MODULE 2

Syllabus: Storage Networking Technologies Fibre Channel Storage Area Networks: Components of FC SAN, FC connectivity, Fibre Channel Architecture, Zoning, FC SAN Topologies, Virtualization in SAN. IP SAN and FCoE: iSCSI, FCIP, FCoE. Network Attached Storage: Components of NAS, NAS I/O Operation, NAS File-Sharing Protocols, File-Level Virtualization, Object-Based Storage and Unified Storage: Object-Based Storage Devices, Content-Addressed Storage, Unified Storage.

Chapter 5: Fibre Channel Storage Area Networks

5.1 Fibre Channel: Overview

The FC architecture forms the fundamental construct of the FC SAN infrastructure. *Fibre Channel* is a high-speed network technology that runs on high-speed optical fiber cables and serial copper cables. The FC technology was developed to meet the demand for increased speeds of data transfer between servers and mass storage systems.

Explain the different components of FC SAN

5.3 Components of FC SAN

Node Ports

- In a Fibre Channel network, the end devices, such as hosts, storage arrays, and tape libraries, are all referred to as nodes.
- Each node is a source or destination of information.
- Each node requires one or more ports to provide a physical interface for communicating with other nodes.
- These ports are integral components of host adapters, such as HBA, and storage front-end controllers or adapters.
- In an FC environment a port operates in full-duplex data transmission mode with a transmit (Tx) link and a receive (Rx) link

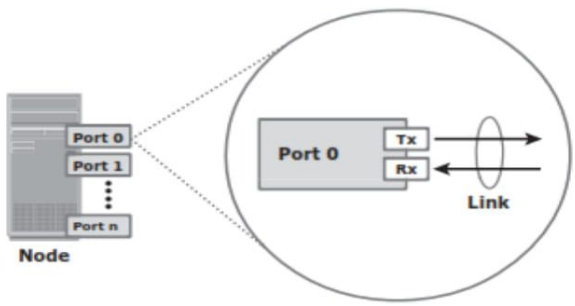


Figure 5-3: Nodes, ports, and links

Cables and Connectors

- SAN implementations use optical fiber cabling.
- Copper can be used for shorter distances for back-end connectivity because it provides an acceptable signal-to noise ratio for distances up to 30 meters.
- Optical fiber cables carry data in the form of light.
- There are two types of optical cables***: multimode and single-mode***.
- Multimode fiber (MMF) cable carries multiple beams of light projected at different angles simultaneously onto the core of the cable (see Figure 5-4 [a]).
- Based on the bandwidth, multimode fibers are classified as OM1 (62.5µm core), OM2 (50µm core), and laser-optimized OM3 (50µm core).
- In an MMF transmission, multiple light beams traveling inside the cable tend to disperse and collide.
- This collision weakens the signal strength after it travels a certain distance — a process known as modal dispersion.
- An MMF cable is typically used for short distances because of signal degradation (attenuation) due to modal dispersion.
- Single-mode fiber (SMF) carries a single ray of light projected at the center of the core
- These cables are available in core diameters of 7 to 11 microns; the most common size is 9 microns.
- In an SMF transmission, a single light beam travels in a straight line through the core of the fiber.
- The small core and the single light wave help to limit modal dispersion.
- Among all types of fiber cables, single mode provides minimum signal attenuation over maximum distance (up to 10 km).

Figure 5-4: Multimode fiber and single-mode fiber

- A ***connector*** is attached at the end of a cable to enable swift connection and disconnection of the cable to and from a port.
- A Standard connector (SC) (see Figure 5-5 [a]) and a Lucent connector (LC) (see Figure 5-5 [b]) are two commonly used connectors for fiber optic cables. (Both are push and pull connectors with the only difference in their size)
- Straight Tip (ST) is another fiber-optic connector, which is often used with fiber patch panels (see Figure 5.5 [c]).

