

Moving Uncompressed Video Faster Than Real Time

By Don Deel, Marc Friedmann, and Howard Green

Economic methods are now available to move production and post-production data faster than real time in networked environments. Using ANSI-standard Fibre Channel serial interfaces on Silicon Graphics workstations and servers with optimized hardware and software, computer-to-disk and disk-to-disk communications have been demonstrated to transfer digital image data at a sustained throughput up to 600 Mbits/sec. Incorporating these interfaces, production and post-production facilities are achieving order-of-magnitude improvements in response time when accessing and transferring large files. Video server applications can use these interfaces for both storage access and communications for transporting up to 100 compressed streams simultaneously through a single port.

Fibre Channel is an inexpensive, expandable interface standard defined to achieve faster-than-real-time digital video data transfers among servers, workstations, disk drives, scanners, recorders, and displays. Fibre Channel combines the best attributes of a channel with those of a network through a simple technique: it transfers data between a buffer at the source device (e.g., a video server) and another buffer at the destination device (e.g., a workstation). Fibre Channel ignores the data itself and how it is formatted and simply takes what is in the sending buffer and transports it to the receiving buffer. Able to operate with multiple protocols simultaneously, Fibre Channel is an excellent interface for environments involving a wide variety of computing and video equipment.

An adapter card provides Fibre Channel connectivity for Silicon Graphics Indigo2 and Challenge M computers, which support the 64-bit version of SGI's Graphical I/O (GIO) Bus. Performance-enhancing hardware and software techniques result in the very highly sustainable throughput performance necessary for faster-than-

real-time digital video data transfers. Occupying a single bus slot and providing either one or two independent, fully functional Fibre Channel ports, the hardware for the adapter card has been optimized for high data throughput.

Specific emphasis is placed on high-performance software. Innovative software facilities called "Transporter" and "IOFS" (input/output file system) offer throughput performance improvements for transfers of large video data files. Transporter is a protocol for making very efficient memory-to-memory data transfers between com-

puters. IOFS is a file access protocol that uses Transporter to provide enhanced network access of files over Fibre Channel. These facilities will be discussed later in this paper.

The Need to Move Data Faster

With the advent of nonlinear editing, video servers, graphics special effects, and compositing, computers now play an increasingly important role in the movie and video production and post-production environment. They reside directly in the production and revenue flow, being ever more broadly used to automate and accelerate functions such as painting, retouching, rotoscoping, color correction, titling, image repair, glass painting, and wire removal. As computer capabilities advance and the number of computer-literate artists grows, project size has increased and the need for broad-based artistic collaboration has risen rapidly. Rapid and responsive communication between computer-based artists, editors, and producers is becoming a critical factor as the ramp-up in digitally processed images accelerates.

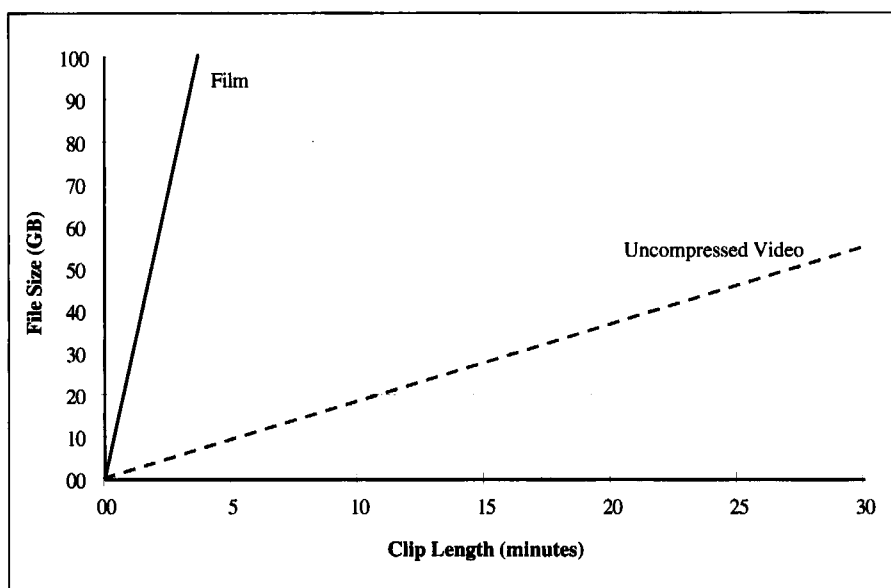


Figure 1. Digitized film and digital video file size.

Presented at the 137th SMPTE Technical Conference and World Media Expo in New Orleans (paper no. 137-21) on September 7, 1995. Don Deel, Marc Friedmann, and Howard Green are with Prisa Networks, Inc., San Diego, CA 92121. An unedited version of this paper appears in *Moving Images: Meeting the Challenges*, SMPTE 1995. Copyright © 1996 by the Society of Motion Picture and Television Engineers, Inc.

When translated to digital format, the video images used in studio applications are among the most data intensive. At 1 to 40 Mbytes/frame, even a clip of a few seconds in length rapidly

grows to more than a gigabyte file (Fig. 1). Moving files this size from central storage or to a collaborator using traditional networks can take 10 min or more, and real-time viewing of

uncompressed digital video stored on remote servers has not been possible. Production and post-production requirements are rapidly exceeding the limits of traditional network approaches and are driving the need for new network solutions. These solutions must address both hardware and software performance bottlenecks in computer-to-computer and computer-to-disk communications.

