ULTRIX Fiber Distributed Data Interface Networking Subsystem Implementation

By Ursula Sinkewicz, Chran-Ham Chang, Lawrence G. Palmer, Craig Smelser, and Fred L. Templin

Abstract

The ULTRIX operating system, Digital's version of the UNIX operating system, supports the first implementation of a host networking subsystem with a fiber distributed data interface (FDDI) network adapter. Digital's FDDIcontroller 700 adapter provides a single FDDI attachment for the reduced instruction set computer (RISC)-based, DECstation 5000 model 200 platform. Combined with the ULTRIX networking subsystem, this adapter brings high-speed communication directly to the workstation.

Introduction

Digital made the decision to adopt fiber distributed data interface (FDDI) local area network (LAN) technology to follow Ethernet. With the FDDI system, Digital is developing products to support improved network performance such as the high-speed interconnection of workstations.

The ULTRIX operating system supports Digital's first implementation of an FDDI host networking subsystem. A key decision in the ULTRIX FDDI program was to design an adapter for reduced instruction set computer (RISC)-based workstations. Consequently, the DEC FDDIcontroller 700 network adapter was designed to support an FDDI single attachment for the DECstation 5000 model 200, RISC-based workstation. This support covers the Defense Advanced Research Projects Agency (DARPA) internet network protocols designed for the ARPANET packet-switched network. The DARPA internet network protocols include the internet protocol (IP), the transmission control protocol (TCP), and the user datagram protocol (UDP).

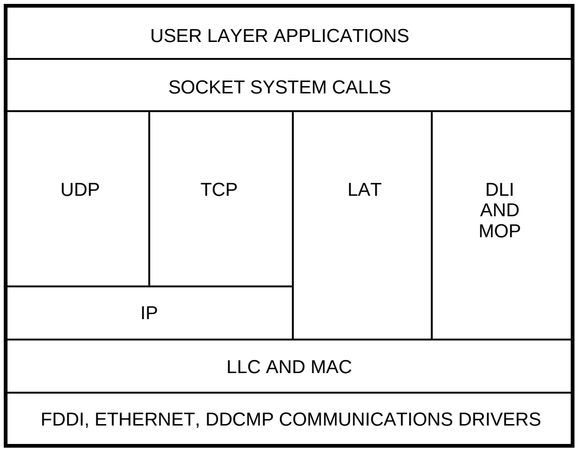
This paper begins with an overview of the ULTRIX operating system. The sections that follow present the implementation details of the network and communication driver, review specific issues in the ULTRIX FDDI implementation, and discuss both performance and future directions.

Overview of the ULTRIX Operating System

The ULTRIX operating system is based on the 4.3 BSD system. (BSD refers to Berkeley Software Distribution, a popular version of the UNIX operating system.) As in other systems based on the UNIX system, the ULTRIX operating system operates in user and kernel modes. A process running in user mode can be preempted. Interrupts are run in the context of the current process. A process running in kernel mode voluntarily relinquishes control of the CPU. ULTRIX networks and communications device drivers run in kernel mode.

The ULTRIX operating system supports network activity through a well-defined, layered hierarchy including user applications, system calls, and compile-time entry points to the protocols and communication device drivers. The layered hierarchy is illustrated in Figure 1.

The user layer consists of applications (e.g., electronic mail) that use specific system calls to support network activity. These interprocess-communication system calls incorporate the notion of a socket and, hence, are referred to as socket system calls. Sockets are endpoints of communication containing information used by the operating system to associate data with specific clients and servers. When executing in kernel mode, the socket system calls perform the memory management, the security checking, and the state management common to all protocols. When the protocol-common processing is complete, the operating system accesses a protocol switch table containing vectors to protocol-specific modules. These modules, in turn, access communications drivers through the network interface table.



