Module-3

IP SAN and FCoE

iSCSI

- iSCSI is an IP based protocol that establishes and manages connections between host and storage over IP, as shown in Fig below.
- iSCSI encapsulates SCSI commands and data into an IP packet and transports them using TCP/IP.
- iSCSI is widely adopted for connecting servers to storage because it is relatively inexpensive and easy to implement, especially in environments in which an FC SAN does not exist.

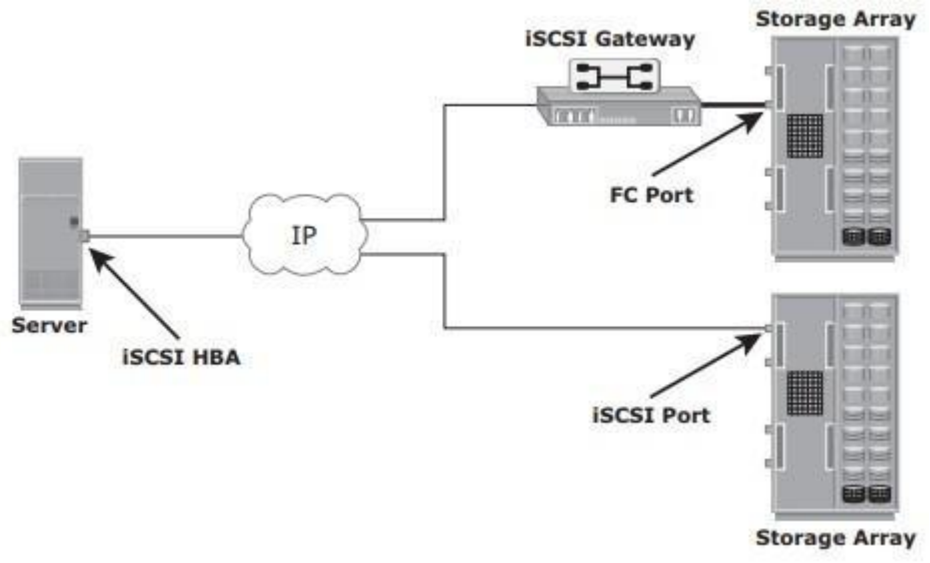


Fig : iSCSI implementation

Components of iSCSI

- An initiator (host), target (storage or iSCSI gateway), and an IP-based network are the key iSCSI components.

- If an iSCSI-capable storage array is deployed, then a host with the iSCSI initiator can directly communicate with the storage array over an IP network.
- However, in an implementation that uses an existing FC array for iSCSI communication, an iSCSI gateway is used.
- These devices perform the translation of IP packets to FC frames and vice versa, thereby bridging the connectivity between the IP and FC environments.

iSCSI Host Connectivity

The three iSCSI host connectivity options are:

- A standard NIC with software iSCSI initiator,
- a TCP offload engine (TOE) NIC with software iSCSI initiator,
- an iSCSI HBA

- The function of the iSCSI initiator is to route the SCSI commands over an IP network.
- A **standard NIC with a software iSCSI** initiator is the simplest and least expensive connectivity option. It is easy to implement because most servers come with at least one, and in many cases two, embedded NICs. It requires only a software initiator for iSCSI functionality. Because NICs provide standard IP function, encapsulation of SCSI into IP packets and decapsulation are carried out by the host CPU. This places additional overhead on the host CPU. If a standard NIC is used in heavy I/O load situations, the host CPU might become a bottleneck. TOE NIC helps reduce this burden.
- A **TOE NIC** offloads TCP management functions from the host and leaves only the iSCSI functionality to the host processor. The host passes the iSCSI information to the TOE card, and the TOE card sends the information to the destination using TCP/IP. Although this solution improves performance, the iSCSI functionality is still handled by a software initiator that requires host CPU cycles.
- An **iSCSI HBA** is capable of providing performance benefits because it offloads the entire

iSCSI and TCP/IP processing from the host processor. The use of an iSCSI HBA is also the simplest way to boot hosts from a SAN environment via iSCSI. If there is no iSCSI HBA, modifications must be made to the basic operating system to boot a host from the storage devices because the NIC needs to obtain an IP address before the operating system loads. The functionality of an iSCSI HBA is similar to the functionality of an FC HBA.

iSCSI Topologies

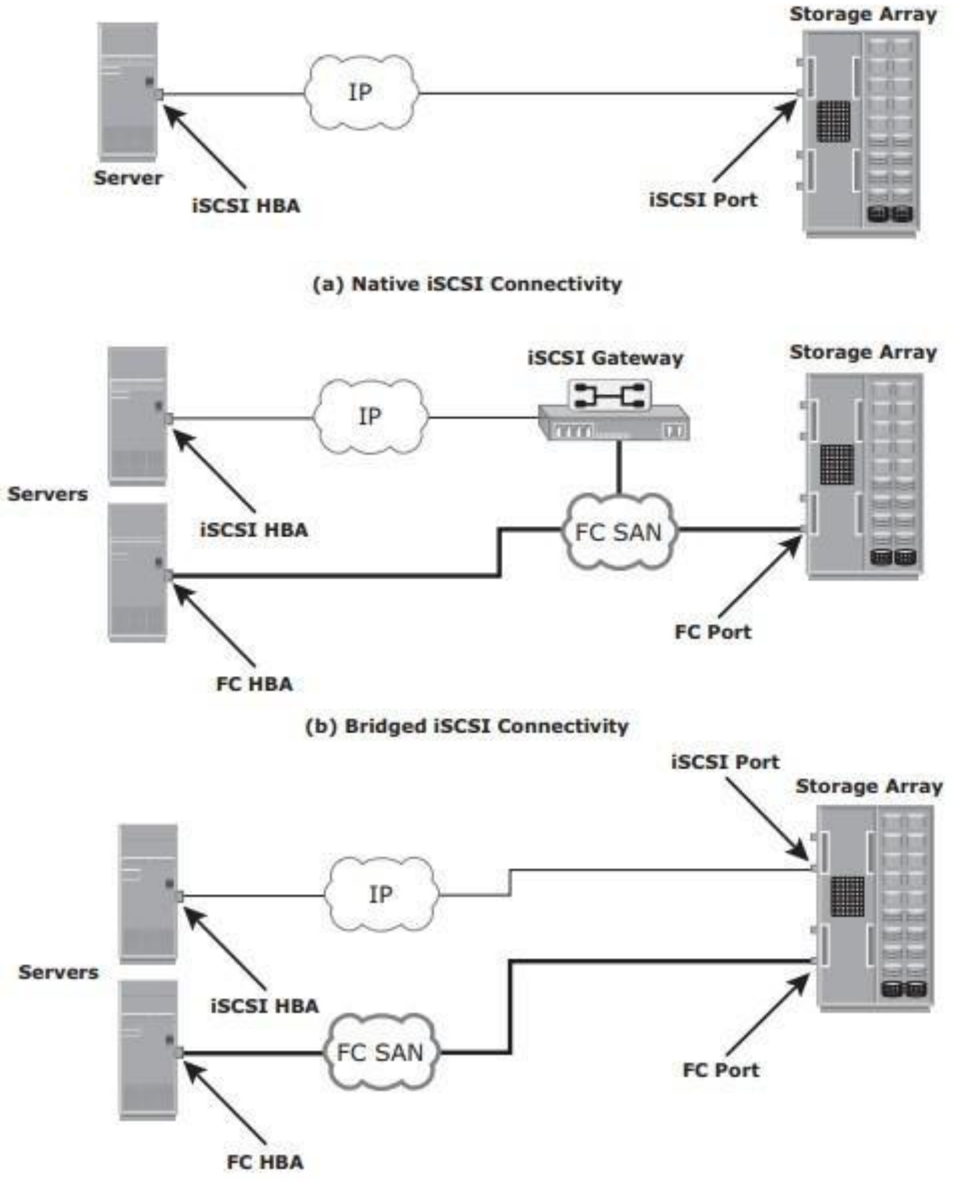
- Two topologies of iSCSI implementations are **native and bridged**.
- Native topology does not have FC components.
- The initiators may be either directly attached to targets or connected through the IP network.
- Bridged topology enables the coexistence of FC with IP by providing iSCSI-to-FC bridging functionality.
- For example, the initiators can exist in an IP environment while the storage remains in an FC environment.