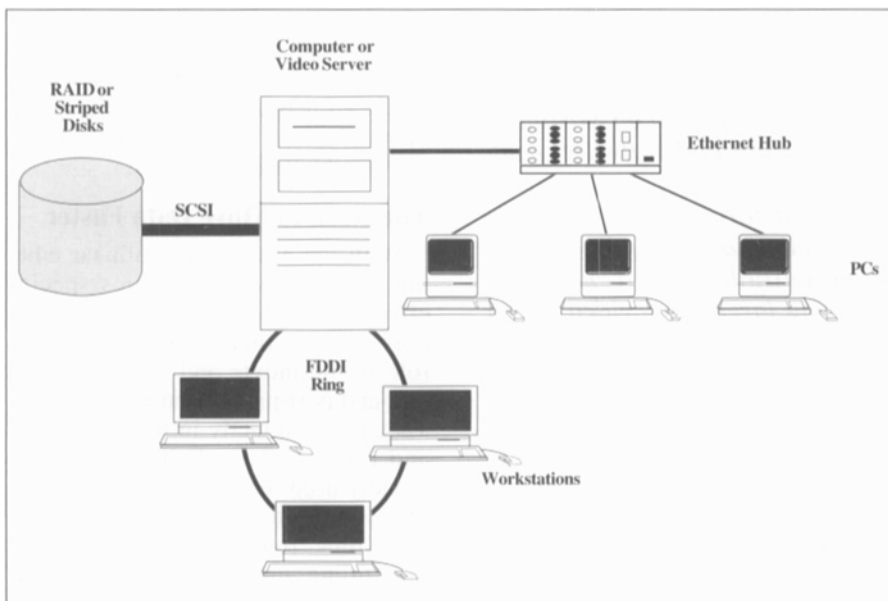


Figure 2. Client-server architecture.

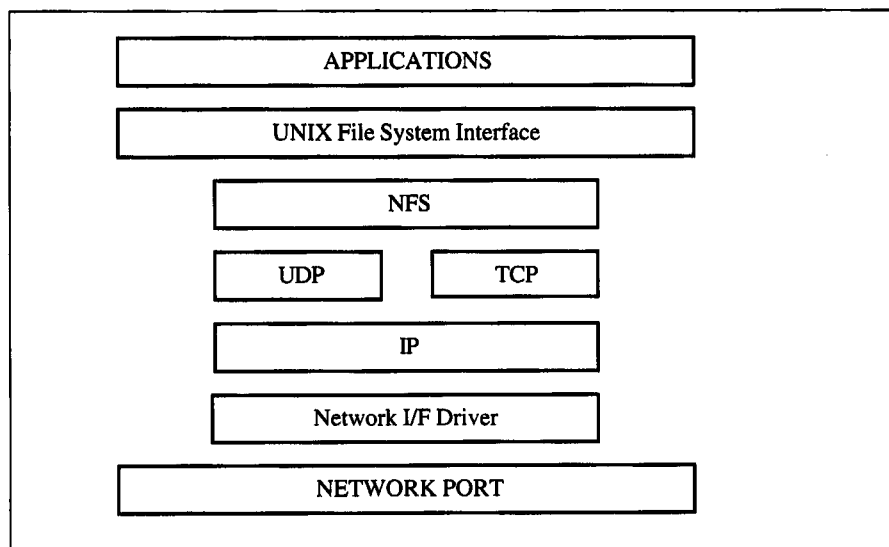


Figure 3. IP-based NFS protocol stack.

Traditional Network Approaches

Faster-than-real-time access to uncompressed digital video data stored in large disk-based files is becoming a key objective in today's studios. Using either the traditional computer or video server approach, many studios have adopted the popular client-server architecture, which centralizes storage access through a large, high-speed processor (Fig. 2). Local workstations access files through the server, which delivers them to local memory or disk. Centralized storage attached to the server can be quite large, frequently exceeding one terabyte, while local storage may be a few gigabytes. Interface from the server-to-storage must be as rapid as possible and is therefore generally in the form of striped disks or a redundant array of inexpensive drives (RAID). To maximize performance, fast/wide SCSI is used for the server to storage connection. This yields transfer rates approaching 20 Mbytes/sec, which is well below the 34 Mbytes/sec required for uncompressed real-time digital video.

File access across the network is in a packetized format using Internet protocol (IP)-based file transfer protocol (FTP) or network file system (NFS) (Fig. 3). While the use of NFS is straightforward and offers flexible file access, its small packet sizes (typically less than 8 kbytes) and the considerable management of each packet by the operating system significantly reduces data throughput rates. Since FTP procedures are disk-to-disk transfers, they are limited to the throughput rate of the slowest disk interface involved. Applications seeking performance higher than NFS or FTP may use the UNIX remote procedure call (RPC) directly, which operates as a much faster computer memory-to-

Table 1 — Network Line Rate vs. Throughput

Network Standard	Line Rate (Mbits/sec)	Throughput (Mbits/sec)
ATM	155	80
FDDI	100	30
Ethernet	10	1

memory transfer. However, even RPC-based transfers are ultimately limited by the transfer rate capability of the network.

Today's computer-based studios generally use Ethernet or FDDI for network communications. To improve throughput, some installations are beginning to experiment with ATM. Ethernet, FDDI, and ATM local area networks have been optimized as enterprise-wide networks, which are dominated by small message traffic rather than the large block transfers common with digital video files. Although specified line rates may range from 10 to 155 Mbits/sec, the combination of network and file system structures yields an observed actual throughput of 1 to 80 Mbits/sec under moderately loaded conditions (Table 1). Transfer of a gigabyte file, common in studio environments, requires two minutes to two hours at these data rates. As the number of editors and artists using workstations within a facility continues to grow, the load is rising and the need for much higher speed networks is increasing dramatically.

Faster-Than-Real-Time Interfaces

Of the new generation of interfaces, Fibre Channel offers the highest performance. Approved as an ANSI standard, Fibre Channel is a scalable interface defined to achieve high-speed data transfers among workstations,

Table 2 — Fibre Channel Features

Feature	Fibre Channel
Line rate	266, 531, or 1062.5 Mbits/sec
Data transfer rate	640-720 Mbits/sec
Frame size	2112-byte payload
Protocols	SCSI, IP, ATM, SDI, HIPPI, 802.3, 802.5
Topology	Loop, switch
Data integrity	10E-12 BER
Distance	Local and campus; up to 10 km

personal and large computers, disk drives, peripherals, and display devices. Using either loop or switch-based topology, it combines attributes of SDI-like channels with packetized computer networks over a serial interconnect capable of operating across campus-wide distances. Having the ability to support multiple protocols simultaneously, Fibre Channel is a hardware-intensive interface for environments involving a wide variety of computer, disk, and studio equipment. Key technical features of Fibre Channel are summarized in Table 2.