KEY:

UDP	USER DATAGRAM PROTOCOL
TCP	TRANSMISSION CONTROL PROTOCOL
IP	INTERNET PROTOCOL
LAT	LOCAL AREA TRANSPORT
DLI	DATA LINK INTERFACE
MOP	MAINTENANCE OPERATION PROTOCOL
LLC	LOGICAL LINK CONTROL
MAC	MEDIA ACCESS CONTROL
DDCMP	DIGITAL DATA COMMUNICATION MESSAGE PROTOCOL

Figure 1 ULTRIX Network Subsystem Layering

The ULTRIX environment is characterized by a large number of servers, e.g., SUN's NFS system, which allows remote access to entire file systems, and many network applications. The servers support a diverse range of activities such as managing mail and ensuring that X Window System managers are available to remote workstations. The underlying protocols for most of these servers are TCP, IP, and UDP.

ULTRIX Support for the FDDI System—Development Strategies and Issues

Presentations by Digital's networking architects in May 1988 brought the earliest news that Digital was pursuing a timed token ring topology (i.e., FDDI) in contrast to Ethernet, which employs a carrier sense multiple access with collision detection (CSMA/CD) data link protocol. Digital's FDDI engineering program began with product requirements for a wiring concentrator, a bridge to link Ethernet and FDDI networks, and an FDDI adapter to the VAX computer. The FDDI program team planned only high-end system direct connectivity to the ring. Workstations would be connected through the existing Ethernet across a bridge.

However, the ULTRIX operating system running network applications on RISC workstations was already saturating the Ethernet. The ULTRIX engineering group advocated FDDI adapters, not only for RISC-based servers but also for the increasing number of high-end, RISC-based workstations.

ULTRIX and VMS engineering groups began architectural discussions with the FDDI development groups to write requirements for FDDI adapters for both RISC and VAX processors. Due to the evolving ULTRIX emphasis on RISC-based solutions, the ULTRIX engineering group represented data structure, virtual addressing, and performance requirements for RISC processors, while the VMS engineering group represented the same requirements for VAX processors. Approximately ten months after the initial network architecture presentations, the FDDI program team drafted product requirements for the ULTRIX implementation, including support for an FDDI workstation adapter.

To provide a workstation solution, members of the ULTRIX engineering group had already begun to work with the Low End Network Systems (LENS) Group on an advanced development project to define a workstation-based FDDI adapter. The team discussed alternatives for FDDI workstation connections, including the emerging DECstation 5000 model 200 TURBOchannel bus, the DECstation 3100 plug-in option, and the industry-standard small computer systems interface (SCSI) bus.

Six months into the adapter advanced development project, the internet community confirmed interest in TCP/IP implementations for FDDI requirements by issuing a draft of the request for comment, RFC 1103 (recently renamed RFC 1188), which defines the encapsulation of internet packets on FDDI networks. Members of the FDDI engineering team were instrumental in providing direction for the internet FDDI task force meetings on RFC 1103 and the FDDI network management information base (MIB). The draft of RFC 1103 prompted internet vendors to hastily implement FDDI workstation-based products and the LENS group to publish plans for FDDI connectivity to RISC-based workstations with ULTRIX support.

In October, at the Interop '89 Conference in San Jose, California, several internet vendors showcased FDDI products. Although Digital did not show FDDI products at the conference, this event prompted Digital to design an architecturally sound, high-quality, FDDI solution to gain a competitive edge.

Soon after the conference, the FDDI Data Link Specification and the project plan for ULTRIX support for the FDDI system were released. Subsequent ULTRIX development efforts to support the FDDI system produced new networking code for the TURBOchannel device driver, the data link layer, and the network layer. These efforts paralleled the TURBOchannel adapter development efforts.

A prototype ULTRIX implementation successfully passed 802.2 frames over an Ethernet connection, as required by the American National Standards Institute (ANSI) FDDI standard, to exercise the data link and network layer changes necessary for FDDI support. The product announcement for the TURBOchannel FDDI adapter assigned the official name DEC FDDIcontroller 700 to the adapter. Prototypes were delivered in May 1990; firmware integration was completed; and the first address resolution protocol (ARP) broadcast packet was sent over an FDDI ring from an ULTRIX host.

Device driver and adapter interoperability problems such as timing considerations, data corruption, and performance issues were solved promptly by close cooperation between the software and hardware groups. Several additional performance enhancements were added to the operating system, bringing the performance of the ULTRIX and adapter combination to nearly 40 percent of the entire FDDI bandwidth—a factor of four times greater than existing Ethernet implementations.

