability to FDDI.)

Routine monitoring is also a problem. For Ethernets, we put an NAT Ethermeter on each major segment and use RMON to collect a variety of performance and error information. Values which exceed certain thresholds, as shown in figure 2, trigger alerts. Further data is collected via SNMP from bridges and routers and from interesting hosts[8, 9]. Alas, no FDDI "Ethermeter" is available yet, and FDDI MIBs in the various devices are incomplete or non-existent. Even if we did have the data, the lack of baseline information makes problem threshold determination difficult, as compared to Ethernet, which by now is well understood.

FDDI performance

Given a functional FDDI network, another hurdle is getting software to take advantage of it. Some kernel tuning was required, especially on AIX, to allocate enough mbufs for the higher data rates, to increase the default TCP buffer size, etc.

Using the larger MTU is also important—one experiment was using NFS to read data from the VM system to the VAX 9000 (admittedly not an ideal choice) and getting horrible performance while making prodigious use of CPU cycles on the mainframe.

It was found that, even with a direct FDDI link, the Multinet NFS software on VMS was using a 512 byte read size. Forcing a 4096 byte size, which fits nicely in FDDI's 4352 byte MTU, improved the performance dramatically. AFS, which SLAC is starting to deploy, similarly used a small MTU, though not so small as to be inefficient even for Ethernet. In this case, the MTU was not tunable but the problem was fixed in AFS 3.3[10, 11].

Router performance with FDDI was disappointing as well, at loads well below what Cisco's numbers suggested should be easy for an AGS+ to handle. While our findings weren't rigorously documented, one subsequently published test demonstrated only 12 Mb/sec when routing from one FDDI to four Ethernets[5]. An average of 3 Mb/sec per Ethernet is 30% of the bandwidth, not a light load, but still below what one could expect from an uncongested Ethernet[12].

Investigation of the router statistics turned up a large number of dropped packets on the FDDI inputs. It turns out that Cisco routers allocate the same amount of buffer memory for each interface, regardless of the bandwidth of the interface[13]. Thus, while the Ethernets had an abundance of buffer space, the FDDI was starved, and incoming packets were being dropped, which, as mentioned in the discussion of avoiding router hops, is particularly bad for NFS traffic. Despite tight budgets, we were forced to upgrade our existing routers and acquire additional routers, which eased the problems.

Analyzing the server network

The ongoing problems and disappointing performance of FDDI, along with the cost of equipping our growing number of servers with FDDI interfaces and adding concentrator ports for them, led us to revisit the decision to use FDDI for our server network. Because of the large volumes of data moving across our network, and because UNIX represents only a small (but growing) part of the computing environment at SLAC, our performance monitoring and problem detection efforts have tended to focus on the networks [8, 9] rather than individual servers.[3]

The most interesting information for studying the server network proved to be the "Top 10 Talkers" report, which, for a given Ethernet segment, shows the top ten source/destination pairs seen in packets during a given hour, with summary reports for yesterday and for today so far. Most pairs on the server network tended to involve at least one router—the information is based on hardware (MAC) addresses, not IP addresses, so traffic going on or off the network has a router on one end.

A significant percentage of traffic involved two routers, with one being the firewall router which connects the SLAC network to the Internet. One could view this as transit traffic which shouldn't be on the server network. A more general view is to look at the Internet as being just another service, albeit a rather special one, with the firewall router being the server providing that service.

Notably, no pair dominated the traffic on the network, and only one pair (the firewall router to our best connected internal router) consistently exceeded 10% of the total traffic. Except for a few short-term anomalies, intra-server traffic only occasionally made the top 10 list, with the RS/6000 fileserver being involved in most such cases.

Further analysis was done using Sun's `etherfind` utility and by examining the usage count field in `netstat -r` output on major servers. This further bolstered the model of lots of servers, each contributing a modest (in terms of Ethernet bandwidth) amount of traffic to the server network, aimed at a variety of clients.

This finding suggested that providing bandwidth greater than that of an Ethernet to each server was unnecessary. Aggregate bandwidth of the server network was what we needed.

A switch-based server network

We had already looked at Ethernet switches for other purposes, but now began to study them as an alternative to FDDI and simple Ethernet for our server network. They offered the advantages of Ethernet for the server connections—low cost per server with thoroughly tested hardware and software—with much higher aggregate bandwidth across the network.

