SCSI, Fibre Channel, and SANs

Reynald Affeldt
affeldt@yl.is.s.u-tokyo.ac.jp

Jean-Marie Cottin
jcottin@is.s.u-tokyo.ac.jp

July 25, 2000

1 SCSI Overview

Like the more familiar IDE (Integrated Drive Electronics) and EIDE (Enhanced IDE), SCSI (Small Computer System Interface) provides a standard means for computers to exchange data among hardware devices such as disk drives and CDROM drives.

1.1 SCSI Standards

SCSI is considered a single standard but it actually comes in a variety of types. Among these are SCSI, SCSI-2, SCSI-3, Fast SCSI, Fast/Wide SCSI, Ultra SCSI, Wide Ultra SCSI, Ultra2 SCSI, Wide Ultra2 SCSI, and more. The latest specification is SCSI-3. The differences relate to the cable size and data transfer rates (see Table 1).

Standard	Cable Length (m)	Clock (MHz)	Width (b)	Speed (MBps)	Number of Devices
SCSI-1	6	5	8	5	8
Fast SCSI-2	3	10	8	10	8
Wide SCSI-2	3	5	16	10	16
Fast-Wide SCSI-2	3	10	16	20	16
Ultra SCSI-3	1.5	20	8	20	8
Ultra-Wide SCSI-3	1.5	20	16	40	16
Ultra2 SCSI-3	12	40	8	40	8
Ultra2-Wide SCSI-3	12	40	16	80	16
Ultra3 (aka Ultra160/m)	12	80	16	160	16

Table 1: SCSI Key Figures

Table 1 can be extended further for SCSI-3 provides an optional 32 bit wide data transfer mode. Other differences not reported here rely on the physical specification (single end vs. voltage differential mode).

1.2 Targets and LUNS

A SCSI device has a built-in controller that interprets commands from the host adapter. A single SCSI controller can handle up to eight devices (including the host adapter card, which counts as a device).

Through the SCSI BIOS, each device is given a distinct address or SCSI ID, ranging from 0 to 7 for a narrow (8-bit) bus or 0 to 15 for a wide (16-bit) bus. Devices that request I/O processes are called *initiators*. Devices that perform operations requested by initiators are called *targets*. Through its built-in controller, each target can accommodate up to 8 other devices; these are known as *logical units*, and they are each assigned a *logical unit number*, or LUN. Commands to the SCSI controller identify devices according to their LUNs; the controller connects to the SCSI bus and executes the commands.

1.3 Independently of CPU

The cable and the host adapter form the SCSI bus that operates independently of the rest of the computer. Tape backups, for example, from a SCSI hard disk to a SCSI tape drive can perform entirely in the background if the backup software is designed for full SCSI support.

1.4 Expandability

Each SCSI address is a target, and since each of these targets is also a SCSI device with its own controller, it can assign addresses to additional SCSI devices. In the simplest system, the host adapter card — which is the first target in the system — acts as a controller for itself and 7 (or 15) additional SCSI addresses. One of the goals of SCSI developers is to create products that can make use of this wasted capacity, primarily through a bus bridge, a two port device that connects to the main SCSI bus and creates an auxiliary bus to which additional logical units can be attached. This is the motivation of the SCSI-3 standardization.

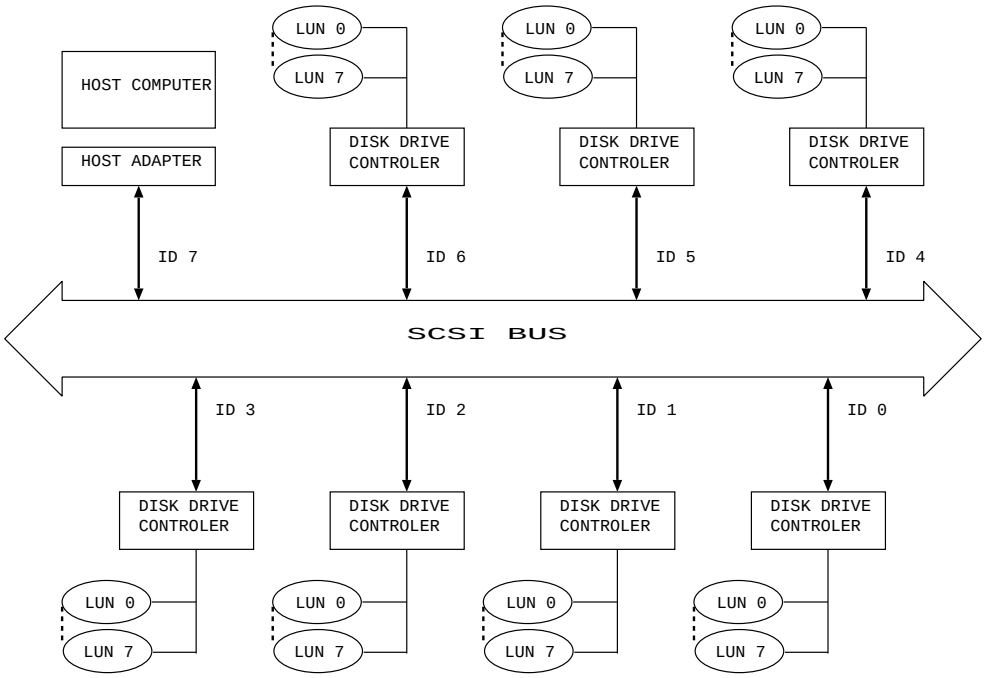

Figure 1: SCSI bus and Logical Unit Expansion

1.5 Protocol

Each SCSI unit on the bus can take control of the bus, as an initiator, by transmitting control signals (commands) aimed at a specific target. When SCSI was introduced, only one command per logical unit could be sent at a time, but SCSI-2 introduced command queuing in which 256 commands per initiator can be queued for each logical unit.