At its trade show, DECWORLD 1990, Digital announced the availability of its FDDI product offerings. These included the DECconcentrator 500 and DECbridge 500 products, and the DEC FDDIcontroller 700 adapter, which runs under the ULTRIX operating system.

ULTRIX Internals

The implementation of FDDI support in the ULTRIX operating system required the development of a link-level architecture and a network device driver. Operating system changes to improve the performance of the network were made later. The next two sections describe the implementation of the link-level architecture and the device driver. Performance changes are discussed in the ULTRIX Network Performance section.

Data Link Support

The ULTRIX operating system implements both the internet protocols (TCP/UDP/IP) and the Digital Network Architecture (DNA) model, including the Digital data link interface (DLI). In both the internet and DNA models, the data link defines services known as the logical link control (LLC) and the media access control (MAC). A major challenge in the implementation of ULTRIX FDDI support was defining a set of common data link routines to satisfy the frame format requirements of both internet and DNA models in a heterogeneous LAN environment.

Prior to the introduction of the FDDI system, all ULTRIX internet networking for LANs ran over Ethernet networks using Ethernet V2 frame formats, even though ULTRIX DLI networking supported both Ethernet V2 and 802.2 LLC frame formats. Figure 2 illustrates the differences among the V2 Ethernet and 802.2 Ethernet and the FDDI frame formats. V2 and 802.2 frames include Ethernet encapsulation; 802.2 frames consist of the MAC, the LLC, and data segments. When the 802.2 frame is sent over the FDDI system, the FDDI framing adds the FDDI-specific encapsulation, as shown in Figure 2. In order to conform to the ANSI FDDI standards, which require 802.2 LLC frame formatting, members of the Internet Network Working Group wrote Internet RFC 1188.[1] This RFC specifies the rules for 802.2 LLC encapsulation of internet frames on an FDDI network. To meet the needs of both Ethernet and FDDI networks, we designed and integrated a set of network-common routines. These routines, the net_output( ) and the net_read( ) functions, support 802.2 encapsulation as required by RFC 1188.

net_output() Function. The net_output( ) function prepares packets for transmission by ULTRIX network communication device drivers. If a driver requires 802.2 LLC support, the net_output( ) function supplies the necessary header, prefixes the MAC header, and enqueues the packet to the appropriate communication driver for transmission.[2] The function, while supporting 802.2 LLC encapsulation, does not preclude protocol modules from supporting their own LLC formatting. The encapsulation is switch-driven so implementors can add special routines to the switch to either replace or bypass the 802.2 LLC encapsulation.

net_read() Function. The net_read( ) function is called by the communication drivers and prepares received packets for delivery to protocol modules. This function first identifies the protocol type from information contained in the MAC and LLC headers, places the packet on the corresponding queue, and finally schedules a software interrupt to alert the appropriate protocol module of the arriving packet.