Out of several possible choices, we found the Alantec PowerHub's unique routing capabilities [16] to be intriguing. Our original idea for a switch-based server network treated routers just like servers, as in figure 3a. The Alantec allowed us to skip routers when going to the most critical networks, leading to the network structure in figure 3b. (Routers are still used for non-IP traffic and for networks which are not directly connected to the switch; they were omitted from the diagram for clarity.)

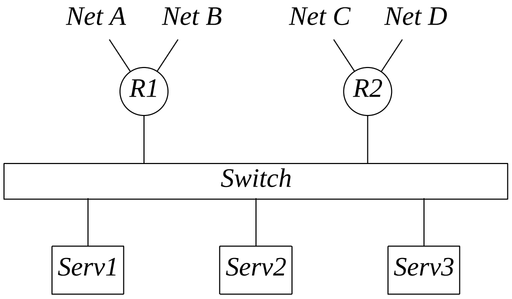
The initial PowerHub 3500 does not support enough ports to dedicate one to each server, but careful balancing of servers amongst the available ports makes this tolerable. Eliminating the need for this manual balancing will likely make the PowerHub 7000 [17] an attractive future upgrade.

Performance benefits of switched Ethernets

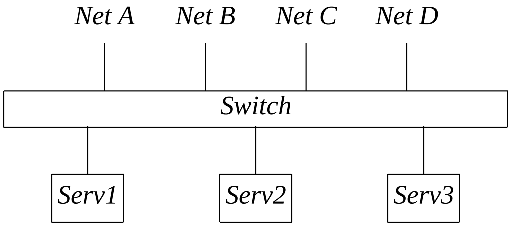
Every node that is added to the collision domain of a CSMA/CD network (such as Ethernet) cuts the available bandwidth for the other hosts on the network two-fold. The new node cuts out a share of the

---

[3] An interesting approach to automated system monitoring (which could be applied to network performance monitoring as well) is contained in [14]. For a discussion of NFS performance monitoring, see [15].



3a. routers to client networks



3b. switch does routing to client networks

Figure 3. Switch-based network topologies

bandwidth for it's own communication, then takes another cut because increasing the number of nodes on a broadcast network decreases the maximum throughput possible—two nodes communicating on an Ethernet can transmit at nearly the full 10 Mb/sec available, but an Ethernet that has many nodes will see the effective throughput of the medium drop to about 4 Mb/sec.

A switched Ethernet allows for aggregate throughput to increase everytime a node is added. There is a limit, of course, and it is dependent on the backplane internal to the switching hub, usually at least several hundred Mb/sec. (The Alantec PowerHub 3500 selected by SLAC has a 400 Mb/sec backplane.) Unlike a 10BaseT hub, a switched Ethernet hub supports a separate collision domain on each segment attached to it. If only one node is connected to a port, it has the capability of transmitting or receiving data at a full 10 Mb/sec. Thus, adding another node to a second port adds another 10 Mb/sec to the aggregate bandwidth, and so on for each additional connection.

Other advantages of switched Ethernets

At first glance, the price of a switched network seems prohibitive. Installation of non-switched LANs start at about $200 per node for a 10BaseT hub, then can quickly climb to $1,000 per node for switched Ethernet, and $2,500 per node for FDDI[18]. However, this perspective ignores performance considerations. Factoring in the bandwidth of the network, switched Ethernet becomes very attractive at a mere $100 per node times Mb/sec (*N·Mb/sec*), followed by FDDI at $250 per *N·Mb/sec*, and finally 10BaseT at $600 per *N·Mb/sec*.

The isolation between different Ethernet segments afforded by switches can also reduce the likelihood of interoperability problems such as those described in [7]. Store-and-forward switches such as the Alantec offer more isolation than cut-through designs such as Kalpana's EtherSwitch, at the cost of higher latency[19].

Disadvantages of switched Ethernets

Switches are not without cost, however. Truly dedicated ports preclude Ethermeters or other monitoring devices, and even if they didn't, the cost of an Ethermeter for each port would be prohibitive. It may be possible to send all or selected traffic to a designated monitoring port, but this defeats much of the point of switches, and if the network is busy the monitoring port will surely be flooded. Sending only selected data to the monitoring port may keep the load down, but precludes any on-going monitoring and automated problem detection. What's really needed is for the switch itself to provide full RMON data for each port.

A switch also represents a single point of failure, a grave concern for a server network which is critical to most of an organization. A coaxial cable may not have the bandwidth of an Ethernet switch, but it also doesn't have power supplies and software which can fail. Some switch vendors are addressing these concerns by offering redundancy and hot-swappable power supplies and other modules.