Native iSCSI Connectivity

- FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed.
- In Fig (a), the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernet switch.
- After an initiator is logged on to the network, it can access the available LUNs on the storage array.
- A single array port can service multiple hosts or initiators as long as the array port can handle the amount of storage traffic that the hosts generate.

Bridged iSCSI Connectivity

- A bridged iSCSI implementation includes FC components in its configuration.
- Fig (b), illustrates iSCSI host connectivity to an FC storage array. In this case, the array does not have any iSCSI ports. Therefore, an external device, called a gateway or a multiprotocol router, must be used to facilitate the communication between the iSCSI host and FC storage.
- The gateway converts IP packets to FC frames and vice versa.
- The bridge devices contain both FC and Ethernet ports to facilitate the communication between the FC and IP environments.
- In a bridged iSCSI implementation, the iSCSI initiator is configured with the gateway's IP address as its target destination.
- On the other side, the gateway is configured as an FC initiator to the storage array.
- **Combining FC and Native iSCSI Connectivity:** The most common topology is a combination of FC and native iSCSI. Typically, a storage array comes with both FC and iSCSI ports that enable iSCSI and FC connectivity in the same environment, as shown in Fig (c).



(a) Native iSCSI Connectivity

(b) Bridged iSCSI Connectivity

(c) Combining FC and Native iSCSI Connectivity

Fig : iSCSI Topologies

iSCSI Protocol Stack

- Fig 2.23 displays a model of the iSCSI protocol layers and depicts the encapsulation order of the SCSI commands for their delivery through a physical carrier.

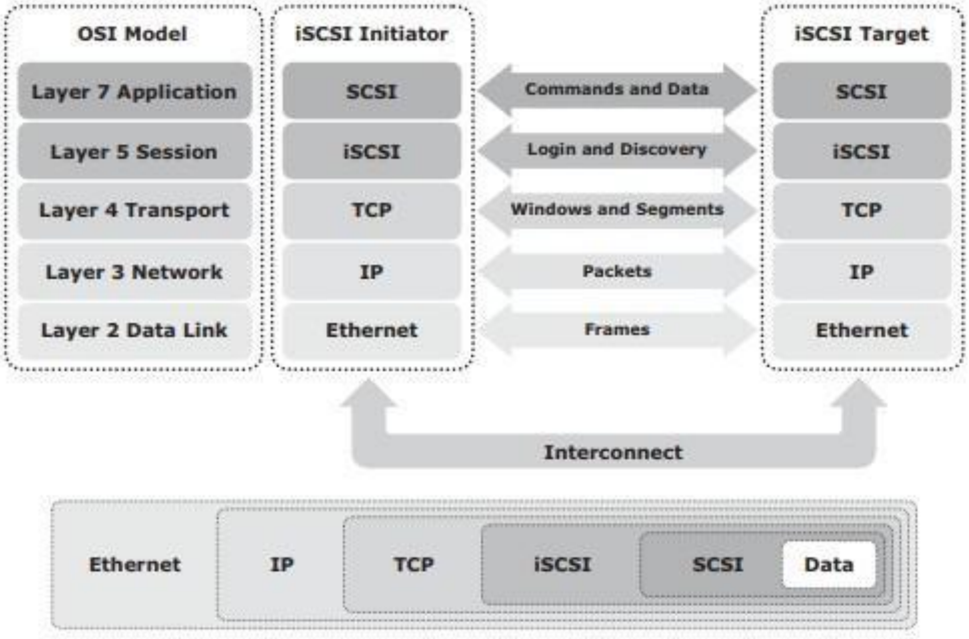


Fig 2.23: iSCSI protocol stack

- SCSI is the command protocol that works at the application layer of the Open System Interconnection (OSI) model.
- The initiators and targets use SCSI commands and responses to talk to each other.
- The SCSI command descriptor blocks, data, and status messages are encapsulated into TCP/IP and transmitted across the network between the initiators and targets.
- iSCSI is the session-layer protocol that initiates a reliable session between devices that recognize SCSI commands and TCP/IP.
- The iSCSI session-layer interface is responsible for handling login, authentication, target discovery, and session management.

- TCP is used with iSCSI at the transport layer to provide reliable transmission.
- TCP controls message flow, windowing, error recovery, and retransmission.
- It relies upon the network layer of the OSI model to provide global addressing and connectivity.
- The Layer 2 protocols at the data link layer of this model enable node-to-node communication through a physical network.

iSCSI PDU

- A *protocol data unit* (PDU) is the basic "information unit" in the iSCSI environment.
- The iSCSI initiators and targets communicate with each other using iSCSI PDUs. This communication includes establishing iSCSI connections and iSCSI sessions, performing iSCSI discovery, sending SCSI commands and data, and receiving SCSI status.
- All iSCSI PDUs contain one or more header segments followed by zero or more data segments.
- The PDU is then encapsulated into an IP packet to facilitate the transport.
- A PDU includes the components shown in Fig below.
- The IP header provides packet-routing information to move the packet across a network.
- The TCP header contains the information required to guarantee the packet delivery to the target.
- The iSCSI header (basic header segment) describes how to extract SCSI commands and data for the target. iSCSI adds an optional CRC, known as the *digest*, to ensure datagram integrity. This is in addition to TCP checksum and Ethernet CRC.
- The header and the data digests are optionally used in the PDU to validate integrity and data placement.