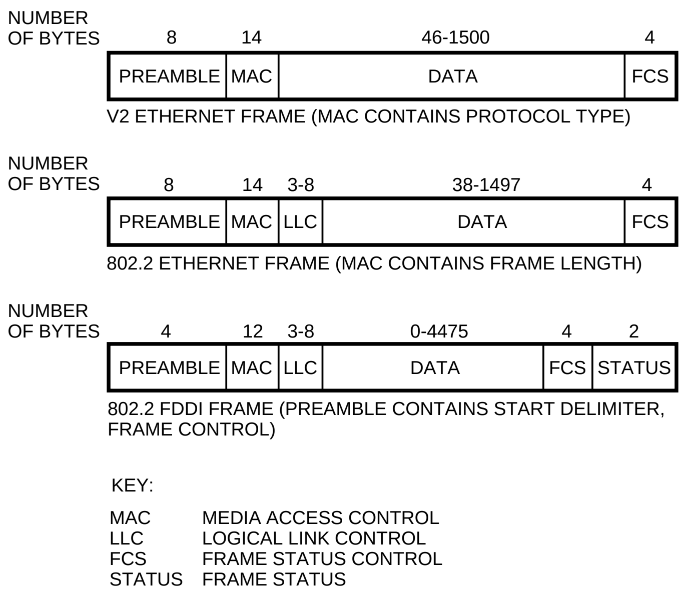


Figure 2 Frame Formats

Communication Driver

The FDDIcontroller 700 adapter connects directly with the DECstation 5000 model 200 TURBOchannel bus. The FDDIcontroller 700 is a 100-megabit (Mb)-per-second, timed token ring adapter that supports an FDDI single attachment for the DECstation 5000 model 200. The adapter provides a host interface with the following features: a packet memory interface (PMI) gate array for receive direct memory access (DMA) data transfer; a packet memory subsystem that contains one megabyte (MB) of dynamic random-access memory (DRAM) for packet store and forward; and the ability to handle FDDI ring initialization, recovery, and SMT frame processing. (SMT refers to the ANSI-specified FDDI station management.[3]) The adapter is controlled by a microprocessor and uses Digital's FDDI chip set, which includes ring memory control (RMC), media access control, and the elasticity buffer and physical link manager (ELM). The ULTRIX communication driver interfaces to the adapter's port architecture.

The port architecture defines the mechanisms to transfer FDDI frames and control or status information between the communication driver and the port. The ULTRIX communication driver interfaces to the adapter through six port registers, six port memory rings, the driver data structures, and the driver data buffers, as shown in Figure 3.

Each port register is represented by 16 bits in adapter packet memory. These registers are described in Table 1.

The adapter uses six queues, called port memory rings, to exchange data, events, and commands with the driver. These port memory rings, represented as circular lists of descriptors, are described in Table 2. Each descriptor is associated with a data buffer in adapter packet memory or in driver memory.