Adding ports may be another hurdle. Kalpana's solution is to cascade EtherSwitches, but this creates a potential bottleneck in the Ethernet between switches. Alantec's 3000 and 5000 models are somewhat limited in the number of ports they can support, but multiple switches can be chained together using FDDI. While this offers higher bandwidth than the Kalpana solution, the marginal cost of the next port after all ports on the first Alantec switch have been used is exceedingly high. Fortunately, newer switches are appearing with dramatically greater capabilities for port expansion.

Finally, the process of selecting a switch is difficult, since each switch seems to have a remarkably different feature set.

FDDI still has a place

Switches can be useful tools, but there remain network applications where they cannot substitute for high-speed networks such as FDDI or 100 Mb/sec Ethernet. The computing environment for the next major experiment at SLAC, the Asymmetric B Factory, and our planned T3 (45 Mb/sec) connection to the Internet are two such examples.

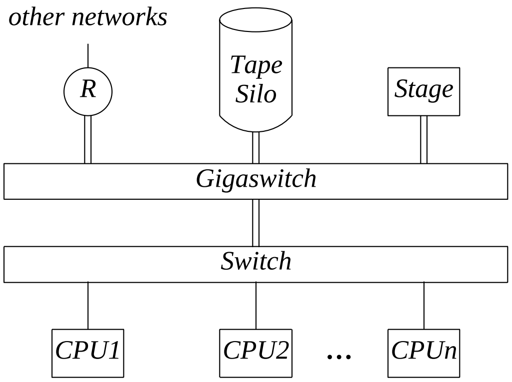


Figure 4. Compute farm for Asymmetric B Factory

The B Factory will involve several hundred terabytes (tera = 10^{12}) of data by the end of the experiment. This data will be stored in a complex of StorageTek silos, and off-line analysis will be done by a farm of workstation-class machines, as illustrated in figure 4. To optimize use of the tape drives, tape data will be staged to disk. Many of these data paths individually require FDDI speeds, and the aggregate speed of the network well exceeds FDDI, so a DEC Gigaswitch will be employed as the backbone. Each port on the Gigaswitch will in effect be its own FDDI ring, reducing concerns about FDDI interoperability. The compute servers in the farm do not individually require high bandwidth, so they will be connected with Ethernet to an Alantec switch, which in turn will connect to the Gigaswitch via FDDI.

The upgrade of SLAC's primary Internet connection from T1 to T3 provides a simpler, and slightly more commonplace, example of the need for networks with FDDI speeds. An Ethernet can readily handle traffic at a T1 line's 1.544 Mb/sec, but a T3, at 45 Mb/sec, is far beyond the ability of an Ethernet. The firewall router, which as seen above already contributes a sizeable amount of traffic to the server network, will be replaced with a larger router connected directly to an FDDI ring. From there, it will be able to send packets to various internal routers and to the B Factory compute farm at full T3 speed. Except for the Gigaswitch, all devices on this FDDI will be Cisco routers, so interoperability concerns are again minimal.

Conclusions

FDDI can handle high traffic volumes to and from a single server, but is expensive and still not mature. In a network with a number of smaller servers handling clients on a variety of networks, aggregate bandwidth of the server network may be more important than the capacity of the connection to individual servers, in which case an Ethernet switch offers higher bandwidth at a lower cost and with fewer potential interoperability problems. Switches are not a universal solution, however—FDDI still has a place where high bandwidth to a single server is required. Good monitoring of the network to characterize traffic patterns is invaluable for choosing the best solution.