The Fibre Channel structure is defined as a multilayered hierarchy of functional levels. Five layers define the physical media and transmission rates, encoding scheme, framing protocol and flow control, common services, and the upper layer application interfaces (Fig. 4). FC-0, the lowest layer, specifies the physical features of the media, connectors, transmitters, and receivers, including electrical and

optical characteristics, transmission rates, and other physical elements of the standard (Table 3). Note that video coax and the 1300-nm, single-mode fiber found in broadcast facilities are incorporated in the standard. FC-1 defines the 8B/10B encoding/decoding scheme used to integrate the data with the clock information as required by serial transmission techniques. FC-2 defines the rules for framing the data to be transferred between ports, a look-ahead sliding-window flow control scheme, different mechanisms for circuit and packet-switched classes of service, the error-detection techniques, and means of managing the sequencing of data transfer. FC-3 provides common services required for advanced features, such as striping and hunt groups. FC-4 provides the seamless integration of existing standards by accommodating a number of other protocols such as SCSI, TCP/IP, FDDI, HIPPI, SDI, ATM, Ethernet, and Token Ring.

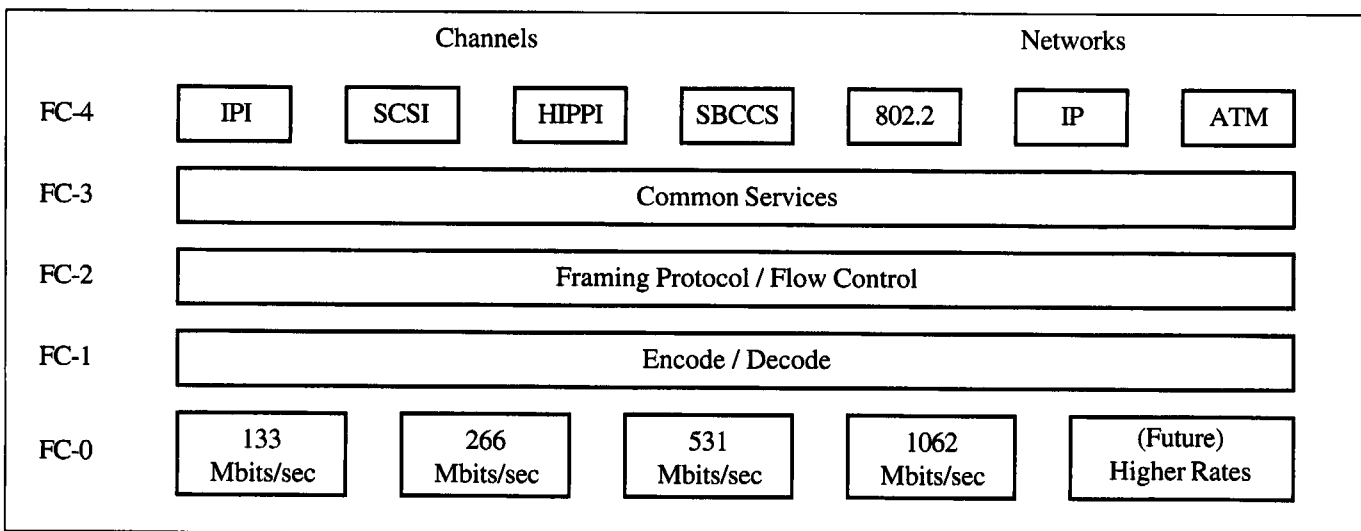


Figure 4. Fibre Channel hierarchy.

Table 3 — Fibre Channel Media

Medium	Maximum Distance	Data Rate (Mbits/sec)	Signal
Single mode fiber	10 km	266, 531, 1062	Long-wave laser
50-µm multimode fiber	2 km	266, 531, 1062	Short-wave laser
67-µm multimode fiber	1.5 km	133, 266, 531, 1062	Long-wave LED
Video coax	100 m	133, 266, 531, 1062	ECL
Miniature coax	35 m	133, 266, 531, 1062	ECL
Shielded twisted pair	100 m	133, 266	ECL

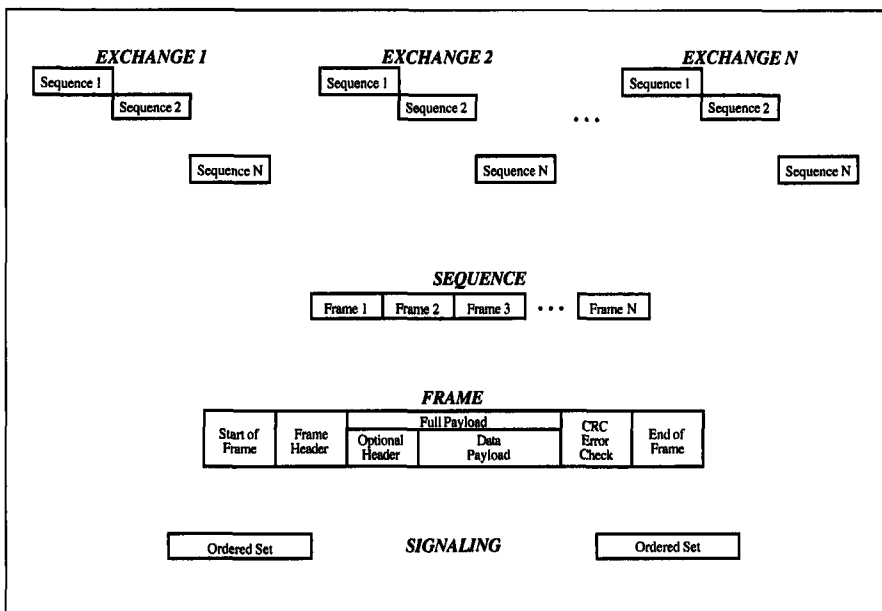


Figure 5. Fibre Channel communications structure.