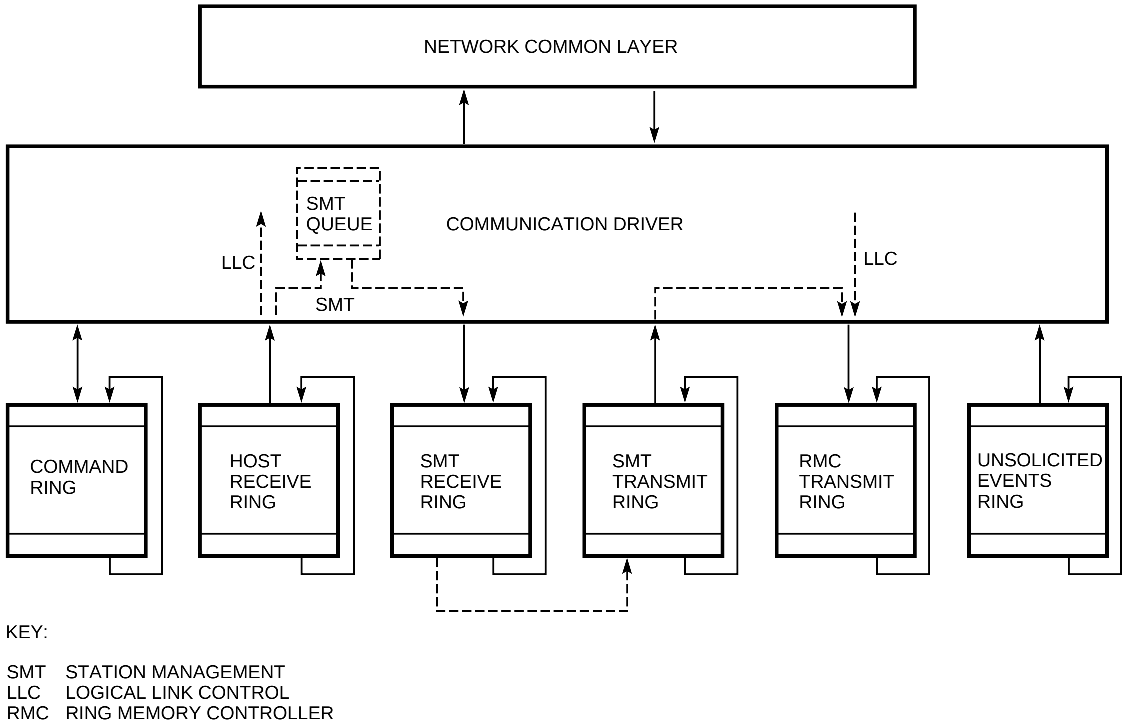


Figure 3 DEC FDDIcontroller 700 Port Architecture

Table 1
Adapter Port Register

Register Name	Written by	Purpose
Port Reset	Driver	Forces the adapter to reset
Port Control A	Driver	Controls adapter operations
Port Control B	Driver	Indicates that the driver is active
Port Interrupt Event	Adapter	Notifies the driver of important events
Port Status	Adapter	Indicates the adapter status
Port Interrupt Mask	Driver	Masks the adapter interrupt events

Table 2
Port Memory Rings

Ring Name	Purpose	Description
Host Receive Ring	Data Flow	Identifies driver data buffers to receive incoming packets
RMC Transmit Ring	Data Flow	Identifies adapter data buffers containing packets to transmit
SMT Receive Ring	Data Flow	Used by the driver to route SMT frames to the adapter for processing
SMT Transmit Ring	Data Flow	Used by the driver to route SMT frames to the adapter for processing
Command Ring	Control	Used by the port driver to initialize, configure, and monitor adapter operations
Unsolicited Events Ring	Control	Used by the adapter to notify the driver of unsolicited events

The packet data flow between the adapter and the ULTRIX communication driver is also shown in Figure 3. For incoming FDDI packets, the adapter uses a direct memory access engine to move the packets from the adapter memory to the receive data buffers of the driver. These buffers are allocated as 4-kilobyte (KB) pages in kernel memory. Each host receive ring descriptor is associated with two receive buffers to handle the maximum FDDI frame (4500 bytes). To achieve the maximum receive throughput, the driver performs packet filtering. If the incoming packet is an LLC frame, the driver processes it and calls the net_read() function. Otherwise, the driver forwards the packet to the adapter's SMT receive ring, and then the adapter queues the packet to the SMT transmit ring after processing. The driver is then notified by the adapter to move this packet from the SMT transmit queue to the RMC transmit ring for transmission. When the net_output() function requests to transmit packets, the driver copies the packets from driver memory to the RMC transit ring, signaling the adapter to transmit the packets.

The communication driver controls and requests information from the adapter by issuing commands through the command ring. These commands initialize the adapter, change the adapter state, modify and read the packet filter address table, read data link counters, and read data link status. In addition, due to the evolving state of the ANSI SMT specification, the driver function now allows the online upgrade of adapter firmware.[3] Finally, the driver supports the ability to recognize unsolicited adapter events communicated through the unsolicited events ring. When received, these events are logged and reported through the console.

ULTRIX Network Performance

Performance is a key factor in the success of Digital's workstation FDDI offering. A great effort was made to characterize the DECstation 5000 machine performance by using the earlier DECstation 3100 workstation performance results to help set realistic goals. Both the characterization and the measurements were essential to predict the performance goals. This section discusses the level of performance achieved by the DECstation 5000 model 200 running the ULTRIX operating system with FDDI support. We present details of the historical precedence for predicting FDDI performance, the evolution of the ULTRIX networking code, and the performance of applications with the FDDI system.

Historical Precedence

We expect FDDI performance to develop similarly to that of Ethernet. Early Ethernet hosts were unable to utilize more than 20 percent of the available network bandwidth because of the limited throughput capability of existing processors. The graph shown in Figure 4 illustrates the historical performance of several different processors using Ethernet adapters. Note that since 1983, network throughput has increased significantly. At some time after the middle of the decade, it was possible to reach a throughput of 10Mb per second, a rate high enough to saturate Ethernet with a single host.