The transfer of data or control signals over the SCSI bus takes place in eight phases:

- BusFree: The initiator determines that the bus isn't under the control of another SCSI device.
- Arbitration: The initiator sends a control signal to set the bus into a busy state and sends its own ID number out to the bus saying it will take control. If another device sends out the same signal at the same time, the arbitrator gives the device with the highest ID number access to the bus.
- Selection: The initiating device specifies a target unit and sends that unit one or more commands, like read or write. When the target receives the command, it sends its ID number onto the bus so that both devices are active. Since only two devices can be active at any given

time, the bus becomes unavailable to any other unit. The target recognizes the initiator by its ID number and the controller stores this information in case of a lost connection between the two. A lost connection can result from an error, but in the case of reading and writing data, the lost connection can also be intended (Section 4, Page 7). During drive access (which takes some time), the target releases the SCSI bus so the other transactions can occur. Once the target has completed the command, it enters the selection phase again during which it recalls the initiatorś ID number and essentially repeats bus free, arbitration, and selection phases to re-establish contact.

- Four final stages: command, data, message, status: During these stages, the target receives commands from the initiator, and the two exchange data, and if necessary, messages concerning the connection and status information for each other. Once the purpose of the connection has been fulfilled, the initiator sends a reset control signal and the bus is returned to a bus free state, ready for other transactions.

2 Physical layer

2.1 signal overview

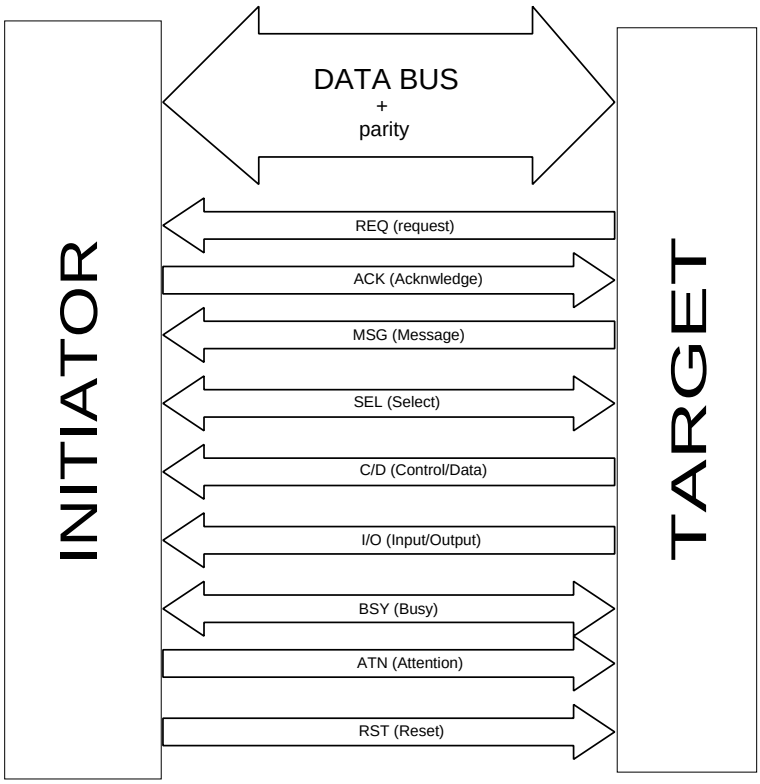

Figure 2: SCSI Bus Signals

The bus signals are represented on Figure 2. They can be grouped as follows:

- D0-D8: Data lines.
- DP: Parity line.

9 control lines:

REQ, ACK: Hand-shake data transfer mode
MSG, SEL, C/D, I/O: Phase indicator
BSY, ATN, RST: Extra-control signals (asynchronous, event, notifications, etc.)

- BSY (Busy) This line can be asserted either by the target and the Initiator. During the arbitration phase, several devices may race for the mastery of the bus. BSY line is asserted by a device to indicate it wants to take the bus. Physically, BSY line is a wired OR. During the selection phase, the initiator chooses the target and activates BSY. As an answer, the target reply by using also the BSY signal.
- ATN (Attention) As noticed previously, the exchange between the Target and the Initiator are controlled by the target. But if the Initiator wants to tell something to the Target, it asserts the ATN line, which forces the Target to change the bus state to 'Message_out'.
- RST is the reset. When asserted, it forces the bus into the 'Bus_Free' state.
- REQ/ACK is the pair of signals which control the information flow on the bus. The Target request information using REQ (Request) and the Initiator replies asserting ACK (Acknowledge). Please, note that this situation is a particularity of SCSI, and is quite paradoxical. The Target has the Initiative, whereas the so called Initiator is passive!
- MSG (Message) indicates a message passing. In the normal scheme, a message is always issued by the target and destinated to the Initiator. When the Initiator wants to send a message, it uses the ATN signal.
- SEL (Selection) indicates the SELECTION and RE-SELECTION phases.
- C/D stands for Control/Data.
- I/O for Input/Output.