Fibre Channel combines the best attributes of a channel with those of a network through a simple technique: it provides a means to transfer data between a buffer at the source device (e.g., a video server drive) and another buffer at the destination device (e.g., a workstation or frame buffer). Fibre Channel ignores the data itself and how is formatted, and simply takes what is in the sending buffer and transports it to the receiving buffer at the full bandwidth of the channel. After initial handshaking, control of the rate of data flow is handled by the receiving device indicating the amount of available memory buffer available. This low-level flow control allows Fibre Channel to avoid any data loss due to congestion. Simple error correction is handled in hardware, much like a channel. If a data transfer fails due to an error, then a retry occurs immediately without consulting system software, thus maintaining above real-time performance.

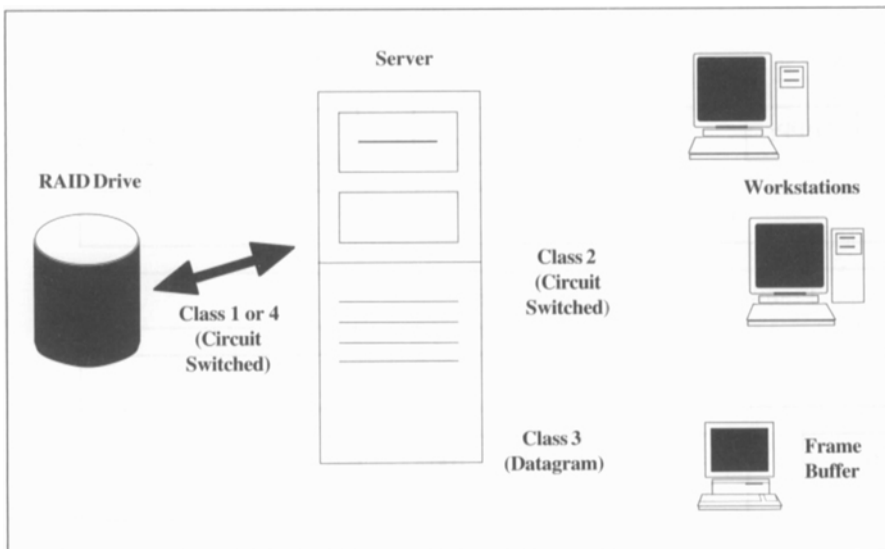


Figure 6. Fibre Channel classes of service.

Fibre Channel has four levels of communication across the links (Fig. 5). *Signaling* occurs via ordered sets, which are sets of four 10-bit characters used for such functions as start-of-frame, end-of-frame, link start-up, and special user-defined commands. A *frame* is the smallest undivided packet of data sent over the connection. Each frame consists of a start-of-frame delimiter, a frame header, an optional payload header, a data payload holding up to 2048 bytes, a 32-bit CRC, and an end-of-frame delimiter. A *sequence* is composed of one or more related frames flowing in the same direction on a link. Sequences constitute the key unit of transfer between

MOVING UNCOMPRESSED VIDEO FASTER THAN REAL TIME

ports that have negotiated available buffers. Each sequence is identified uniquely, and every frame within each sequence is individually numbered to facilitate error detection and reassembly upon arriving at its destination. An *exchange* consists of one or more non-concurrent sequences in a series of

communications between two devices. Several exchanges between the same two devices may be occurring at the same time, with each exchange in a different phase of progress (e.g., initial handshake, data transfer, close of transfer, etc.).

To accommodate on-line, off-line,

video, and computer needs, Fibre Channel defines four different classes of service (Fig. 6). *Class 1*, a circuit-switched connection, functions much in the same way as today's SDI physical channels. No other devices can share the engaged link when a *Class 1* connection has been established between two devices. *Class 2* is a connectionless, frame-switched link that provides guaranteed delivery with acknowledgment of receipt. As with traditional packet-switched networks, the path between two ports is not dedicated, allowing for shared use of the link's bandwidth. *Class 3* is a connectionless "datagram" service that allows data to be sent rapidly to multiple devices attached to the fabric, but no confirmation of receipt is given. By not having to wait for confirmation, *Class 3* service speeds up the time of transmission. However, if a single user's link is busy, the hardware will not immediately know to retransmit the data. *Class 4* offers constant available minimum bandwidth or guaranteed latency and is useful for isochronous applications such as single or multiple streams of real-time digital video.

Fibre Channel supports a variety of fabric topologies. It is a closed system

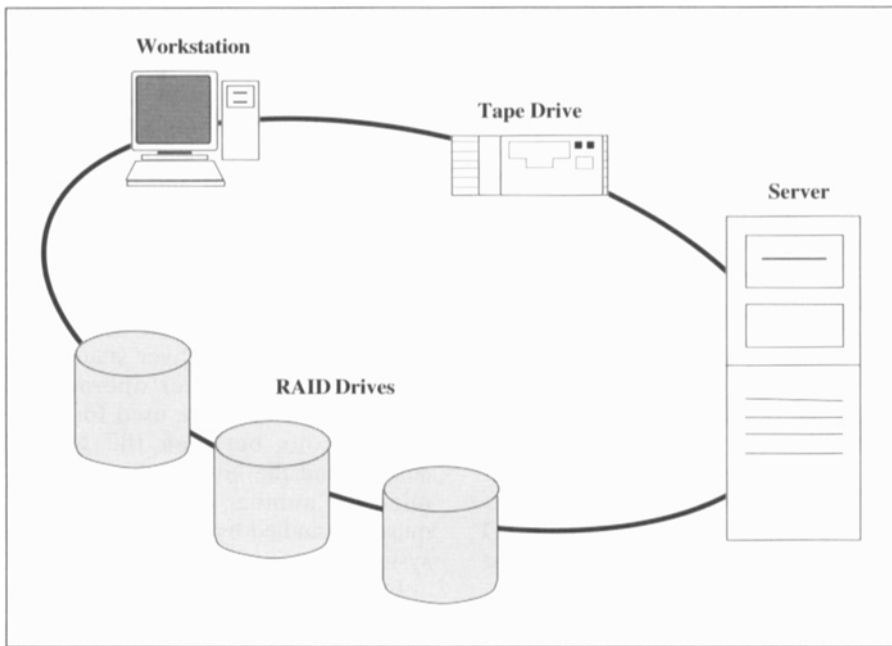


Figure 7. Fibre Channel arbitrated loop topology.