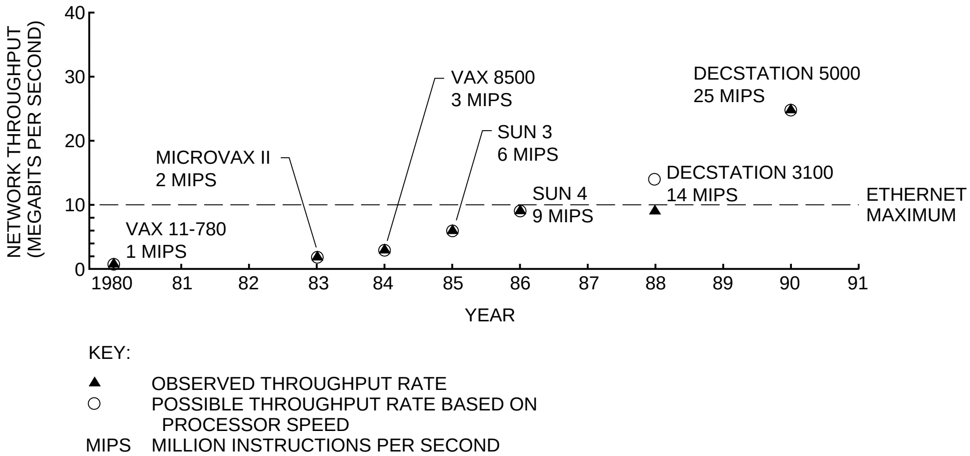


Figure 4 Historical Performance of Processors and Ethernet Adapters

Figure 4 also shows that, in nearly all cases, the achievable throughput is limited by the speed of the processor. In addition, an experimental constant of 1 mips/Mb is measured in cases where the processor is saturated. (1 mips equals one million instructions per second.) This constant means that 1 mips of processor speed is needed to generate 1Mb of network traffic. For example, the 1-mips VAX–11/780 processor is able to generate about 1Mb of network traffic. The data presented in Figure 4 shows that a processor follows the 1 mips/Mb ratio unless the adapter becomes a limiting factor, as in the case of the DECstation 3100 system. The 1 mips/Mb ratio allows us to predict that a 100-mips processor is required to saturate the FDDI system. Thus, the FDDI system satisfies the throughput requirements of available processors and allows for the growth of faster processors. Finally, if the present trend of doubling processor speed every two to three years continues, the FDDI graph will resemble the Ethernet graph of the 1980s, with the saturation of the FDDI system possible in 1996 or 1997.

Evolution of the ULTRIX Internet Code

The early implementations of ULTRIX internet networking code were based on robust BSD networking code. In 1987, the ULTRIX internet networking code was updated to incorporate performance enhancements available from a recent BSD release. Later, ULTRIX network performance was further improved to include the implementation of a dynamic buffer allocation policy to replace the inefficient static allocations. With FDDI systems, the challenge then became adapting the code to effectively use the higher network throughput.

We attacked this problem by isolating and optimizing each networking subsystem. The ULTRIX networking code is divided into three major subsystems: sockets, protocols, and drivers. Each of these subsystems can be further subdivided: sockets into the system call interface and the socket-to-protocol interface; protocols into the IP and UDP components; and drivers into the ARP, buffer management, and driver interrupt components. We used kernel profiling, a means for making run-time measurements of time spent in system-level routine calls, and known benchmarks to track progress.

Using an unreliable protocol without error recovery, such as UDP, instead of TCP with reliability and packet sequencing features, the packet-per-second (pps) rate of each component can be easily determined. Figure 5 shows the sizable packet rate measured on the DECstation 5000 model 200 for both the nonoptimized and the optimized ULTRIX network, i.e., before and after performance improvements are incorporated. Note that the pps rate of each layer reflects improvements in the layers below.

LAYERS OF THE ULTRIX OPERATING SYSTEM	RATE (PACKETS PER SECOND) NONOPTIMIZED	OPTIMIZED
SYSTEM CALL	16000	16000
SOCKET-TO-PROTOCOL	1750	2650
UDP	1700	2570
IP	700	1400
ARP	610	1230
DRIVER START	600	1200
DRIVER DONE	300	600

NOTE: THE DECSTATION 5000 PLATFORM RUNNING UNDER THE ULTRIX OPERATING SYSTEM HAS A 4096-BYTE PACKET-PER-SECOND RATE.