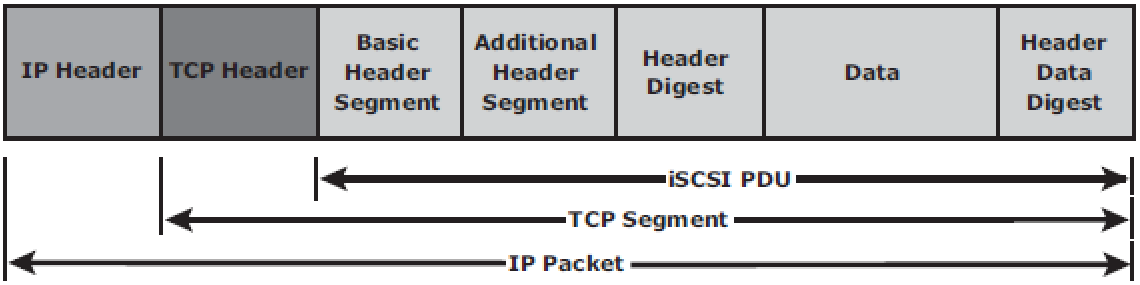


Fig : iSCSI PDU encapsulated in an IP packet

iSCSI Discovery

- An initiator must discover the location of its targets on the network and the names of the targets available to it before it can establish a session.
- This discovery can take place in two ways:
  - **SendTargets discovery**
  - **internet Storage Name Service (iSNS).**
- In *SendTargets discovery*, the initiator is manually configured with the target's network portal to establish a discovery session. The initiator issues the SendTargets command, and the target network portal responds with the names and addresses of the targets available to the host.
- iSNS (Fig below) enables automatic discovery of iSCSI devices on an IP network. The initiators and targets can be configured to automatically register themselves with the iSNS server. Whenever an initiator wants to now the targets that it can access, it can query the iSNS server for a list of available targets.
- The discovery can also take place by using service location protocol (SLP). However, this is less commonly used than SendTargets discovery and iSNS.

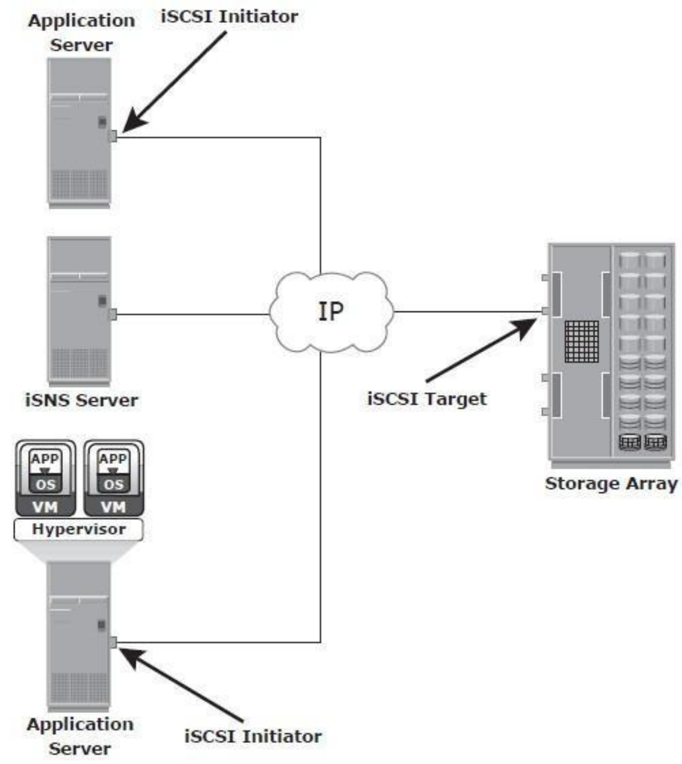


Fig : Discovery using iSNS

iSCSI Names

- A unique worldwide iSCSI identifier, known as an *iSCSI name*, is used to identify the initiators and targets within an iSCSI network to facilitate communication.
- The unique identifier can be a combination of the names of the department, application, or manufacturer, serial number, asset number, or any tag that can be used to recognize and manage the devices.
- Following are two types of iSCSI names commonly used:
    - **iSCSI Qualified Name (IQN):**
    - **Extended Unique Identifier (EUI)**

- **iSCSI Qualified Name (IQN):** An organization must own a registered domain name to generate iSCSI Qualifi ed Names. This domain name does not need to be active or resolve to an address. It just needs to be reserved to prevent other organizations from using the same domain name to generate iSCSI names. A date is included in the name to avoid potential conflicts caused by the transfer of domain names.

  An example of an IQN is iqn.2008-02.com.example:*optional_string*. The *optional_string* provides a serial number, an asset number, or any otherdevice identifiers.

- **Extended Unique Identifi er (EUI):** An EUI is a globally unique identifier based on the IEEE EUI-64 naming standard. An EUI is composed of the eui prefix followed by a 16-character hexadecimal name, such aseui.0300732A32598D26.
- In either format, the allowed special characters are dots, dashes, and blank spaces.

iSCSI Session

- An iSCSI session is established between an initiator and a target, as shown in Fig.
- A session is identified by a session ID (SSID), which includes part of an initiator ID and a target ID.
- The session can be intended for one of the following:
  - The discovery of the available targets by the initiators and the location of a specific target on a network
  - The normal operation of iSCSI (transferring data between initiators and targets)
- There might be one or more TCP connections within each session. Each TCP connection within the session has a unique connection ID (CID).
- An iSCSI session is established via the iSCSI login process. The login process is started when the initiator establishes a TCP connection with the required target either via the well-known port 3260 or a specified target port.

- During the login phase, the initiator and the target authenticate each other and negotiate on various parameters.
- After the login phase is successfully completed, the iSCSI session enters the full-feature phase for normal SCSI transactions. In this phase, the initiator may send SCSI commands and data to the various LUNs on the target.
- The final phase of the iSCSI session is the connection termination phase, which is referred to as the logout procedure.
- The initiator is responsible for commencing the logout procedure; however, the target may also prompt termination by sending an iSCSI message, indicating the occurrence of an internal error condition.
- After the logout request is sent from the initiator and accepted by the target, no further request and response can be sent on that connection.