MSG	C/D	I/O	SEL	PHASE	TRANSFERT DIRECTION
*	*	0	1	Selection	$I \rightarrow T$
	*	1	1	Re-selection	$I \leftarrow T$
0	0	0	0	Data Out	$I \rightarrow T$
0	0	1	0	Data In	$I \leftarrow T$
0	1	0	0	Command	$I \rightarrow T$
0	1	1	0	Status	$I \leftarrow T$
1	0	0	0	undefined	
1	0	1	0	undefined	
1	1	0	0	Message Out	$I \rightarrow T$
1	1	1	0	Message In	$I \leftarrow T$

Table 2: Phase Indicators

2.2 Asynchronous Transfer Mode

For the Initiator to write a byte from the target, the following procedure is applied:

- The target asserts REQ which, during a Data_Out phase, indicates it is ready to receive a byte from the Initiator.
- The Initiator set the bits D0-D7 (or D0-D7 and D8-D16 for WIDE SCSI). When the data are ready, the Initaitor asit asserts ACK.
- The target reads the data on the bus and asserts REQ.
- Then the Initiator negates ACK.

The scheme is pretty much the same in the case of a 'read':

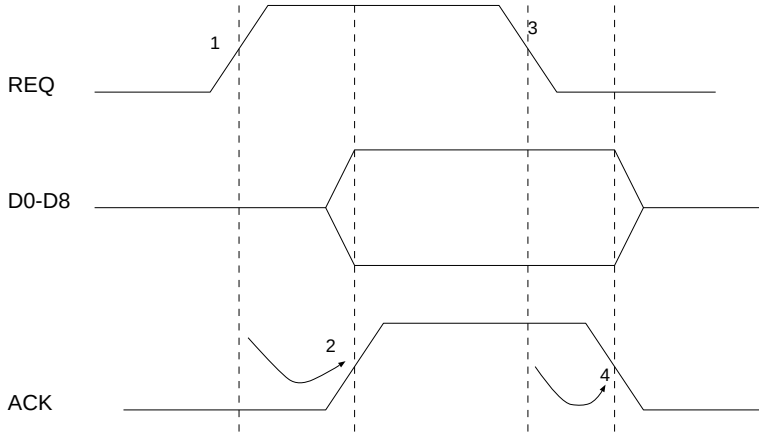

Figure 3: Asynchronous $I \rightarrow T$ Data Transfer (Data_Out Phase)

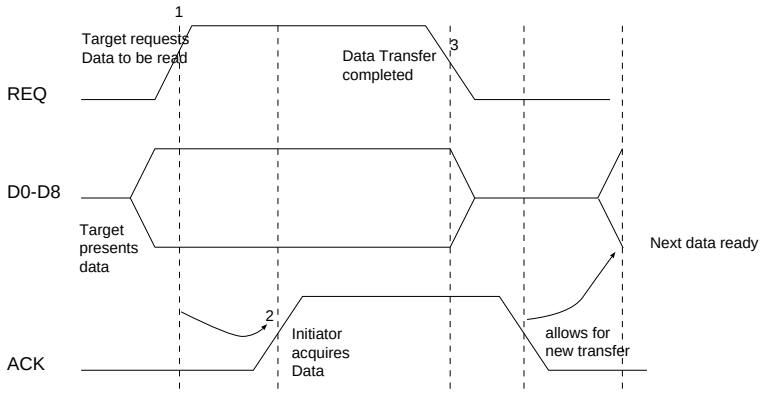

Figure 4: Asynchronous $T \rightarrow I$ Data Transfer (Data_In Phase)

2.3 Wide vs Narrow SCSI

SCSI-2 allows for a 16 bit large data bus (WIDE SCSI-2). This is achieved using a two cables: the basic cable A and a second cable, called B. Cable A and B both carry one byte large data. Each has its hand-shake signals REQ/ACK for A and REQB/ACKB for cable B.

Using two cables enables simple SCSI devices to work together with SCSI-2 compliant devices as showed on figure 5 below. The data exchange between a WIDE SCSI device and a basic SCSI device is negotiated using messages.

3 Protocol in detail

Bus free

This is the state where no device is using the bus. One returns to that state when both SEL and BSY are negated.

Arbitration

Coming from a bus free state, a device asserts BSY to indicate its intension of using the bus.

Within $1.8\mu s$ (Bus Set Delay), other devices can also assert BSY to participate to the race. Each device then sets to '1' the data line corresponding to its priority number (ID number): Device with ID 0 will assert D_0, device with ID i will assert D_i.

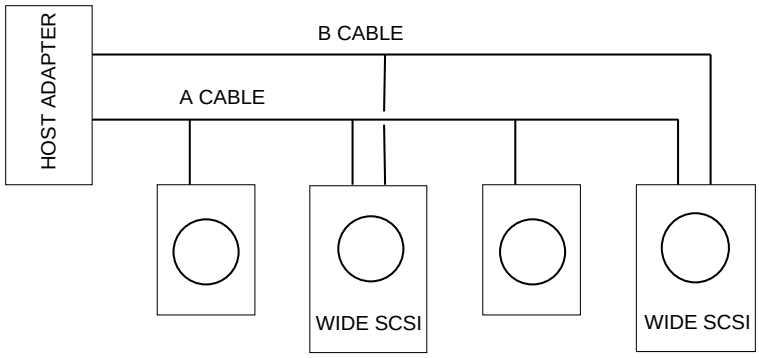

Figure 5: Connecting SCSI and WIDE-SCSI-2 Devices Using A and B Cables

During $0.6\mu s$, priorities are compared. The device with highest priority wins and asserts SEL. The selection phase has started.

Selection

The Initiator:

1. waits $1.2\mu s$ for other devices to leave the bus (BSY negated).
2. negates line I/O (Input state).
3. put its own ID and the ID of the Target to whom it wants to talk.
4. release BSY.
5. waits (within $200\mu s$) for the addressed device (the Target) to reply by asserting BSY.
6. reply to the activation of BSY by releasing SEL.