KEY:

UDP USER DATAGRAM PROTOCOL
IP INTERNET PROTOCOL
ARP ADDRESS RESOLUTION PROTOCOL

Figure 5 Optimized and Nonoptimized Packet Rates for ULTRIX Network Components

Packet rate values for the system call through the ARP components are determined by processor speed and code. Rates below the ARP depend on adapter speed and not processor speed. The packet rate value for the ARP is the maximum packet rate for a processor. With an optimized packet rate, Figure 5 shows a maximum rate of 1230 pps for the DECstation 5000 workstation. Since each test packet contains 4096 bytes, this rate is equivalent to 40Mb per second, which is a 40 percent FDDI bandwidth utilization.

A significant amount of work is performed at the socket-to-protocol, IP, and driver start layers because in each case, an effective copy of the data is performed. The socket layer copies data from the user into the kernel, the IP layer checksums the data, and the driver start routine copies the data to the adapter. We focused our efforts on the socket-to-protocol layer and found that considerable processor time was spent allocating kernel buffers to hold the data. Reworking this code to buffer the data more efficiently resulted in the performance change between the optimized and nonoptimized columns shown in Figure 5.

Application of the FDDI System

The greatest long-term benefit of end-node access to FDDI will probably come to those utilizing a distributed computing environment since this paradigm relies heavily on the performance of the underlying network. While Ethernet currently serves this growing set of applications well, it is expected that as the application demand for network service increases, so will the total network bandwidth and performance requirements of the participating end node.

Even today, some users of distributed network file systems (e.g., NFS) will notice a significant performance improvement as a direct result of using FDDI. This improvement is particularly evident on cached file reads and writes, where no disk access is required but the aggregate bandwidth advantage of FDDI is beneficial. However, overall NFS performance is currently limited by CPU speed and disk write capabilities; so we expect that with improvements in processor performance, disk access, and data caching, network performance at the level provided by FDDI will soon be essential.

Table 3 contains FDDI performance measurements with respect to Ethernet for various applications and transports at the application layer. The spray application is most often used to measure how an unreliable transport performs. Applications such as the BSD rcp (the remote file copy program over TCP /IP protocols), the file transfer protocol (FTP), and the test TCP (TTCP) program all measure performance of a reliable transport protocol. An NFS test is used to measure how FDDI performs as a file server. Note that, in all cases except FTP, performance improves by at least a factor of two. FTP does not take advantage of the larger buffering gained by using the FDDI system and, thus, shows no performance change over Ethernet.

Table 3
Application Performance in Relation to Ethernet

Transport	Application	Rate (Megabits) (UDP Checksum on)	RATE (Megabots) (UDP Checksum Off)	Change
TCP	rcp	18	18	2X
TCP	FTP	5	5	1X
UDP/NFS	NFS Read	20	30	2X (3X)
UDP	Spray	22	35	2.5X (3.3X)
TCP	TTCP	18	18	2X
UDP	TTCP	22	38	2.5X (4X)

Another aspect of network performance is the routing function. Using the DEC FDDIcontroller 700 adapter, the DECstation 5000 model 200 can perform FDDI-to-FDDI routing, FDDI-to-Ethernet routing, or both, for internet traffic. With a built-in Ethernet port and the ability to accept up to three additional TURBOchannel adapters, a 5000 model 200 can connect to as many as four different networks.

The performance of such a host-based router is difficult to characterize. A wide range of factors influences this performance, including the protocols routed, the efficiency of the routing algorithms, the system load, and the available data link bandwidth. Nonetheless, it is useful to consider the performance of the 5000 model 200 serving as an FDDI router because we expect this product feature to be popular. Table 4 shows the performance results for a DECstation 5000 workstation performing FDDI-to-FDDI routing and FDDI-to-Ethernet routing, both under minimal system and network load. Note that the data presented for the TCP-based applications shows that throughput is limited by the way TCP calculates its flow control window when data is destined for a remote network. All current TCP implementations have this same limitation because all nonlocal connections must have a small flow control window size of 576 bytes to avoid the saturation of intermediate gateways. Since both Ethernet and FDDI systems can transmit frames larger than this flow control window, the advantage of transmitting maximum-sized frames is lost. UDP does not have this limitation; thus, throughput numbers are only slightly lower than in the nonrouting case.