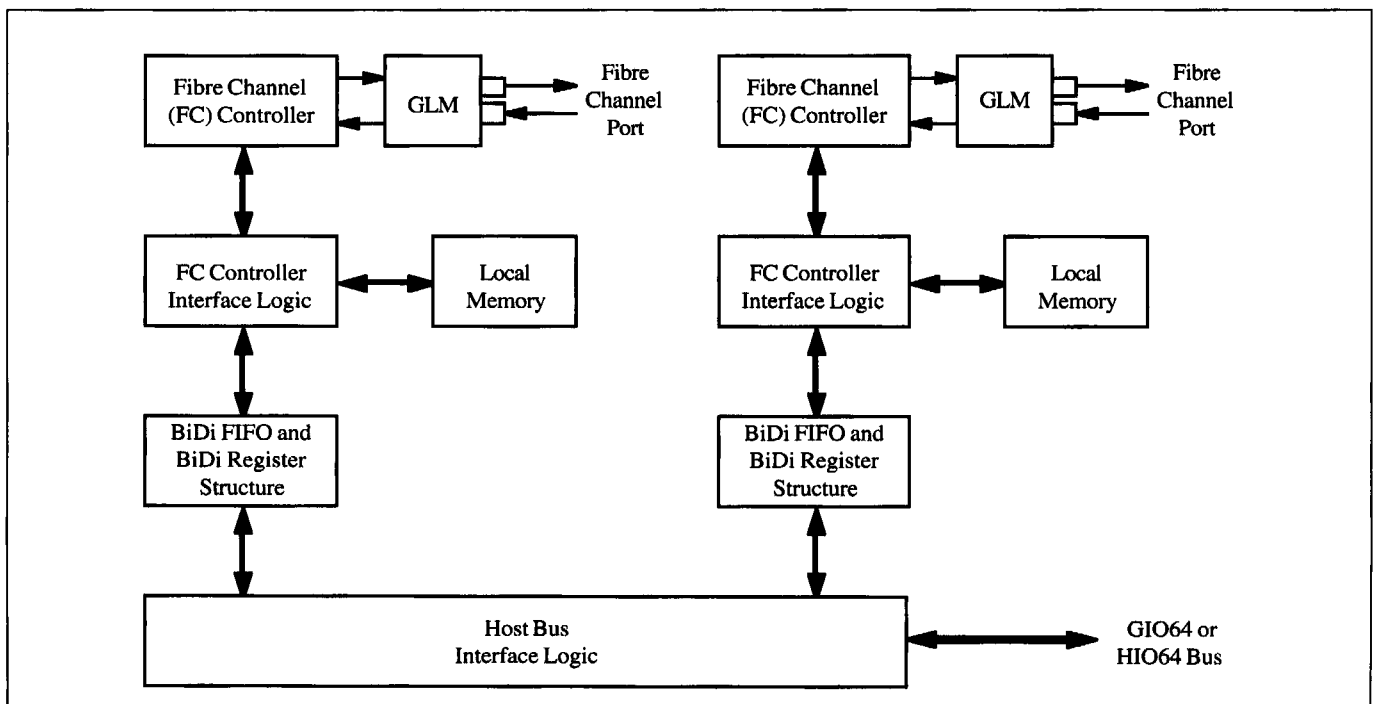


Figure 8. NetFX hardware block diagram.

that relies on ports logging in with each other and the fabric trading information on attributes in order to determine if information can be exchanged. Possible fabric topologies include point-to-point, crosspoint-switched, and arbitrated loop. The highest-performance Fibre Channel fabric is based on a point-to-point or a crosspoint-switched topology. The Fibre Channel arbitrated loop topology (Fig. 7) offers a lower-cost connection alternative that does not require the use of switches. With up to 127 ports connected on a single loop, each port can compete for a connection through a defined arbitration process. When arbitration is complete, the two successful nodes obtain access to the full bandwidth of the link. All classes of service can be supported by either the switch or arbitrated loop topologies.

Fibre Channel Adapters for the Studio Environment

The most popular servers and workstations used to work with digitized film and digital video are made by Silicon Graphics, Inc. On the higher-performing end of the spectrum, Challenge DM, L, and XL machines are used as servers, and Onyx systems are used as workstations. On the lower-cost end of the spectrum, Challenge M machines are used as servers and Indigo2 systems are used as workstations. All of these machines can communicate effectively using Fibre Channel.

To achieve the high data throughput rates made possible by Fibre Channel, it is necessary to have access to the fastest I/O bus present in the host machine. In the SGI Onyx and Challenge DM, L, and XL systems, this is the HIO bus. In the Indigo2 and Challenge M systems, this is the GIO64 bus. Both of these I/O buses are capable of providing I/O adapters with access to host memory at data rates in excess of 200 Mbytes/sec. NetFX Fibre Channel adapters are made for both GIO64 bus connections and HIO bus connections.

NetFX Fibre Channel adapters provide one or two independent, fully capable Fibre Channel ports, allowing the SGI system to communicate with disk and network traffic simultaneously. Each port supports point-to-point,

arbitrated loop, and switched fabric Fibre Channel topologies and provides Class 1, Class 2, Class 3, and Intermix classes of service. Copper or fiber-optic media can be used by the ports at the standard speeds of 266, 531, and 1062 Mbits/sec.

The NetFX Fibre Channel ports have been designed to support an average sustained data transfer rate in excess of 80 Mbytes/sec (640 Mbits/sec) for multi-megabyte transfers. Delivered performance is highly dependent upon the specific mix of large block data transfers and small message traffic, which trade off against each other. Because of per-transfer handling overheads, large block data transfers result in higher data throughput rates than small message traffic.

NetFX Hardware

The hardware for the NetFX Fibre Channel adapter includes gigabaud link modules (GLMs), Fibre Channel (FC) controllers, FC controller interface logic, local memories, BiDi FIFO and BiDi register structures, and a host bus interface logic section (Fig. 8).