References

1. Karl L. Swartz, ‘‘Optimal Routing of IP Packets to Multi-Homed Servers,’’ *Proceedings of the 6th USENIX Large Installation System Administration Conference (LISA VI)*, pp. 9-16, Long Beach, October 1992. Also published as SLAC-PUB-5895.
2. Hal Stern, *Managing NFS and NIS,* O'Reilly & Associates, Inc., Sebastopol, California, 1991.
3. Art Wittmann, ‘‘An FDDI Primer: Riding the Photons,’’ *Network Computing*, pp. 134-144, January 1993.
4. *IBM FDDI Workstation Adapters,* IBM Product Announcement 192-132, IBM Corp., May 19, 1992.
5. Todd Tannenbaum and Michael Lee, ‘‘FDDI Adapters Take The SBus To High Performance,’’ *Network Computing*, pp. 108-110, April 1, 1994.
6. Bob Metcalfe, ‘‘From The Ether,’’ *InfoWorld*, November 15, 1993.
7. Wesley Irish, ‘‘High utilization Ethernet performance problems traced to controller,’’ *comp.dcom.lans.ethernet (Usenet newsgroup)*, October 15, 1993.
8. Connie Logg and Les Cottrell, ‘‘Adventures in Network Performance Analysis,’’ talk at the *1994 IEEE Network Operations and Management Symposium*, Kissimmee, Florida, February 1994.
9. Connie Logg and Les Cottrell, ‘‘Network Performance Monitoring and Analysis at SLAC,’’ talk at the *1994 Dept. of Energy Telecommunications Conference*, Baltimore, August 1994.
10. Lyle Seaman, ‘‘AFS Performance Evaluation: Observations, Analysis, and Plans,’’ *Proceedings of DECORUM '94*, Orlando, March 1994.
11. *AFS 3.3 Release Notes*, p. 51, Transarc Corp., Pittsburgh, January 1994.
12. David R. Boggs, Jeffrey C. Mogul, and Christopher A. Kent, ‘‘Measured Capacity of an Ethernet: Myths and Reality,’’ *Computer Communications Review*, vol. 18(4), pp. 222-234, ACM, August 1988.
13. Michael Lee and Art Wittmann, ‘‘Four Steps To Better FDDI,’’ *Network Computing*, pp. 140-141, April 1, 1994.
14. Peter Hoogenboom and Jay Lepreau, ‘‘Computer System Performance Problem Detection Using Time Series Models,’’ *Proceedings of the USENIX Summer 1993 Technical Conference*, pp. 15-32, Cincinnati, June 1993.

15. Gary L. Schaps and Peter Bishop, ‘‘A Practical Approach to NFS Response Time Monitoring,’’ *Proceedings of the 7th USENIX Large Installation System Administration Conference (LISA VII)*, pp. 165-169, Monterey, California, November 1993.
16. *PowerHub Reference Manual*, Alantec, July 1993.
17. Skip MacAskill, ‘‘Alantec powers up new high-end switching hub,’’ *Network World*, p. 1,56, July 18, 1994.
18. Peter Sevcik and Mary Johnston Turner, ‘‘Enterprise Networks: Architecture and Planning,’’ tutorial at *Networld+Interop 94*, Las Vegas, May 1994.
19. J. Scott Haugdahl, ‘‘Switch trio boosts bandwidth,’’ *Network World*, pp. 47-54, July 25, 1994.

Trademarks

UNIX — AT&T
Cisco — Cisco Systems, Inc.

Acknowledgments

Thanks to Bob Cook, Charlie Granieri, Diana Gregory, Krissie Griffiths, Connie Logg, Jo Ann Malina, Alexander Shaw, Lois White, and others whose contribution is not diminished by our failure to credit them. Our gratitude goes to all of them.

Author Information

Karl Swartz is a member of the Server Systems Group within SLAC Computing Services at the Stanford Linear Accelerator Center, where he is the resident UNIX guru. Prior to joining SLAC, he worked at the Los Alamos National Laboratory on computer security and nuclear materials accounting, and in Pittsburgh at Formtek, a start-up that is now a subsidiary of Lockheed, on vector and raster CAD systems. He attended the University of Oregon where he studied computer science and economics. Karl instructs at high-speed driving schools and enjoys good food and good beer (though not while driving) and hikes on the beach with his Golden Retriever, Alexander. Reach him via e-mail at kls@chicago.com or kls@slac.stanford.edu, or see his Web page at http://www.slac.stanford.edu/˜kls/kls.html.

Les Cottrell left the University of Manchester, England in 1967 with a Ph.D. in Nuclear Physics. He joined SLAC as a research physicist focusing on real-time data acquisition and analysis. In 1972/73 he spent a year’s leave of absence as a visiting scientist at CERN in Geneva, Switzerland, and in 1979/80 at the IBM UK Laboratories at Hursley, England. He is currently Assistant Director of SLAC Computing Services and focuses on networking and distributed computing technologies. Reach him via U.S. Mail at Mail Stop 97, SLAC, P.O. Box 4349, Stanford, California, 94309. Reach him via e-mail at cottrell@slac.stanford.edu, or see his Web page at http://www.slac.stanford.edu/˜cottrell/cottrell.html.

Marty Dart is a Network Engineer at SLAC where he previously worked as a Technical Coordinator and Network Technician. He received his BSEE from San Francisco State University in 1988. Marty can be reached via e-mail at dart@slac.stanford.edu, or via U.S. Mail at Mail Stop 97, SLAC P.O. Box 4349, Stanford, California, 94309.