The bus enters the Information Transfer Phase.

Information transfer Phase

It includes 4 sub-phases: message, Command, Status and Data.

Message Phase

It is indicated by the MSG C/D signals asserted by the Target (see Figure 2).

Messages can be exchanged from Initiator to Target or from Target to Initiator according to I/O signal.

A message is a one byte word giving information (like 'Command completed without error') or commands to the bus, eventually followed by additional bytes.

For example:

00h Command Complete

04h Disconnect

02h Save Data Pointer (see Paragraph 4, Page 7).

80h → FFh Identify
It is necessarily sent by the Initiator to the Target after the selection phase. It establishes the 'Nexus' (logical connection) Initiator-Target-Logical_Unit.

7	6	5	4	3	2 1 0
1	Disc.Priv.	LUN-TAR	0	0	LUN/TRN

LUN-TAR: 0 if LUN/TRN is a Logical Unit Number or 1 if LUN/TRN is a Target Routine Number.

Disc.Priv. authorizes Disconnection to the target.

Rq: at the re-selection phase, it is sent by the target.

01h + 03h + 01h + m h + x h	synchronous data transfer request

where m is the offset between REQ and ACK and x is the transfer rate.

Command Queuing messages

A Target can memorize commands in a command queue, and execute them in an optimal order (think of the accesses to a disk and the elevator algorithm to optimize the head motion). But the host computer might want requests to be executed in a different order, optimal for the CPU for instance. Message enable to act in this way.

Command

The target enters this phase to ask the Initiator for commands.

- MSG, I/O are negated.
- C/D is asserted.
- REQ is asserted to express the traget is waiting for the first byte of the command. That first byte identifies the group (ie the length of the 'CDB' or Command Descriptor Bloc) and the command code itself.
- The Initiator sends the command (CDB) using Hand-shake mode (REQ/ACK).

Data

MSG and C/D are negated. The transfer is performed using the REQ/ACK pair of signals. In SCSI-2, the transfer in this phase (and only here) can be performed synchronously, that is without waiting for the acknowledgement ACK of the Initiator.

Status

The Target negates MSG and activates C/D and I/O, then sends one byte as a satus information. The satus condition is an information non how the last command has been executed: Good (command completed normaly), Busy, Queue full, Reservation Conflict, etc.

4 Context Switching and Queuing

A scsi process is internally represented in the Initiator as 3 memory spaces: the command descriptor block, the process data, the status.

As evoked in the overview (Page 3), the target has the possibility to interrupt a current processing of a command, either to perform other commands or to release the bus. It sends to the Initiator the 'Save Data Pointer message. The Initiator will save internally the current position in the data memory space, and eventully save the command and status pointer to switch to an other process or waits for reselection from a previous target.

5 Limitations and Enhancements

5.1 Critical time and limitations on the length

- Arbitration phase.
- Selection phase.
- The information transfer is asynchronous. Handshake mode (ACK). (however, SCSI-2 allows synchronous transfer mode).

5.2 SCSI-2 SCSI-3

To improve performances, SCSI-2 and SCSI-3 double the data bus width, passing from 8 to 16 bits in SCSI-2 and from 16 to 32 in SCSI-3. The data transfer rate is therefore multiply by two or four compared to the basic SCSI. WIde data transfert is negociated between target and initiator with the WDTR (Wide Data Transfer Request) message.

Further enhancements are:

5.2.1 SCSI-2 and the FAST SCSI

SCSI-2 introduces as an option, the capacity to transfer data synchronously on the bus. The clock speed and the delays are negotiated between the Initiator and the Target after the selection phase.

5.2.2 SCSI-3 and the Storage Area Network

The SCSI-3 specification greatly enhance SCSI by designing:

at the physical level: parallel interface (SCSI-2 compatible) and Serial Interface.

at the protocol level: interlocked protocol (SCSI-2 compatible) and packetized protocol.

This 'serialized' SCSI corresponds to the need for long distance (Fiber Channel) and high-speed data transfer (Fire-Wire)

From here on, fiber channel is presented and exemplified as a Storage Area Network.

6 Small Computer Systems Interface (SCSI)

6.1 SCSI and IDE

Like the more familiar IDE (Integrated Drive Electronics) and EIDE (Enhanced IDE), SCSI (pronounced "scuzzy") provides a standard means for computers to exchange data among hardware devices such as disk drives and CDROM drives.

SCSI's most obvious strength is the number of devices it can control. Where IDE interfaces were restricted to two disk drives and today's EIDE interfaces are restricted to four devices (including hard disks and CDROM drives), a single SCSI controller can handle up to eight devices (including the host adapter card, which counts as a device).

6.2 SCSI Flavors

7 Fibre Channel (FC)

7.1 Overview

Fibre Channel is a protocol for communication between computer-related devices. It allows many well-known existing channel and network protocols to run over the same physical media with a very high bandwidth and with a connectivity of several kilometers.

A *channel* is a mechanism for transmitting data, typically between a computer and its devices. Once a channel is set up, it allows for a very fast communication. It is usually a simple protocol that operates over small distances and within which errors are hardware-checked. For instance, SCSI is such a channel protocol.

A *network* is a mechanism for transmitting data between many entities. As it can automatically adjust itself to route data from one device to another, it is more flexible but also slower than a channel, mainly because of a heavier protocol. For instance, Ethernet is such a network protocol.

Fibre Channel combines both a channel and a network protocol.