GLMs perform high-speed 20:1 parallel-to-serial and 1:20 serial-to-parallel

functions and allow each port to communicate using either copper or fiber-optic media. Since different GLMs are required to support specific Fibre Channel speed and media combinations, each port on the NetFX adapter has a GLM socket that allows it to be configured independently. This permits NetFX adapters to support two different types and speeds of Fibre Channel media simultaneously.

The FC controller performs the high-speed FC-1 and FC-2 functions and assists with some of the FC-4 functions. Each FC controller connects directly to its GLM and handles the lower-level communications functions in hardware. Most of the FC controller's interactions with the host system are via direct memory access (DMA) operations, which are used to access commands, deliver status, and perform data transfer operations. Memory structures are used for communications between the NetFX adapter and the host system to minimize the number of interrupts that must be handled by the host operating system.

Each FC controller interface logic block manages requests made by its FC controller for DMA activities, and

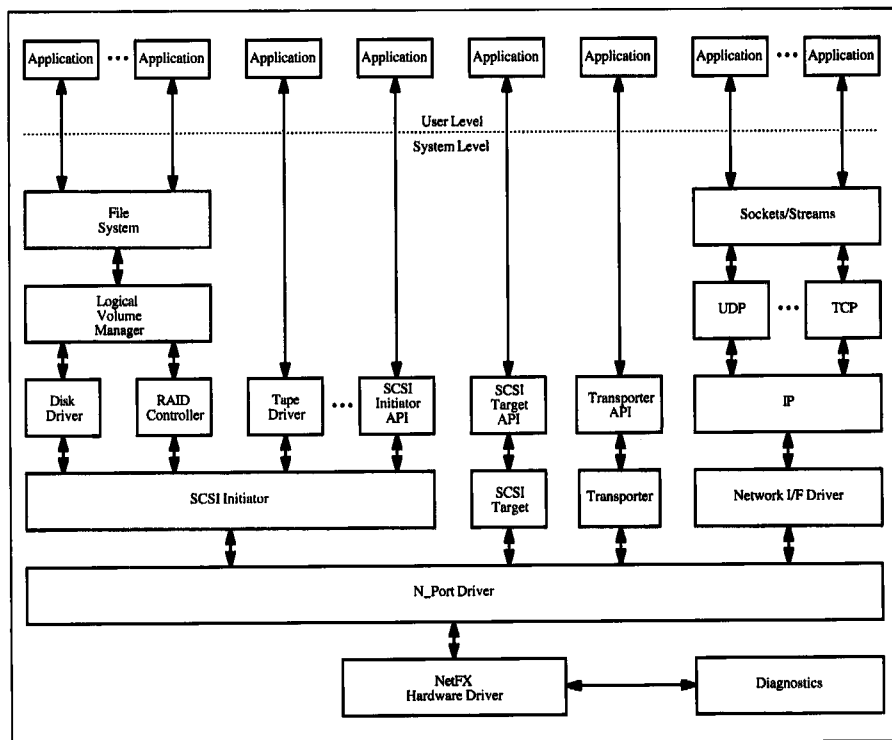


Figure 9. NetFX software hierarchy.

handles requests from the host system for accessing the port's local memory, FC controller, and GLM. DMA requests from the FC controller are directed to either the local memory or to the host memory, depending upon the referenced address. Requests from the host system are for accessing data structures in the local memory, for accessing the control and status registers in the FC controller, and for exercising control and status functions over the GLM.

Local memory is provided for each FC port to enhance overall system performance. Each port's local memory

can be accessed both by the FC controller and by the host system. By placing most of the memory data structures for commands and status in local memory, DMA operations to and from host memory are largely limited to data transfers. This enhances performance by minimizing the time it takes for the FC controller to access the memory structures it shares with the host system.

The BiDi FIFO and BiDi register structures are used by the FC controllers and the host bus interface logic as communications buffers. The BiDi FIFOs are used to "pipeline" DMA

data transfers and sustain high data throughput rates by minimizing the performance impacts introduced by bus arbitration and host memory access latency. The BiDi registers are used to facilitate host-driven accesses of the local memories and the configuration and status registers in the FC controllers.

The host bus interface logic manages communications via the connection to the host system's I/O bus and includes special provisions for improving data throughput rates by pipelining all DMA write and read operations to and from host memory. This helps to minimize the performance impact caused host memory access latency. Both master and slave modes of bus operation are supported by the host bus interface logic, which also provides diagnostic control and status functions that support testing and servicing the NetFX adapter.

NetFX Software

The NetFX Fibre Channel adapter supports a hierarchical software facility that includes low-level drivers, protocols, network management support and diagnostics (Fig. 9). NetFX software has been optimized for very highly sustained data throughput while supporting high transaction rates for small-message traffic. Implemented using standard interfaces provided by the SGI IRIX operating system, the hardware and N_Port drivers allow the host operating system software to communicate with the NetFX hardware. Protocol software enables the

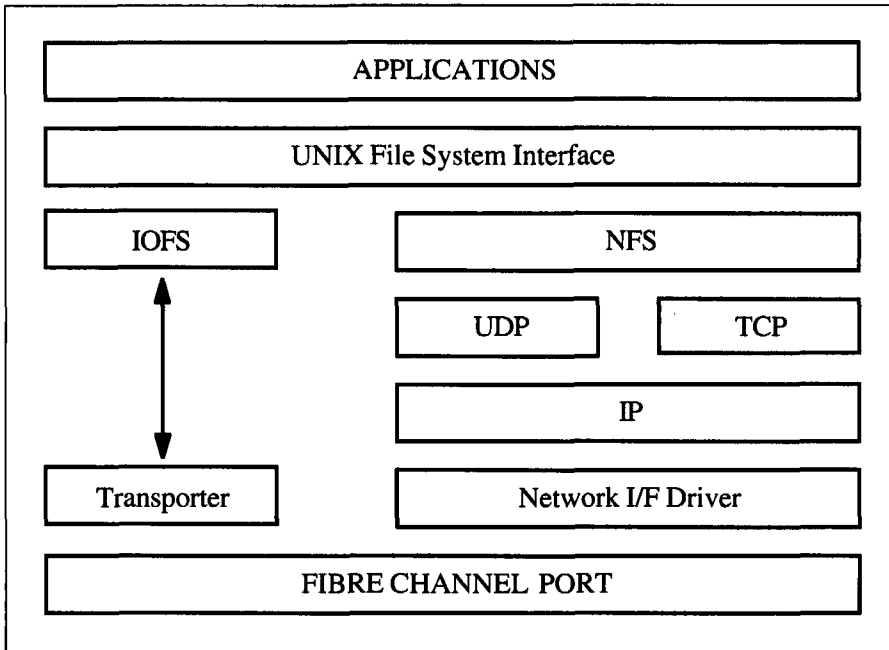


Figure 10. IOFS and NFS protocol stacks.