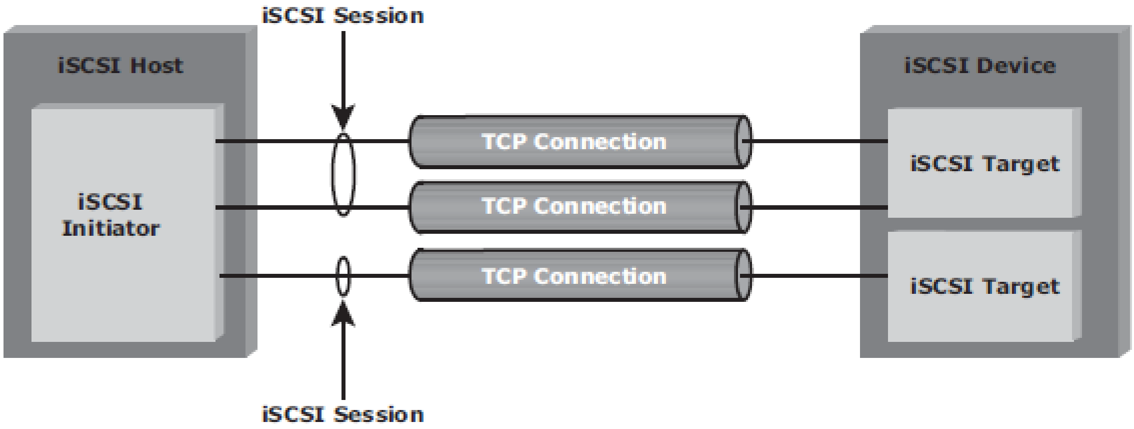


Fig : iSCSI session

Command Sequencing

- The iSCSI communication between the initiators and targets is based on the request-response command sequences.

- A command sequence may generate multiple PDUs.
- A ***command sequence number*** **(CmdSN**) within an iSCSI session is used for numbering all initiator-to-target command PDUs belonging to the session.
- This number ensures that every command is delivered in the same order in which it is transmitted, regardless of the TCP connection that carries the command in the session.
- Command sequencing begins with the first login command, and the CmdSN is incremented by one for each subsequent command.
- The iSCSI target layer is responsible for delivering the commands to the SCSI layer in the order of their CmdSN.
- Similar to command numbering, a ***status sequence number*** **(StatSN)** is used to sequentially number status responses, as shown in Fig.
- These unique numbers are established at the level of the TCP connection.
- A target sends ***request-to-transfer*** **(R2T)** PDUs to the initiator when it is ready to accept data.
- A *data sequence number* (DataSN) is used to ensure in-order delivery of data within the same command.
- The DataSN and R2TSN are used to sequence data PDUs and R2Ts, respectively.

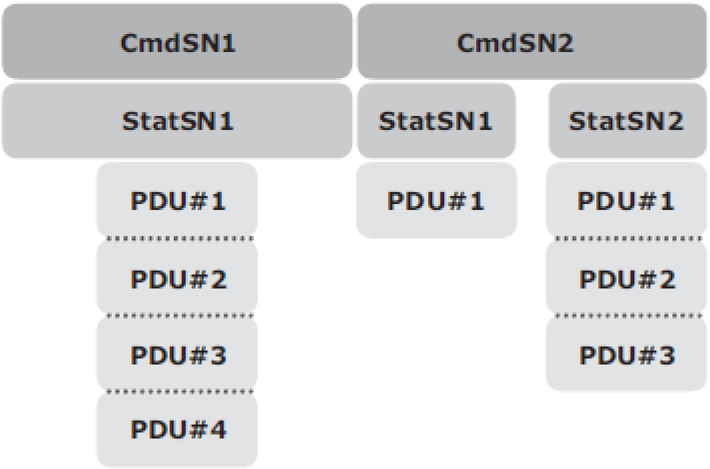


Fig : Command and status sequence number

FCIP (Fibre channel over IP)

- FCIP is a IP-based protocol that is used to connect distributed FC-SAN islands.
- Creates virtual FC links over existing IP network that is used to transport FC data between different FC SANS.
- It encapsulates FC frames into IP packet.
- It provides disaster recovery solution.

FCIP Protocol Stack

- The FCIP protocol stack is shown in Fig below. Applications generate SCSI commands and data, which are processed by various layers of the protocol stack.
- The upper layer protocol SCSI includes the SCSI driver program that executes the read-and-write commands.
- Below the SCSI layer is the Fibre Channel Protocol (FCP) layer, which is simply a Fibre Channel frame whose payload is SCSI.
- The FCP layer rides on top of the Fibre Channel transport layer. This enables the FC frames to run natively within a SAN fabric environment. In addition, the FC frames can be encapsulated into the IP packet and sent to a remote SAN over the IP.

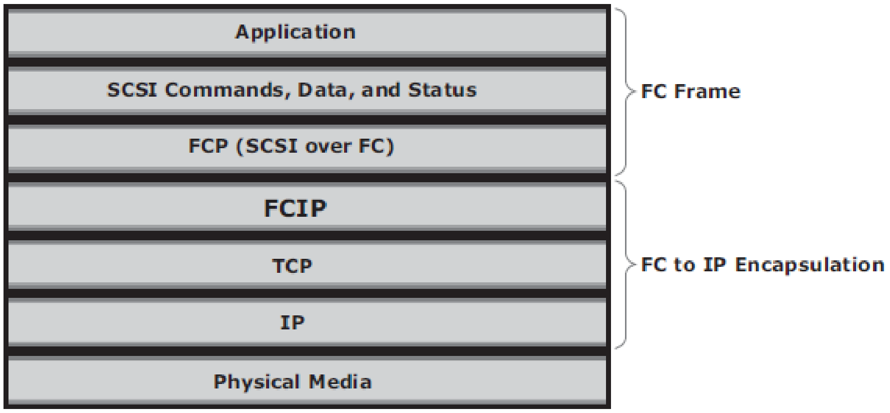


Fig : FCIP protocol stack

- The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Fig). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IP traffic.
- Encapsulation of FC frame into an IP packet could cause the IP packet to be fragmented when the data link cannot support the maximum transmission unit (MTU) size of an IP packet.
- When an IP packet is fragmented, the required parts of the header must be copied by all fragments.
- When a TCP packet is segmented, normal TCP operations are responsible for receiving and re-sequencing the data prior to passing it on to the FC processing portion of the device.

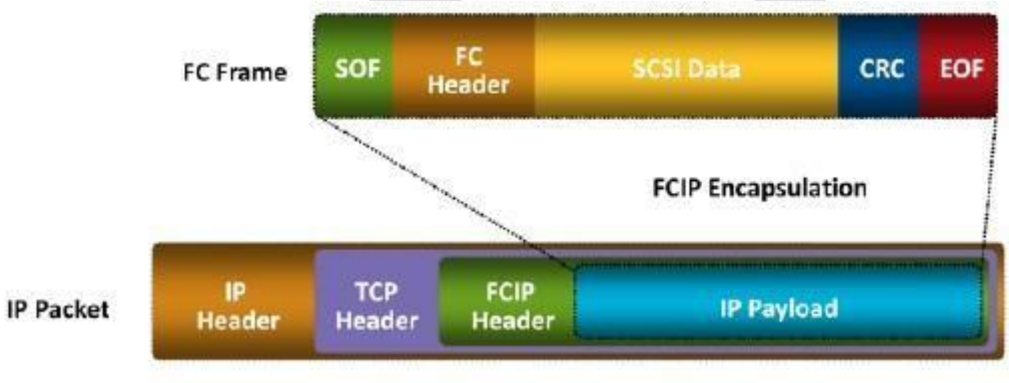


Fig FCIP encapsulation

FCIP Topology

- In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (Fig).
- The FCIP gateway at one end of the IP network encapsulates the FC frames into IP packets.
- The gateway at the other end removes the IP wrapper and sends the FC data to the layer 2 fabric.
- The fabric treats these gateways as layer 2 fabric switches.
- An IP address is assigned to the port on the gateway, which is connected to an IP network. After the IP connectivity is established, the nodes in the two independent fabrics can communicate with each other.