Main documents	ANSI X3.230-1994, *Fibre Channel — Physical and Signaling Interface* ANSI X3.272-1996, *Fibre Channel — Arbitrated Loop* ANSI X3.269-1996, *SCSI-FCP — Fiber Channel Protocol for SCSI*
Data Rates	133 Mbps (eighth speed) 266 Mbps (quarter speed) 531 Mbps (half speed) 1.063 Gbps (full speed) 2.126 Gbps (double speed) 4.252 Gbps (quadruple speed)
Cable Length	up to 30 m over copper up to 10 km over fiber
Devices	up to 127 per arbitrated loop

Table 3: Fibre Channel Key Figures

7.2 Fibre Channel Layers

The Open Systems Interconnect (OSI) model is the International Standards Organization (ISO) organization for an ideal protocol. It consists of seven layers:

1. *Physical.* Cable, connector, and other signaling specifications.
2. *Data Link.* Is broken into two sublayers:
   (a) *Medium Access Control (MAC).* That gets packets on and off the wire.
   (b) *Logical Link Control (LLC).* That does error detection and correction.
3. *Network.* Deals with getting data packets from one device to another in a reasonable period of time. It does not care about the previous layers. In other words, it performs routing and flow control.
4. *Transport.* Provides a logical data stream that hides the previous layers.
5. *Session.* Controls communications between applications. It checks for out-of-sequence packets and handles dialogs.
6. *Presentation.* Deals with differences in data representation such as line endings (CR under UNIX become CRLF under MS-DOS) or character sets (EBCIDIC to ASCII).
7. *Applications.* Handles the user software applications such as file transfer, terminal emulation, and interprocess communication.

The Fibre Channel protocol doesn't follow the OSI model. It is in fact broken into five layers:

1. *FC-0.* Receiver/transmitter, cable plants, media types, and signaling specifications.
2. *FC-1.* IBM patended 8B/10B character encoding. Every transmitted byte of data is converted into a 10-bit value. This improves in particular the detection of errors.
3. *FC-2.* Frame format (see Figure 6), sequence management, exchange management, flow control, classes of service (see Section 7.4), login/logout, topologies (see Section 7.6), segmentation/reassembly (two methods are available for segmentation/reassembly: Sequence Count and Relative Offset).
4. *FC-3.* Set of services for devices having more than one port.It allows *striping* (one piece of information is sent through many ports to increase the bandwidth), multicast, and *hunt groups* (groups of nodes that ensure availability of a free node inside the group).
5. *FC-4.* Upper Layer Protocols (ULPs) mapping onto the Fiber Channel Protocol: Small Computer Interface (SCSI), Internet Protocol (IP), High Performance Parallel Interface (HIPPI), Asynchronous Transfer Mode — Adaption Layer 5 (ATM-AAL5), Intelligent Peripheral Interface — 3 (IPI-3), Single Byte Command Code Sets (SBCCS), and other ULPs.

The layers from FC-0 to FC-2 are sometimes referred to as the FC-PH layer.

Comparison between Fibre Channel layers and the OSI model FC-0 and FC-1 define the Physical layer and the LLC sublayer of the Data Link layer of the OSI model. FC-2 defines the MAC sublayer of the Data Link layer of the OSI model. FC-3 is not a layer. It is barely defined. FC-4 defines how other protocols are mapped onto the Fibre Channel protocol.

7.3 Running ULPs

IP and SCSI can be considered as application level payload. They are Information Units at the FC-4 level that lower levels will map to a real FC Sequence.

7.3.1 SCSI over Fibre Channel

SCSI is the only widely used protocol over Fibre Channel.

The Fibre Channel Protocol for SCSI defines a mapping that uses the services defined by Fibre Channel to transmit SCSI informations between a SCSI initiator and a SCSI target. The transmission is performed across the Fibre Channel network using only the standard Fibre Channel services.

An application client begins a Fibre Channel I/O operation when it provides the Fibre Channel protocol with a request for a command service. The request contains all the information necessary for the execution of one SCSI command so that the Fibre Channel port acts as a SCSI initiator. The Fibre Channel protocol then performs actions using FC-PH services to perform the SCSI command. When the target as completed the interpretation of the command, it sends back an acknowledgment, and the transfer may take place. After all the data has been transfered, a frame is sent back that contains all the SCSI status information.

The Fibre Channel takes full advantage of the multiplexing capabilities of the Class of service 2 proposed.

7.3.2 IP over Fibre Channel

IP over Fibre Channel is not yet widely used.

The purpose of this ULP is to specify a way of encapsulating IP (Internet Protocol) over Fibre Channel and to describe a mechanism for IP addresses to Fibre Channel addresses resolution. How do we associate Fibre Channel port identifiers to MAC (Medium Access Control) addresses? This problem has a strong connection with the ARP (Address Resolution Protocol) protocol. More details may be found in [6].

Thanks to such a method, servers may be able to communicate to each other using IP over Fibre Channel, making this technique useful for clustering.

7.4 The Transmission Hierarchy

Every transmitted byte is converted into a *Transmission Character* by the FC-1 layer.

All information is transmitted in groups of four Transmission Characters called *Transmission words.*

Some Transmission Words are special *Ordered Sets.* Examples of Ordered Sets are the frame delimiters Start of Frame and End of Frame (See Figure 6). Loop Initialization Primitive (LIP) is another important Ordered Set. Transmission Words are transmitted within frames whose structure is shown below.