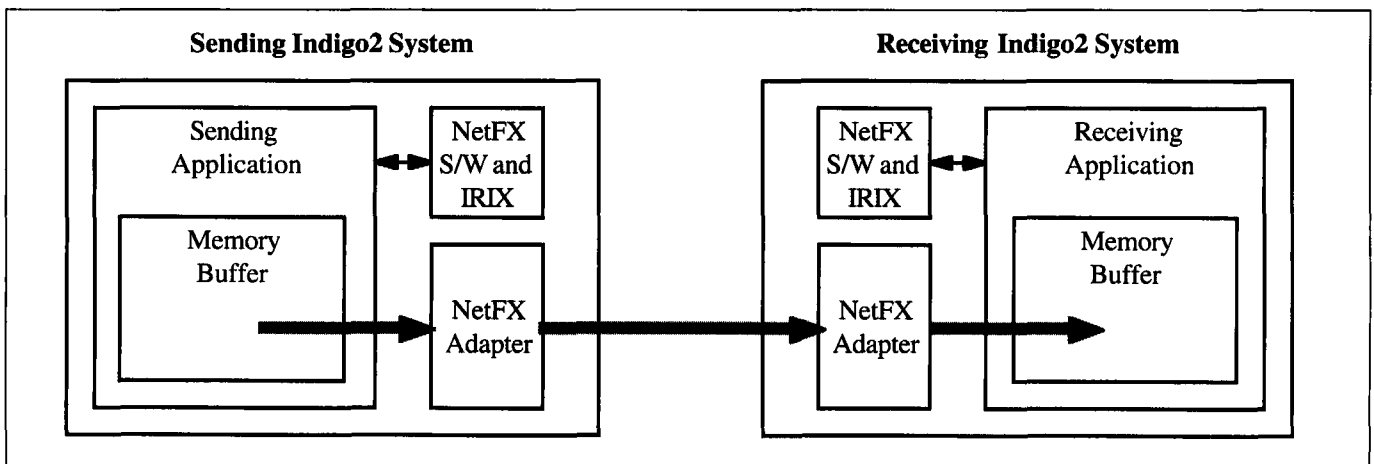


Figure 11. Throughput test setup.

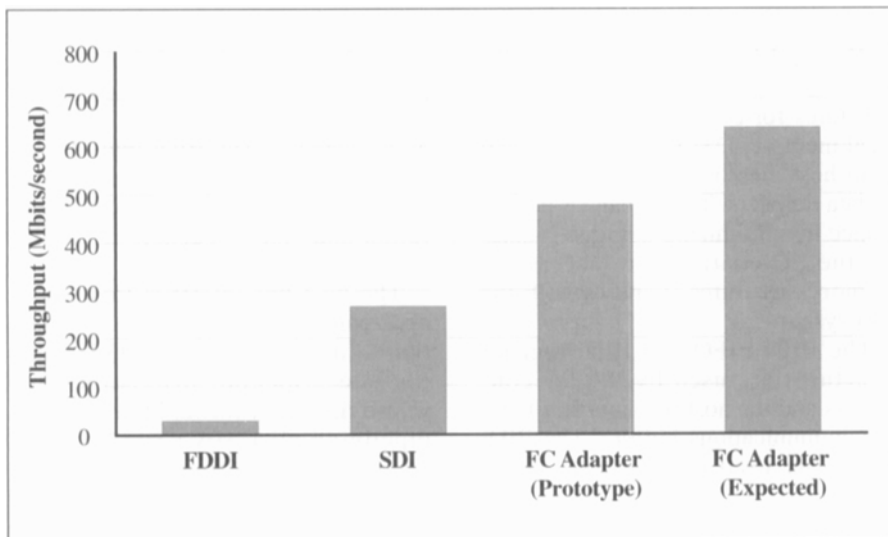


Figure 12. Relative throughputs.

operation system to communicate with attached Fibre Channel devices. Facilities for SNMP-based network managers are supported, as are diagnostics, to maintain the network and verify the correct operation of the hardware.

Hardware driver software generates commands and responds to interrupts and status from the Fibre Channel adapter hardware. N_Port driver software coordinates interactions with the hardware to implement Fibre Channel FC-2 functions including link management, fabric and N_Port login, exchange resource allocation, sequence sending and receiving, acknowledgment handling, and input stream demultiplexing. These functions perform essential services and handshake functions for FC-4 protocols.

NetFX I/O and network software facilities are made available to the operating system and to user-level applications by the FC-4 protocols above the N_Port driver. I/O-oriented FC-4 protocols for encapsulating SCSI traffic are provided for communicating with attached Fibre Channel I/O devices, such as disk drives. Both SCSI initiator and SCSI target mode operations enable software to initiate or receive SCSI commands, perform data transfers, and receive or return responses. These FC-4 protocols comply with the interoperability requirements set forth in the Fibre Channel Systems Initiative (FCSI) *SCSI Profile* document.

A network-oriented FC-4 protocol for encapsulating Internet protocol (IP) is also made available to the operating system. This software allows the operating system to send and receive IP traffic over Fibre Channel for standard data communications protocols, such as TCP, UDP, NFS, SNMP, and Telnet. This FC-4 protocol supports functionality specified in the *FCSI IP Profile* document.

NetFX Transporter and IOFS Software

A proprietary FC-4 called the "Transporter" protocol has been developed that optimizes transfers of large blocks of data between computers that are connected by Fibre Channel. This FC-4 protocol is ideal for moving digitized film and uncompressed digital video data between different computers very quickly; in the case of digital video data, it can make these transfers happen faster than real time.