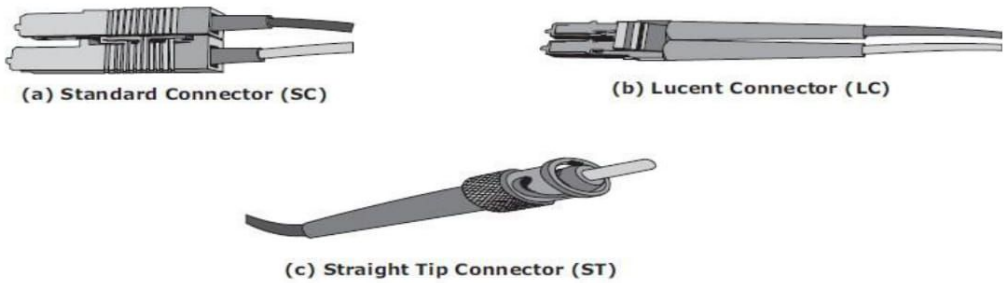


Figure 5-5: SC, LC, and ST connectors

Interconnect Devices

- FC hubs, switches, and directors are the interconnect devices commonly used in FC SAN.
- Hubs are used as communication devices in FC-AL implementations.
- Hubs physically connect nodes in a logical loop or a physical star topology.
- All the nodes must share the loop because data travels through all the connection points.
- Because of the availability of low-cost and high-performance switches, hubs are no longer used in FC SANs
- Switches are more intelligent than hubs and directly route data from one physical port to another.
- Therefore, nodes do not share the bandwidth. Instead, each node has a dedicated communication path.
- Directors are high-end switches with a higher port count and better fault tolerance capabilities.
- A port card or blade has multiple ports for connecting nodes and other FC switches

SAN Management Software

- SAN management software manages the interfaces between hosts, interconnect devices, and storage arrays.
- The software provides a view of the SAN environment and enables management of various resources from one central console.
- It provides key management functions, including mapping of storage devices, switches, and servers, monitoring and generating alerts for discovered devices, and zoning

*** Explain the different types of connectivity in FC Configuration***

5.4 FC Connectivity

Point-to-Point

- Point-to-point is the simplest FC configuration — two devices are connected directly to each other, as shown in Figure 5-6.
- This configuration provides a dedicated connection for data transmission between nodes.
- However, the point-to-point configuration offers limited connectivity, because only two devices can communicate with each other at a given time.
- Moreover, it cannot be scaled to accommodate a large number of nodes.
- Standard DAS uses point-to-point connectivity

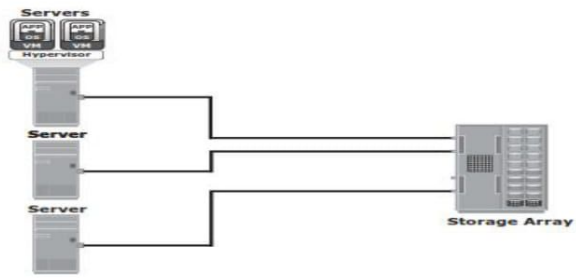


Figure 5-6: Point-to-point connectivity

Fibre Channel Arbitrated Loop

- In the FC-AL configuration, devices are attached to a shared loop.
- FC-AL has the characteristics of a token ring topology and a physical star topology.
- In FC-AL, each device contends with other devices to perform I/O operations.
- Devices on the loop must "arbitrate" to gain control of the loop.
- At any given time, only one device can perform I/O operations on the loop (see Figure 5-7).

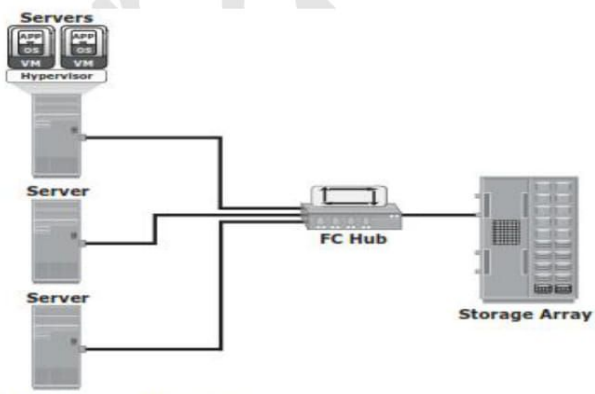


Figure 5-7: Fibre Channel Arbitrated Loop

Fibre Channel Switched Fabric

- FC-SW is also referred to as ***fabric connect***.
- A fabric is a logical space in which all nodes communicate with one another in a network.