A *Frame* consists of 36 bytes of overhead and up to 2112 bytes of payload. The total size must anyway be an even multiple of four bytes so that only Transmission Words are sent. Pad bytes may be appended to the end of the payload in order to achieve this requirement. The Cyclic Redundancy Check (CRC) uses the same 32 bits polynomial used in Fiber Distributed Data Interface (FDDI) and is specified in ANSI X3.139, *Fiber Distributed Data Interface.* The frame header is 24 bytes long and contains information about the payload. Three kinds of optional headers are defined: Network_Header (16 bytes long), Association_header (32 bytes long), and Device_header (up to 64 bytes long). A frame header also carries a soure Port identifier and a destination Port identifier.

One may compare the frame structure with an Ethernet packet (Figure 7). An Fibre Channel is larger than an Ethernet packet, and far larger than an ATM packet.

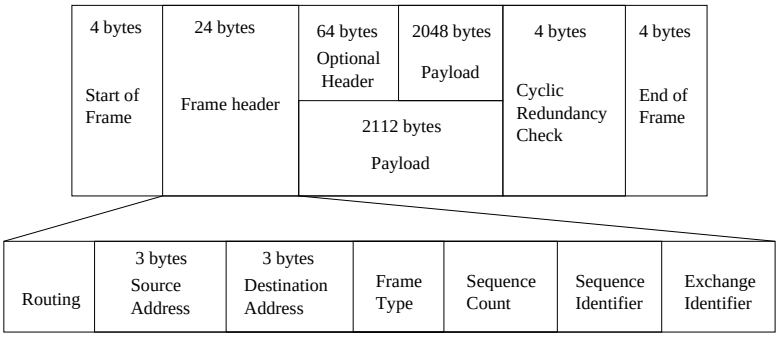

Figure 6: A Fibre Channel Frame

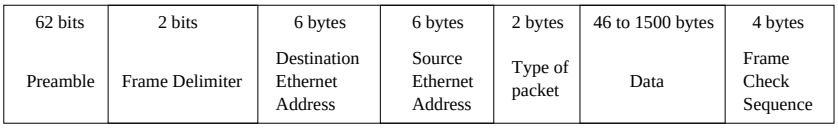

Figure 7: An Ethernet Packet

In between frames, a *Primitive Signal* (a single Ordered Set) or a *Primitive Sequence* (a stream of the same Ordered Set) can be transmitted. For an example of how those Ordered Sets are used, see in 7.6 how a Loop is initialized.

A *Sequence* is a series of related Frames. All frames must be part of a sequence. Frames from a Sequence are recognized by their SEQ_ID field. They are distinguished within the Sequence by their SEQ_CNT field that orders them.

An *Exchange* is a series of Sequences between two ports. The Sequences may be in either direction. All Sequences must be part of an Exchange. The initiator assigns the OX_ID field while the recipient assigns the RX_ID field.

The *Flow Control* guarantees that a device can transmit Frames to another device only when the other device is ready to accept them. Before the devices can send data to each other, they must login to each other and exchange informations about how many frames they can receive at a time.

The *Classes of Service* are communication strategies.

1. *Class 1.* A private connection is established between two devices that may use the full bandwidth. There, the Fibre Channel network merely looks like a channel protocol. All other devices in between may simply repeat the data. It looks like conventional telephone networks.

2. *Class 2.* There are notifications of delivery and non-delivery of Frames instead of a dedicated connection. A device can transmit frames to and receive frames from more than one port. The bandwidth of the links is shared with other network traffics: we say it is *multiplexed*. It is more like packet switching.

3. *Class 3.* It is a connectionless class of service without acknowledgements and with optional multicasting.

4. *Class 4.* In a Pure Fabric topology, fractional bandwidth can be allocated along a path. A port set up a virtual circuit by sending a request to the Fabric indicating the remote port. The connection consists of two unidirectional virtual circuit that may be of different bandwidth between the two ports.

5. *Class 5.* Undefined.

6. *Class 6.* It provides multicast service through a Fabric.

7.5 Media

Fibre Channel can run over copper and fiber media.

The most common copper medium is the shielded twisted pair using a DB-9 connector. Speeds over 100 MBs require fiber media.

Long distances can be achieved thanks to fiber. Fiber media can be multi-mode or single-mode fiber using the SC connector.

Fiber Channel offers a wide variety of speeds. After factoring 8B/10B encoding and frame headers, the transmission speed is normally 1063 Mbs. This is said to be the full speed. Eighth from quadruple speeds also exists. See Table 3.

7.6 Topologies

Fibre Channel devices are identified by Nodes and their Ports. A Node is a collection of Ports identified by a unique non-volatile 64 bits name. Each Port is identified by a unique volatile Port identifier.

7.6.1 Point-to-point

Two devices are connected directly together. The transmit fiber of one device goes to the receiver fiber of the other device, and vice-versa. We say that Fibre Channel is *full-duplex* because is such a connection the two devices can enjoy the full bandwidth. See Figure 8.

7.6.2 Arbitrated Loop

The media is shared among the devices. See Figure 8.

Before it is usable, it must be initialized so that each port obtains an *Arbitrated Loop Physical Address*. Only 127 values are valid. If more than 127 devices ask for an address, some will not participate in arbitration. *Loop initialization* consists of three steps:

1. The *Loop Initialization Primitive* (LIP) Sequence begins the process. Transmission from a device starts after it is turned on. When *Loop Failure* is detected, the Sequence is transmitted around the Loop.

2. Then Loop devices start constantly emitting *Loop Initialization Select Master* (LISM) frames. If a Fabric is present, it becomes the *Loop Master*. If not, the Loop devices with the lowest address becomes the Loop Master.

3. Eventually, the Loop devices select an Arbitrated Loop Physical Address. The Loop Master emits frames in which Loop devices select a bit. If a Loop device had a valid address before initialization, it will try to reclaim this address.