Table 4
FDDI Routing for a DECstation 5000 Workstation

Transport	Application	Ethernet-to-FDDI Rate (Megabits)	FDDI-to-FDDI Rate (megabits)
TCP	rcp	4.8	4.8
TCP	FTP	2.7	2.7
UDP/NFS	NFS Read	8.0	16.8
UDP	Spray	9.0	18.0
TCP	TTCP	6.4	6.4

Futures

This section describes some areas of research that may impact the use of the FDDI system. Included are discussions on protocol alternatives, future performance work, and how this system will facilitate new software technologies.

New Protocols

As illustrated in Figure 4, processor speed is the current bottleneck in FDDI throughput. While processor speed continues to increase, emerging protocols are making efficient use of available processing power and are yielding additional gains in network performance.

A development relevant to this discussion of protocol alternatives is the versatile message transport protocol (VMTP) research project from Stanford University.[4] VMTP demonstrates that a reliable transport is achievable with no greater overhead than existing unreliable transports. VMTP, therefore, represents an alternative to TCP that would nearly double the throughput of some remote procedure call (RPC) applications. Also, knowledge gained from the VMTP research may influence future internet or open systems interconnection (OSI) directions.

Future ULTRIX Network Performance Work

In addition to examining new protocols, performance work is continuing with our existing ULTRIX protocols. One area being studied is data copy. Currently, user data is copied twice as it traverses the internet protocol stack. One copy occurs in the socket subsystem, and the other one takes place in the device driver. Data copies account for 50 percent of CPU utilization time when large amounts of data are transferred. Eliminating one copy can yield increased performance, with savings of at least 25 percent of the total processing time.

An FDDI adapter optimized for the internet protocol stack may also provide improved performance. This decrease in processing time may result from calculating internet checksums in the adapter or from moving the complete protocol stack to the adapter. For example, researchers have proposed protocol engine chips that would off-load all protocol processing to customized chips. With such real-time protocol engines, existing processors could easily outpace current FDDI speeds.

Conclusions

Digital brought FDDI to the ULTRIX workstation to satisfy the growing network demands of its customers. The number of network-intensive applications that run on ULTRIX workstations is growing at a fast pace. Graphics and imaging applications have the potential of generating megabytes of network data. Also, multimedia applications can strain FDDI networks and are not practical using Ethernet. The best scenario for a live motion video application is the requirement of at least a 1-MB, continuous network connection, enough to easily saturate Ethernet. Even a live audio application will require a 200- to 300-KB-per-second network connection. Clearly, with applications that demand these data rates, FDDI bandwidth is necessary.

The DEC FDDIcontroller 700 adapter brings FDDI to the desktop. The adapter is well matched to the DECstation 5000 model 200, joining a 25-mips processor to a 100-Mb-per-second data link. As the next generation of LAN, FDDI extends the base for network applications by allowing those applications that run on Ethernet to run faster and by providing the bandwidth for a whole new generation of applications. FDDI is the network of the '90s, as Ethernet was the network of the '80s.

Acknowledgments

We would like to thank Kathy Wilde for her early participation on the project, Kent Ferson for general support and encouragement, and Fred Glover for his help in reviewing the document. Nolan Ring patiently edited the early draft. Julia Fey helped with the collection of performance data and with the data analysis. We would especially like to thank the staff of the *Digital Technical Journal* for their help and support.

References

- *A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks*, Internet Network Working Group, RFC 1188 (October 1990).
- *Fiber Distributed Data Interface (FDDI) - Token Ring Media Access Control (MAC)*, ANSI X3.139-1987.
- *ANSI FDDI Station Management Standard*, ANSI X3T9.5, Revision 5.1 (September 1989).
- W. Mason, "VMTP: A High Performance Transport Protocol," *Connexions*, vol. 4, no. 6 (June 1990): 2-10.

General References

A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, Internet Network Working Group, RFC 1042 (February 1988).

ANSI/IEEE Standards for Local Area Networks: Logical Link Control, ANSI/IEEE Standard 802.2-1985 (New York: The Institute of Electrical and Electronics Engineers, Inc., 1985).