- This virtual space can be created with a switch or a network of switches.
- Each switch in a fabric contains a unique domain identifier, which is part of the fabric's addressing scheme.
- In FC-SW, nodes do not share a loop; instead, data is transferred through a dedicated path between the nodes.
- Each port in a fabric has a unique 24-bit Fibre Channel address for communication. Figure 5-8 shows an example of the FC-SW fabric.
- In a switched fabric, the link between any two switches is called an ***Interswitch link (ISL).***
- ISLs enable switches to be connected together to form a single, larger fabric.
- ISLs are used to transfer host-to-storage data and fabric management traffic from one switch to another.
- By using ISLs, a switched fabric can be expanded to connect a large number of nodes.

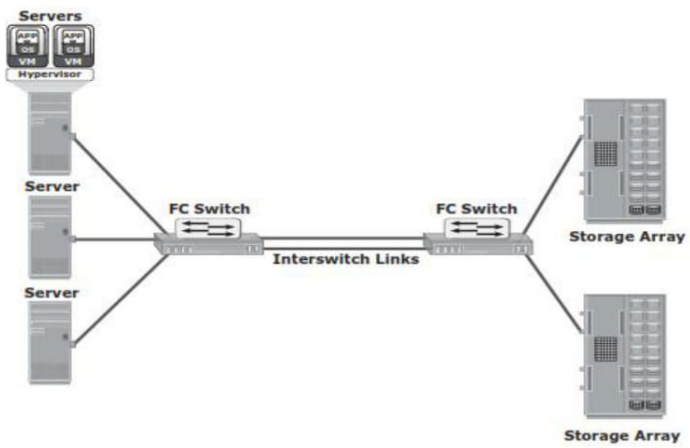


Figure 5-8: Fibre Channel switched fabric

- A fabric can be described by the number of tiers it contains
- When the number of tiers in a fabric increases, the distance that the fabric management traffic must travel to reach each switch also increases.
- This increase in the distance also increases the time taken to propagate and complete a fabric reconfiguration event, such as the addition of a new switch or a zone set propagation event.
- Figure 5-9 illustrates two- tier and three-tier fabric architecture.

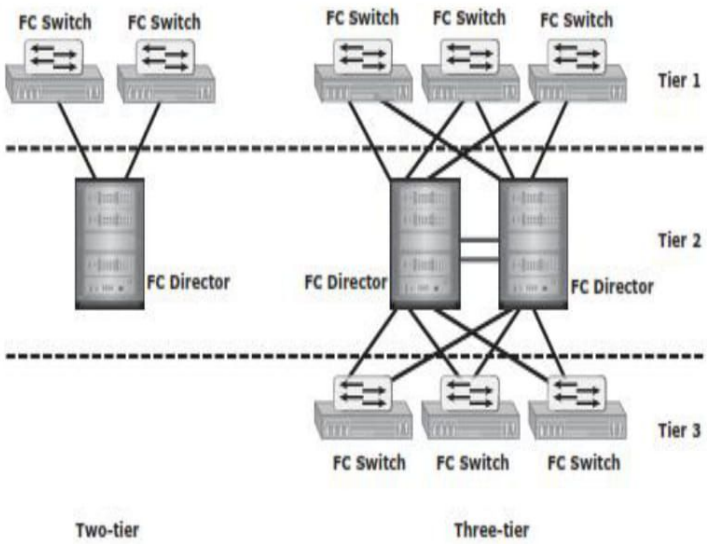


Figure 5-9: Tiered structure of Fibre Channel switched fabric

FC-SW Transmission

- FC-SW uses switches that can switch data traffic between nodes directly through switch ports.
- Frames are routed between source and destination by the fabric.
- As shown in Figure 5-10, if node B wants to communicate with node D, the nodes should individually login first and then transmit data via the FC-SW.
- This link is considered a dedicated connection between the initiator and the target.

** *Different types of Fabric ports***

5.6 Switched Fabric ports

Types

- **N_Port:** An end point in the fabric. This port is also known as the node port. Typically, it is a host port (HBA) or a storage array port connected to a switch in a switched fabric.
- **E_Port:** A port that forms the connection between two FC switches. This port is also known as the expansion port.
  The E_Port on an *FC* switch *con*nects to the E_Port of another FC switch in the fabric through ISLs.
- **F_Port:** A port on a switch that connects an N_Port. It is also known as a fabric port.
- **G_Port:** A generic port on a switch that can operate as an E_Port or an F_Port and determines its functionality automatically during initialization.