Once the Loop is usable, data transmission can take place. It just consists of two steps:

1. When a device X is ready to transmit data, it first arbitrates the Arbitrated Loop. It does that by sending the *Arbitrated Primitive Signal* together with its own Arbitrated Loop Physical Address. When it receives back its own Arbitrate Primitive Signal, it has gained control of the Loop and can now communicate by transmitting an *Open Primitive Signal* to device Y.

2. There is now merely a point-to-point communication between devices X and Y. The communication may benefit from many strategies described in terms of Classes of service as seen in Section 7.4.

If more than one device on the Loop is arbitrating at the same time, their Arbitrate Loop Physical Address are compared. Arbitrating devices block higher Arbitrate Loop Physical Address and forward lower Arbitrate Loop Physical Address. Thus, the device with the lower Arbitrate Loop Physical Address will gain control of the Loop first. The other device must wait for the Loop to be released.

There is no limit on how long a device may retain control of the Loop (that's why we say it's a channel). Yet, there is however an (optional) Access Fairness Algorithm to prevent devices from arbitrating over again.

Hubs The cabling is made easier by connecting all the devices to a hub.

7.6.3 Fabric

The Fabric topology is used to connect many devices in a switched configuration. The media is not shared so that many devices can communicate at the same time. See Figure 8.

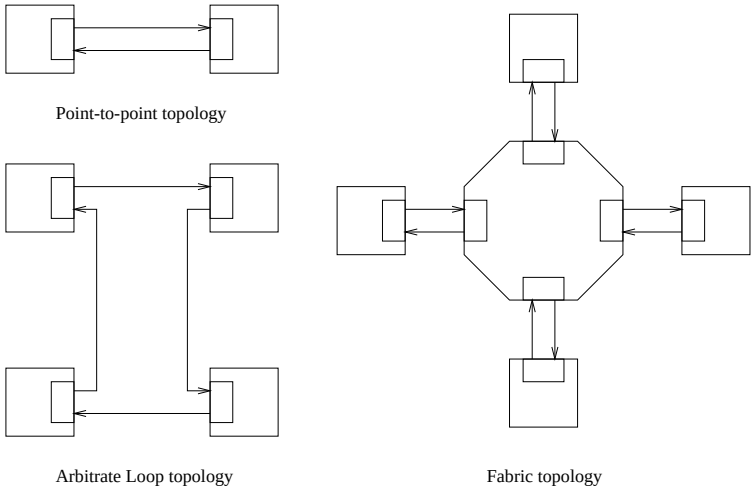

Figure 8: Fibre Channel Topologies

8 Storage Area Network (SAN)

8.1 From SCSI to SAN

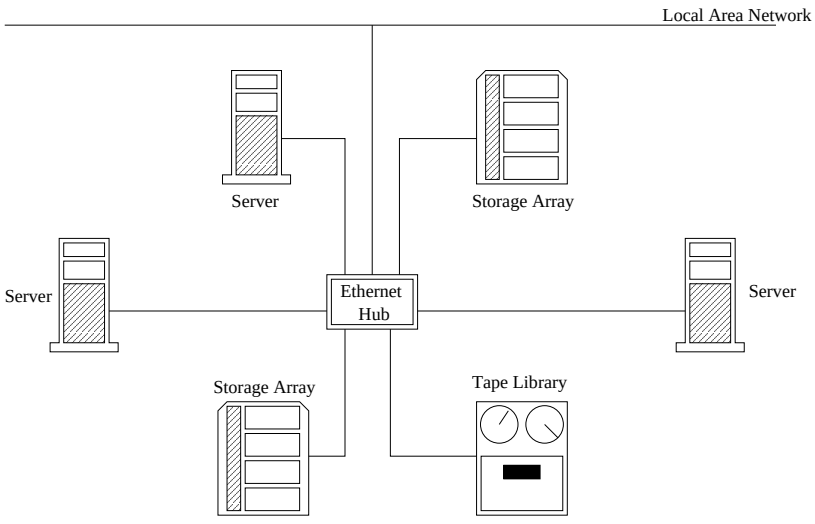

Figure 9: Network Attached Storage

To have storage devices available on a network, one may attach them to a server using a SCSI interface. But SCSI has many limitations that restrict placing any kind of storage device anywhere on a network. One solution is to use a Network Attached Storage (NAS): a network to which storage devices are connected through a traditional network interface such as Ethernet. This way, a disk may connect to a hub as computers do and be available to all the computers on the network. See Figure 9.

However, this model has many drawbacks inherited from the underlying network. For instance, the network bandwidth limits the storage devices capabilities. Although SCSI may operate at speeds up to 160 MBps, most NASs use 10 Mbps or 100 Mbps Ethernet Local Area Networks (LANs). Indeed, most vendors today only offer interfaces up to Fast Ethernet.

Storage Area Networks (SANs) intend overcoming those drawbacks. A SAN interconnects servers and storage devices (Redundant Arrays of Independent Disks (RAIDs), optical storage systems, tape libraries, etc.) through a fast network, typically Fibre Channel. There are two ways two create a SAN based on a Fibre Channel network. Storage devices and servers may be linked by a Fibre Channel Arbitrated Loop (see Figure 10) or by a Fibre Channel Fabric (see Figure 11). The servers still communicates with each other over a LAN.