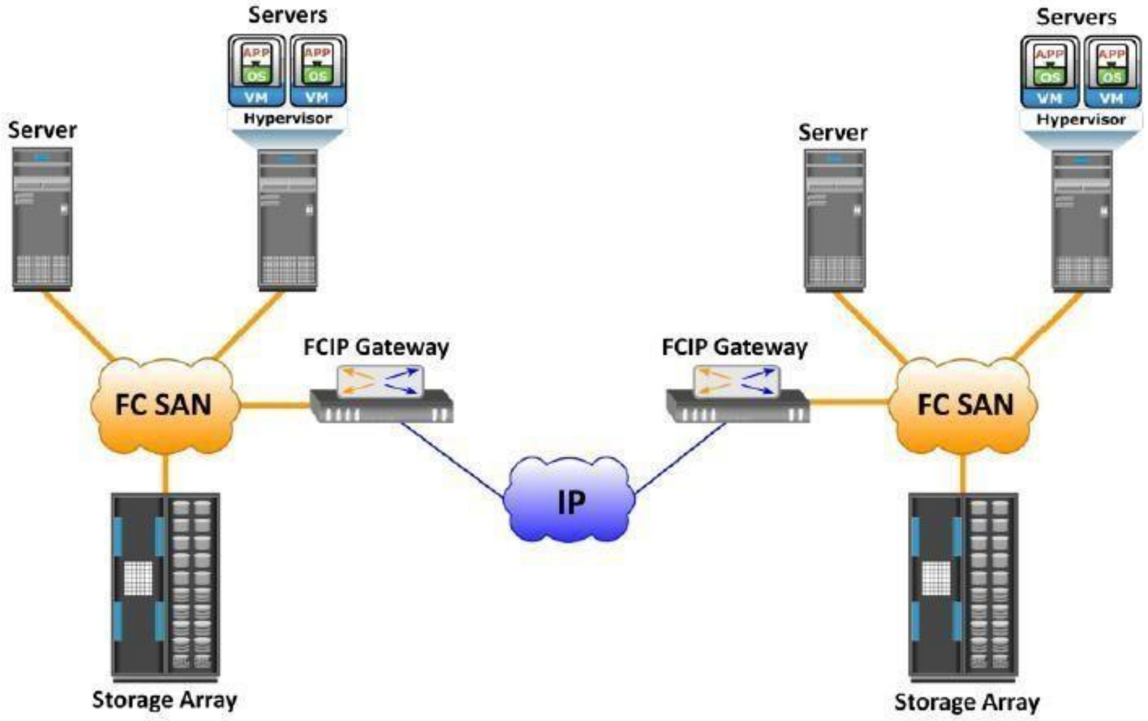


Fig : FCIP topology

FCoE (Fibre Channel over Ethernet)

- Data centers typically have multiple networks to handle various types of I/O traffic — for example, an Ethernet network for TCP/IP communication and an FC network for FC communication.

- TCP/IP is typically used for client-server communication, data backup, infrastructure management communication, and so on.
- FC is typically used for moving block-level data between storage and servers.
- To support multiple networks, servers in a data center are equipped with multiple redundant physical network interfaces — for example, multiple Ethernet and FC cards/adapters. In addition, to enable the communication, different types of networking switches and physical cabling infrastructure are implemented in data centers.

- The need for two different kinds of physical network infrastructure increases the overall cost and complexity of data center operation.
- Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffic over a single physical interface infrastructure.
- FCoE helps organizations address the challenges of having multiple discrete network infrastructures.
- FCoE uses the Converged Enhanced Ethernet (CEE) link (10 Gigabit Ethernet) to send FC frames over Ethernet.