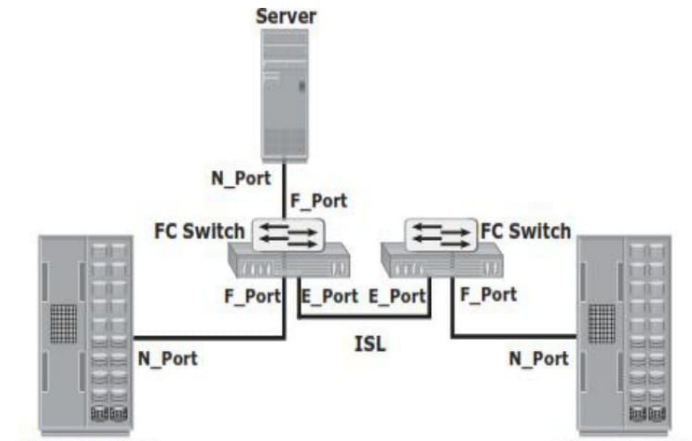


Figure 5-11: Switched fabric ports

5.6 Fibre Channel Architecture

- Traditionally, host computer operating systems have communicated with peripheral devices over channel connections, such as ESCON and SCSI.
- Channel technologies provide high levels of performance with low protocol overheads.
- Such performance is achievable due to the static nature of channels and the high level of hardware and software integration provided by the channel technologies.
- The FC architecture represents true channel/network integration and captures some of the benefits of both channel and network technology.
- FC SAN uses the Fibre Channel Protocol (FCP) that provides both channel speed for data transfer with low protocol overhead and scalability of network technology.

The key advantages of FCP are as follows:

- Sustained transmission bandwidth over long distances.
- Support for a larger number of addressable devices over a network. Theoretically, FC can support more than 15 million device addresses on a network.
- Support speeds up to 16 Gbps (16 GFC).

Explain the different layers in FCP stack with the neat diagram

Fibre Channel Protocol Stack

- It is easier to understand a communication protocol by viewing it as a structure of independent layers.
- FCP defines the communication protocol in five layers: FC-0 through FC-4 (except FC-3 layer, which is not implemented).
- In a layered communication model, the peer layers on each node talk to each other through defined protocols. Figure 5-12 illustrates the Fibre Channel protocol stack.

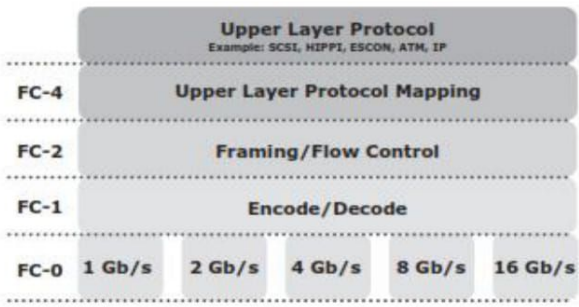


Figure 5-12: Fibre Channel protocol stack

FC-4 Layer

- FC-4 is the uppermost layer in the FCP stack.
- This layer defines the application interfaces and the way Upper Layer Protocols (ULPs) are mapped to the lower FC layers.
- The FC standard defines several protocols that can operate on the FC-4 layer (see Figure 5-12).
- Some of the protocols include SCSI, High Performance Parallel Interface (HIPPI) Framing Protocol, Enterprise Storage Connectivity (ESCON), Asynchronous Transfer Mode (ATM), and IP.

FC-2 Layer

- The FC-2 layer provides Fibre Channel addressing, structure, and organization of data (frames, sequences, and exchanges).
- It also defines fabric services, classes of service, flow control, and routing.

FC-1 Layer

- The FC-1 layer defines how data is encoded prior to transmission and decoded upon receipt.
- At the transmitter node, an 8-bit character is encoded into a 10-bit transmissions character.
- This character is then transmitted to the receiver node.
- At the receiver node, the 10-bit character is passed to the FC-1 layer, which decodes

the 10-bit character into the original 8-bit character.