Transporter treats data transfers between computer systems as memory-to-memory I/O operations, rather than as the more traditional data communications networking operations used in conventional local area networks, such as Ethernet and FDDI. It avoids many of the software overheads and inefficiencies associated with standard IP-based data communications protocols by utilizing the "native" capabilities of Fibre Channel, most of which are implemented in hardware for maximum throughput efficiency.

Fibre Channel has the ability to interconnect multiple systems, as do conventional networks. It also allows large data transfers to proceed at nearly the full speed of the communications media and incorporates hardware-level flow control and error-detection capabilities. These characteristics are typical of conventional I/O channels. Transporter takes advantage of these capabilities by implementing network-style functions in an I/O-style fashion.

The Transporter software supports a new file access protocol called "IOFS." Offering a client-server functionality very similar to NFS, IOFS uses Transporter to send large blocks of data between machines at very high data transfer rates. This allows it to bypass many of the software-level flow control and error-detection protocols used in conventional networks (Fig. 10). IOFS does not require changes to either application files or programs because it allows files to be manipulated in the same ways as NFS does. Additionally, IOFS can coexist with NFS to provide parallel access paths to the same files.

IOFS makes files accessible to remote systems in a manner similar to NFS. IOFS server software "exports," or makes available, any of the file systems that are local to the server system. IOFS client software "mounts," or locally attaches, any remote file system that it is authorized to use.

Once an IOFS client has mounted a remote file system, it accepts file access requests from its local applications programs. The UNIX system directs requests to the IOFS client based upon the name of the file being accessed. Requests are for any of the standard kinds of file-related operations, such as reading, writing, creating, renaming, deleting, or inquiring about files.

IOFS clients communicate requests to the server. IOFS servers process each request by referring it to the local file system; in the case of IRIX, this would typically be an EFS or XFS file system. The local file system is responsible for actually carrying out the request. When the local file system completes a request, the IOFS server returns a reply to the client. The reply not only provides the client — and

ultimately the application — with the requested information, but also provides confirmation that the request was correctly communicated.

IOFS is not concerned with actually managing stored information on disks or other media. For that purpose, IOFS makes use of the storage management services provided by the local file systems. The main purpose of IOFS is to provide throughput-effective access to files. Its function is essentially one of communication, and in this respect it is identical to NFS. The difference between IOFS and NFS is how data is moved from one system to another. While NFS uses traditional IP-based data communications protocols and techniques, IOFS uses Transporter, which takes advantage of Fibre Channel's hardware-based high data throughput capabilities.

NetFX Throughput

Two SGI Indigo2 workstations were connected together with prototype NetFX Fibre Channel adapters to run throughput tests. Application-level

programs were used to repeatedly send a multi-megabyte data buffer from one Indigo2's memory to the other Indigo2's memory (Fig. 11), and the actual throughput was measured.

Running this test with prototype boards, 60 Mbytes/sec (480 Mbits/sec) was measured for the end-to-end, application-to-application sustained throughput. This transfer rate is approximately twice that needed for real-time digital video.

It is important to note, however, that the prototype NetFX Fibre Channel adapters are running with an early version of the FC controller device that does not support the full speed that the hardware was designed for. The new version of the FC controller device is now becoming available, and when it is incorporated into the NetFX hardware, the resulting throughput is expected to go up to 80 Mbytes/sec (640 Mbits/sec).

The differences between the sustainable throughput rates that are possible using FDDI, SDI, and the NetFX Fibre Channel adapters are dramatic

(Fig. 12). The sustainable throughput rate shown for FDDI is a typical number for moderately loaded conditions, and the rate shown for SDI is the line rate.

Conclusion

Using Fibre Channel, it is possible to move uncompressed digital video around in the studio environment at faster-than-real-time speeds. Fibre Channel adapters have been created for the computer systems most commonly used in studios, and early throughput measurements show data transfer rates of 480 Mbits/sec between these systems, with the transfer rate expected to go up to 640 Mbits/sec soon. By incorporating these adapters, studios can achieve order-of-magnitude improvements when accessing and transferring large digital video data files. The speed of data movement will become critically important as the industry ramp-up of digitally processed images accelerates and Fibre Channel adapters provide a solution to a growing problem.

THE AUTHORS

Marc D. Friedmann is president of Prisa Networks, a supplier of high-performance networks for movie and television studios. Prior to founding the company in 1994, Friedmann worked in a variety of management positions at Applied Micro Circuits Corp. (AMCC), most recently as vice-president of network products.

He participates in several networking-related industry standards bodies and associations, including SMPTE, ANSI X3T11, and the Fibre Channel Association. Friedmann holds a bachelor's degree in Physics from UCLS and an M.B.A. from Harvard Business School.

Donald A. Deel, vice-president of product development at Prisa Networks, has been involved in the design and development of high-speed network hardware and software

interfaces for the past 15 years. For the three years prior to founding Prisa Networks, he served as system engineering manager at AMCC, where he supervised the definition and development of more than a dozen very large scale integrated (VLSI) circuits targeted at gigabit networking applications.

Prior to his tenure at AMCC, Deel was senior engineering manager in charge of product development at In-Net Corp., San Diego, Calif., a start-up supplier of fiber-distributed data interface backbone and channel extension systems. From 1984 to 1989, he headed the I/O subsystems group at Scientific Computer Systems, a start-up mini computer supplier, where he managed the I/O hardware and software development of the world's first 1.4 Gbit/sec commercial computer network.

Deel holds a B.A. in applied physics and information science and an M.S. in computer science from the University of California, San Diego.

Howard H. Green is a systems architect at Prisa Networks. For the past 15 years, he has been involved in the definition and development of high-speed computer hardware and software systems. Prior to joining Prisa in November 1994, he was an independent software contractor concentrating on very high speed computer interfaces. From 1987 to 1989, as a key team member of the I/O subsystems group at Scientific Computer Systems, Green architected and implemented the software for the world's first 1.4 Gbit/sec commercial network. Green holds a B.A. in information science from the University of California, San Diego.