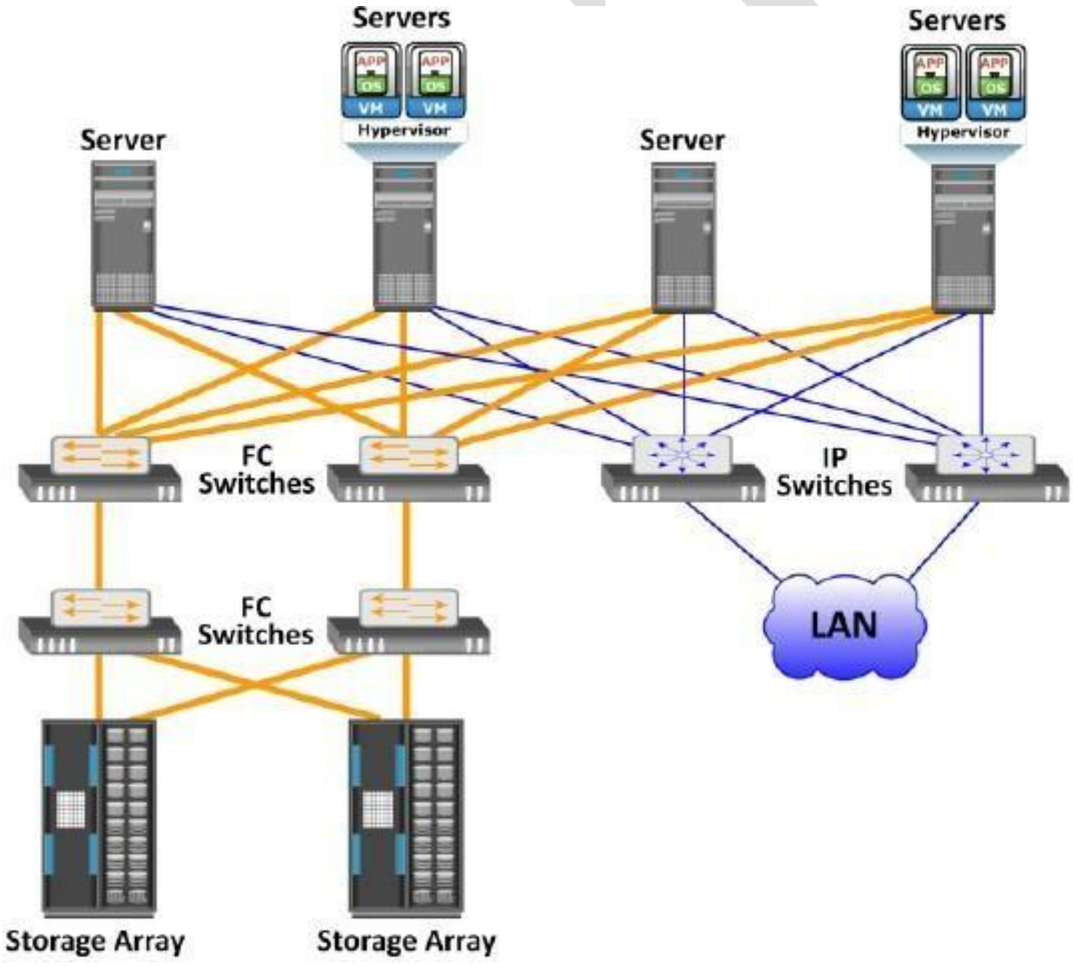


Fig Before using FCOE

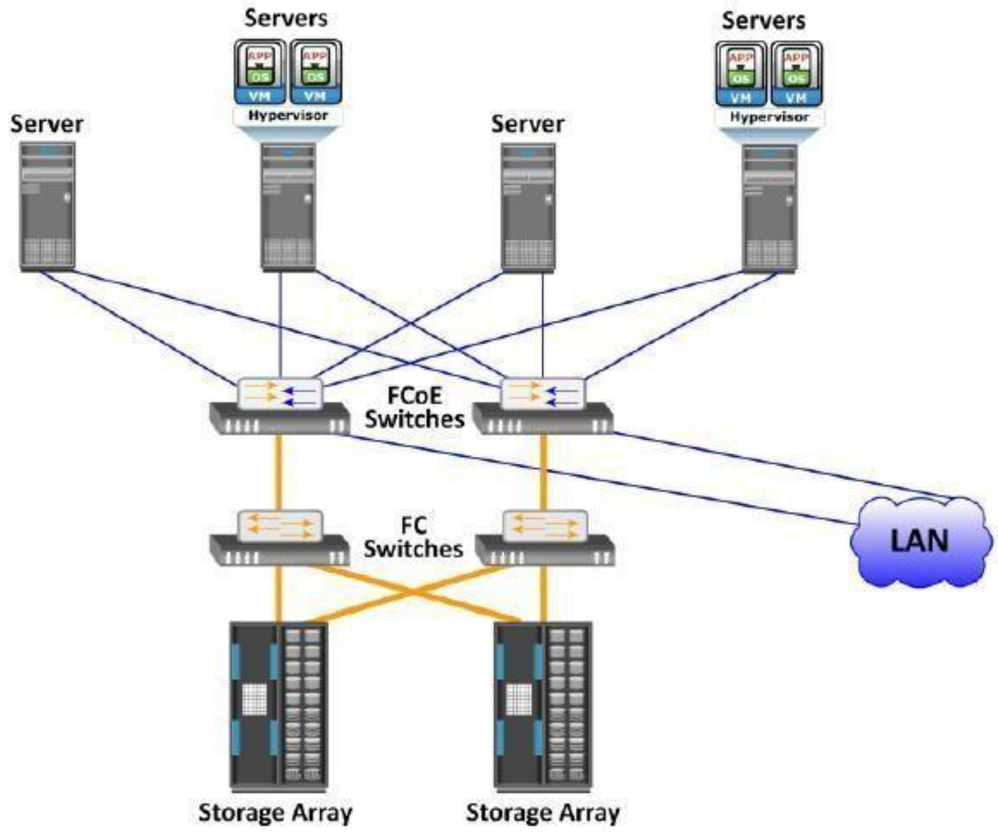


Fig After using FCOE

Components of FCOE

The key components of FCOE are :

- Converged Network Adaptors(CNA)
- Cables
- FCOE Switches

Converged Network Adaptors(CNA)

- A CNA provides the functionality of both a standard NIC and an FC HBA in a single adapter and consolidates both types of traffic. CNA eliminates the need to deploy separate adapters and cables for FC and Ethernet communications, thereby reducing the required number of server slots and switch ports.

- As shown in Fig below, a CNA contains separate modules for 10 Gigabit Ethernet, Fibre Channel, and FCoE Application Specific Integrated Circuits (ASICs). The FCoE ASIC encapsulates FC frames into Ethernet frames. One end of this ASIC is connected to 10GbE and FC ASICs for server connectivity, while the other end provides a 10GbE interface to connect to an FCoE switch.

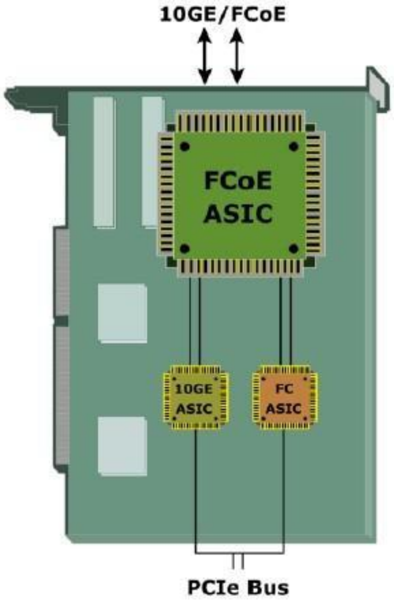


Fig : Converged Network Adapter

Cables

- There are two options available for FCoE cabling:

  1. Copper based Twinax
  2. standard fiber optical cables.

- A Twinax cable is composed of two pairs of copper cables covered with a shielded casing. The Twinax cable can transmit data at the speed of 10 Gbps over shorter distances up to 10 meters. Twinax cables require less power and are less expensive than fi ber optic cables.

- The Small Form Factor Pluggable Plus (SFP+) connector is the primary connector used for FCoE links and can be used with both optical and copper cables.

FCoE Switches

- An FCoE switch has both **Ethernet switch** and **Fibre Channel switch** functionalities.
- As shown in Fig below, FCoE switch consists of:
    1. *Fibre Channel Forwarder (FCF),*
    2. *Ethernet Bridge*,
    3. set of Ethernet ports
    4. optional FC ports
- The function of the FCF is to encapsulate the FC frames, received from the FC port, into the FCoE frames and also to de-encapsulate the FCoE frames, received from the Ethernet Bridge, to the FC frames.
- Upon receiving the incoming traffic, the FCoE switch inspects the **Ethertype** (used to indicate which protocol is encapsulated in the payload of an Ethernet frame) of the incoming frames and uses that to determine the destination.
    - If the Ethertype of the frame is FCoE, the switch recognizes that the frame contains an FC payload and forwards it to the FCF. From there, the FC is extracted from the FCoE frame and transmitted to FC SAN over the FC ports.
    - If the Ethertype is not FCoE, the switch handles the traffic as usual Ethernet traffic and forwards it over the Ethernet ports.