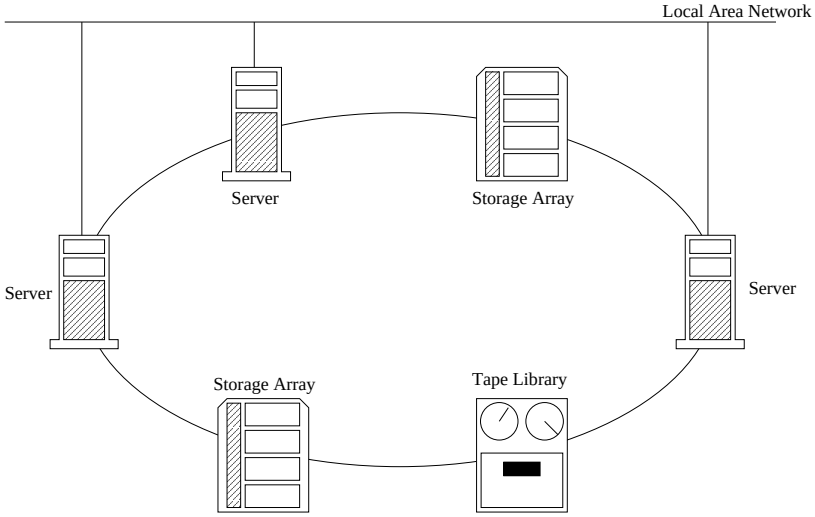

Figure 10: Arbitrated Loop SAN

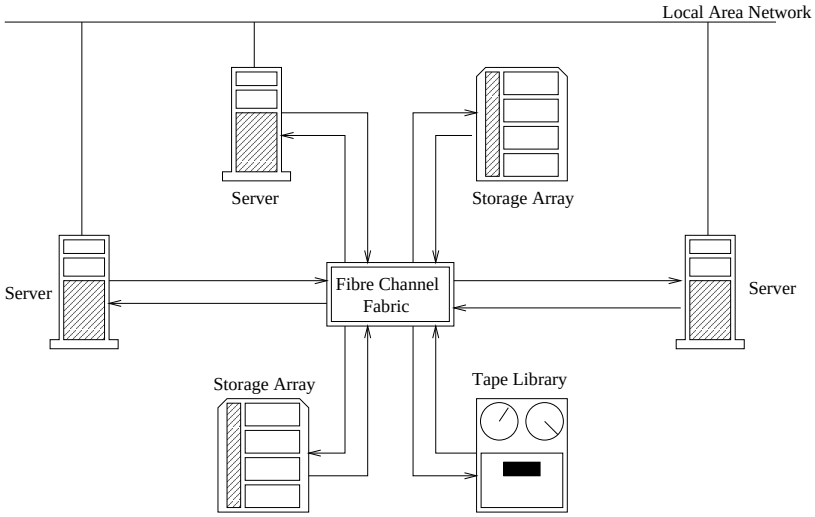

Figure 11: Fabric SAN

8.2 Advantages of SANs

The concept of SAN really emerged with new networking technologies as Fibre Channel on the advantages of which it relies.

- *Fast network.* Two devices on a SAN may achieve communications at higher speed (see Table 3 in case of a Fibre Channel SAN).

- *Alternative paths.* If a particular server is slow or unavailable, another server on the SAN can provide an alternative path to the desired storage device at the same (high) speed.
- *Hot-pluggability.* When more storage is needed, additional storage devices can simply be added to the SAN. Similarly, local failures doesn't bring the whole network to be down. Indeed, Fibre Channel supports dynamic addition and removal without interruption of the data flow (see Section 7.6).
- *Cost-effectiveness.* (See Section 7.2) Many companies are shipping SCSI adapter cards to Fibre Channel, so that one may still benefit from SCSI devices. The same paths may be used by many protocols, which reduced costs and complexity. Moreover, SANs allow for a better sharing of storage resources.
- *Today's networks.* Such a storage-centric model is better suited to today's Local and Wide Area Networks (LANs and WANs) that are overwhelmed by data. Data transfers are indeed moved to the SAN and therefore leave more LAN/WAN bandwidth for applications.

8.3 Applications

SANs allow for new kinds of application that alleviate maintenance and increase usability of LANs and WANs.

- *Mirroring.* SANs make it possible to easily make multiple copies of the same storage available to distant users over a broad corporate network.
- *Recovery.* A SAN has advantages in disaster recovery, since copies may be placed offsite.
- *Data-mining.* SANs offer dedicated high-performance paths for applications that depend on massive data transfers such as data-mining applications.
- *Backup.* SANs are today typically used to backup and restore data without overlo[ading the corporate network.
- *Disk on demand.* Through a user-friendly Graphical User Interface (GUI), one may assign disk capacity to SAN servers without even rebooting any part of the network.

References

[1] The Fibre Channel Industry Association, *Fibre Channel Frequently Asked Questions*, `http://www.fibrechannel.com/FAQs/fcfaqstart.html`.

[2] University of New Hampshire InterOperability Lab's Fibre Channel Consortium, *UNH IOL Fibre Channel Consortium*, `http://www.iol.unh.edu/consortiums/fc/index.html`, 1998.

[3] Zoltán Meggyesi, *Fibre Channel Overview*, `http://www1.cern.ch/HSI/fcs/spec/overview.html`, 1994.

[4] Erik van der Bij, *Fibre Channel Specifications*, `http://hsi.web.cern.ch/HSI/fcs/spec.htm`, 1994.

[5] Technical Committee T11, *Welcome to the Home Page for TC T11*, `http://www.t11.org/index.html`, 2000.

[6] M. Rajagopal, R. Bhagwat, W. Rickard, *IP and ARP over Fibre Channel*, `http://www.ietf.org/rfc/rfc2625.txt`, 1999.