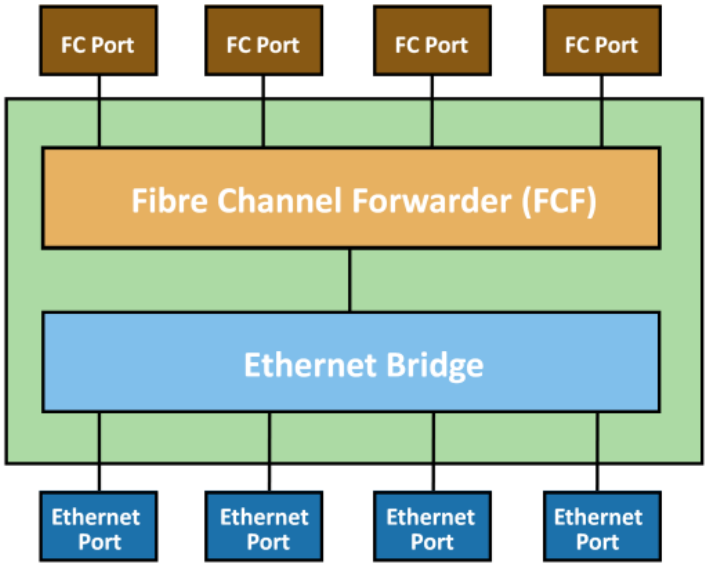


Fig FCoE switch generic architecture

NETWORK ATTACHED STORAGE (NAS)

File Sharing Environment

- File sharing enables users to share files with other users
- In file-sharing environment, the creator or owner of a file determines the type of access to be given to other users and controls changes to the file.
- When multiple access a shared file at the same time, a locking scheme is required to maintain data integrity and also make this sharing possible. This is taken care by file-sharing environment.
- Examples of file sharing methods:
  - File Transfer Protocol (FTP)
  - Distributed File System (DFS)
  - Network File System (NFS) and Common Internet File System (CIFS)
  - Peer-to-Peer (P2P)

What is NAS?

- NAS is an IP based dedicated, high-performance file sharing and storage device.
- Enables NAS clients to share files over an IP network.
- Uses network and file-sharing protocols to provide access to the file data.
- Ex: Common Internet File System (CIFS) and Network File System (NFS).
- Enables both UNIX and Microsoft Windows users to share the same data seamlessly.
- NAS device uses its own operating system and integrated hardware and software components to meet specific file-service needs.
- Its operating system is optimized for file I/O which performs better than a general-purpose server.
- A NAS device can serve more clients than general-purpose servers and provide the benefit of server consolidation.

Components of NAS

- NAS device has *two* key components (as shown in Fig 2.33): **NAS head** and **storage**.
- In some NAS implementations, the storage could be external to the NAS device and shared with other hosts.
- NAS head includes the following components:
  - CPU and memory
  - One or more network interface cards (NICs), which provide connectivity to the client network.
  - An optimized operating system for managing the NAS functionality. It translates file-level requests into block-storage requests and further converts the data supplied at the block level to file data
  - NFS, CIFS, and other protocols for file sharing

- Industry-standard storage protocols and ports to connect and manage physical disk resources

➢ The NAS environment includes clients accessing a NAS device over an IP network using file-sharing protocols.

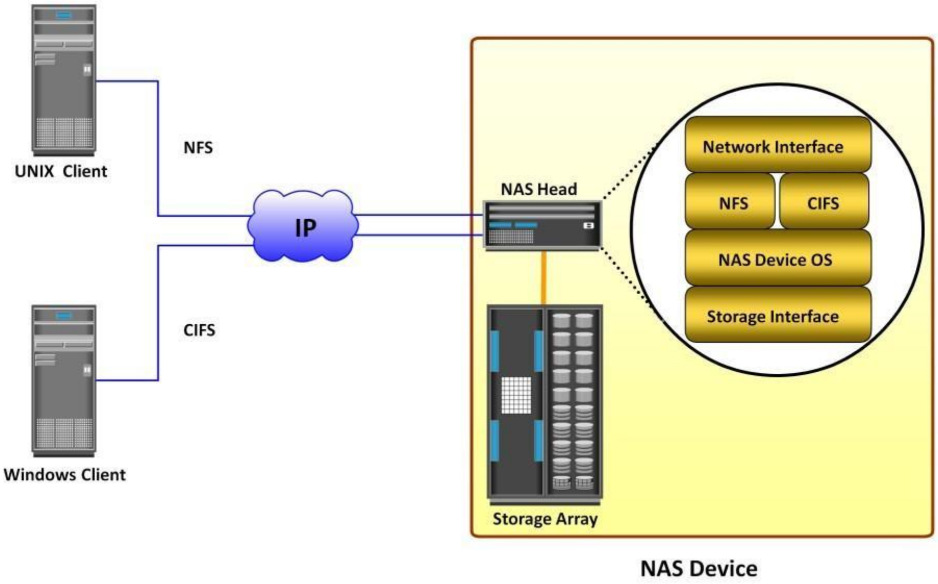


Fig 2.33 Components of NAS

NAS I/O Operation

➢ NAS provides *file-level data access* to its clients. File I/O is a high-level request that specifies the file to be accessed.

➢ Eg: a client may request a file by specifying its name, location, or other attributes. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrieve data.

➢ The process of handling I/Os in a NAS environment is as follows:

1. The requestor (client) packages an I/O request into TCP/IP and forwards it through the network stack. The NAS device receives this request from the network.

2. The NAS device converts the I/O request into an appropriate physical storage request, which is a block-level I/O, and then performs the operation on the physical storage.
3. When the NAS device receives data from the storage, it processes and repackages the data into an appropriate file protocol response.
4. The NAS device packages this response into TCP/IP again and forwards it to the client through the network.

- Fig 2.34 illustrates the NAS I/O operation

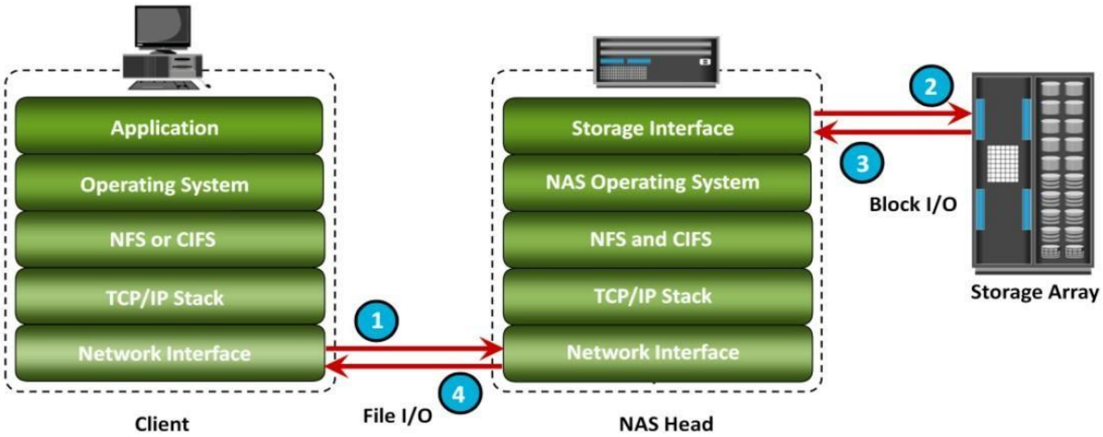


Fig 2.34 NAS I/O Operation

NAS File Sharing Protocols

- NAS devices support multiple file-service protocols to handle file I/O requests
- Two common NAS file sharing protocols are:
  - Common Internet File System (CIFS)
  - Network File System (NFS)
- NAS devices enable users to share file data across different operating environments
- It provides a means for users to migrate transparently from one operating system to another

Network File System (NFS)

- NFS is a **client-server protocol** for file sharing that is commonly used on **UNIX systems**.
- NFS was originally based on the connectionless *User Datagram Protocol (UDP).*
- It uses *Remote Procedure Call (RPC)* as a method of inter-process communication between two computers.
- The NFS protocol provides a set of RPCs to access a remote file system for the following operations:
    - Searching files and directories
    - Opening, reading, writing to, and closing a file
    - Changing file attributes
    - Modifying file links and directories
- NFS creates a connection between the client and the remote system to transfer data.
- NFSv3 and earlier is a stateless protocol
- It does not maintain any kind of table to store information about open files and associated pointers. Each call provides a full set of arguments - a file handle, a particular position to read or write, and the versions of NFS - to access files on the server .
- Currently, three versions of NFS are in use:
    1. **NFS version 2 (NFSv2):** Uses *UDP* to provide a *stateless* network connection between a client and a server. Features, such as locking, are handled outside the protocol.
    2. **NFS version 3 (NFSv3):** Uses *UDP or TCP*, and is based on the *stateless protocol* design. It includes some new features, such as a 64-bit file size, asynchronous writes, and additional file attributes to reduce refetching.
    3. **NFS version 4 (NFSv4):** Uses TCP and is based on a *stateful protocol* design. It offers enhanced security. The latest NFS version 4.1 is the enhancement of NFSv4 and includes some new features, such as session model, parallel NFS (pNFS), and data retention.

Common Internet File System (CIFS)

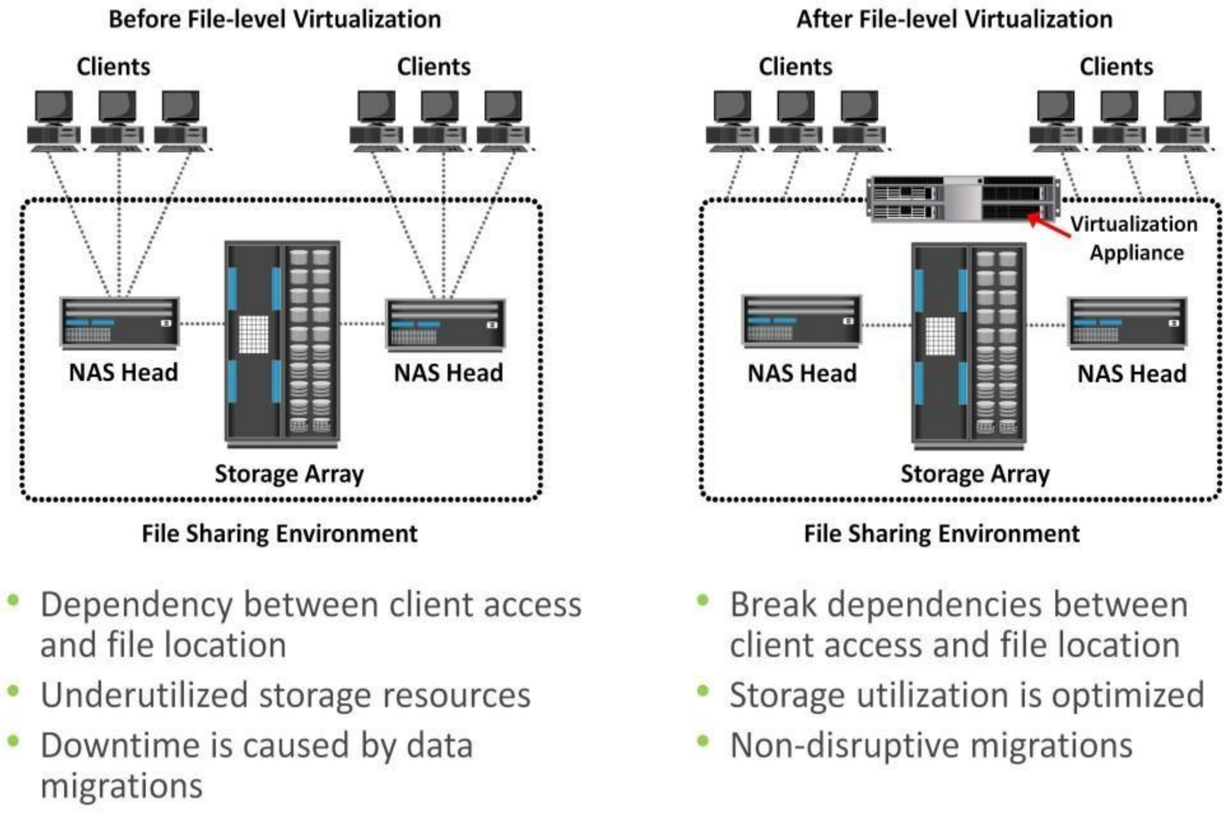
- CIFS is a *client-server application* protocol
- It enables clients to access files and services on remote computers over **TCP/IP**.
- It is a public, or open, variation of **Server Message Block (SMB)** protocol.
- It provides following features to ensure data integrity:
    - It uses file and record locking to prevent users from overwriting the work of another user on a file or a record.
    - It supports fault tolerance and can automatically restore connections and reopen files that were open prior to an interruption. This feature depends on whether an application is written to take advantage of this.
    - CIFS is a stateful protocol because the CIFS server maintains connection information regarding every connected client. If a network failure or CIFS server failure occurs, the client receives a disconnection notification. User disruption is minimized if the application has the embedded intelligence to restore the connection. However, if the embedded intelligence is missing, the user must take steps to reestablish the CIFS connection.
- Users refer to remote file systems with an easy-to-use file-naming scheme:
- Eg: \\server\share or \\servername.domain.suffix\share

File-level Virtualization

- File-level virtualization, implemented in NAS or the file server environment, provides a simple, non disruptive file-mobility solution.
- It eliminates the dependencies between data accessed at the file level and the location where the files are physically stored.
- It creates a logical pool of storage, enabling users to use a logical path, rather than a physical path, to access files.

- A global namespace is used to map the logical path of a file to the physical path names. File-level virtualization enables the movement of files across NAS devices, even if the files are being accessed.

Before and After File-level Virtualization



- Dependency between client access and file location
- Underutilized storage resources
- Downtime is caused by data migrations

- Break dependencies between client access and file location
- Storage utilization is optimized
- Non-disruptive migrations

.