- FC links with speeds of 10 Gbps and above use 64-bit to 66bit encoding algorithms.
- The FC-1 layer also defines the transmission words, such as FC frame delimiters, which identify the start and end of a frame and primitive signals that indicate events at a transmitting port.
- In addition to these, the FC-1 layer performs link initialization and error recovery.

FC-0 Layer

- FC-0 is the lowest layer in the FCP stack. T
- his layer defines the physical interface, media, and transmission of bits.
- The FC-0 specification includes cables, connectors, and optical and electrical parameters for a variety of data rates.
- The FC transmission can use both electrical and optical media.

Fibre Channel Addressing

- An FC address is dynamically assigned when a node port logs on to the fabric. The FC address has a distinct format, as shown in Figure 5-13.
- The addressing mechanism provided here corresponds to the fabric with the switch as an interconnecting device.

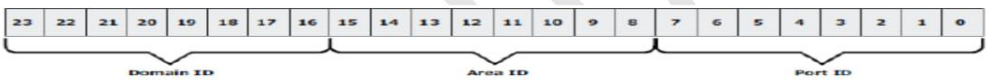


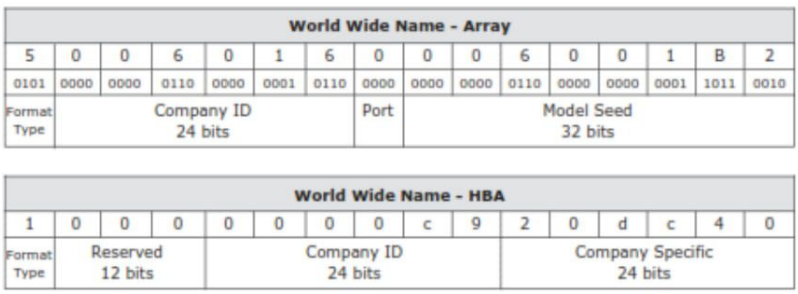
Figure 6-14: 24-bit FC address of N_port

- The first field of the FC address contains the domain ID of the switch.
- A domain ID is a unique number provided to each switch in the fabric.
- Although this is an 8-bit field, there are only 239 available addresses for domain ID because some addresses are deemed special and reserved for fabric management services. For example, FFFFFC is reserved for the name server, and FFFFFE is reserved for the fabric login service.
- The area ID is used to identify a group of switch ports used for connecting nodes.
- An example of a group of ports with a common area ID is a port card on the switch.
- The last field, the port ID, identifies the port within the group.
- Therefore, the maximum possible number of node ports in a switched fabric is calculated as: 239 domains *256 areas *256 ports = 15,663,104

World Wide Names

- Each device in the FC environment is assigned a 64-bit unique identifier called the World-Wide Name (WWN).

- The Fibre Channel environment uses two types of WWNs: ***World Wide Node Name (WWNN) and World-Wide Port Name (WWPN).***
- WWNs are burned into the hardware or assigned through software.
- Unlike an FC address, which is assigned dynamically, a WWN is a static name for each device on an FC network.
- WWNs are similar to the Media Access Control (MAC) addresses used in IP networking
- Several configuration definitions in a SAN use WWN for identifying storage devices and HBAs.
- The name server in an FC environment keeps the association of WWNs to the dynamically created FC addresses for nodes.
- Figure 5-14 illustrates the WWN structure examples for an array and an HBA.



World Wide Name - Array															
5	0	0	6	0	1	6	0	0	0	6	0	0	1	B	2
0101	0000	0000	0110	0000	0001	0110	0000	0000	0000	0110	0000	0000	0001	1011	0010
Format Type	Company ID 24 bits						Port	Model Seed 32 bits							

World Wide Name - HBA															
1	0	0	0	0	0	0	0	c	9	2	0	d	c	4	0
Format Type	Reserved 12 bits			Company ID 24 bits						Company Specific 24 bits					

Figure 5-14: World Wide Names

FC Frame

- An FC frame (Figure 5-15) consists of five parts***: start of frame (SOF), frame header, data field, cyclic redundancy check (CRC), and end of frame (EOF).***

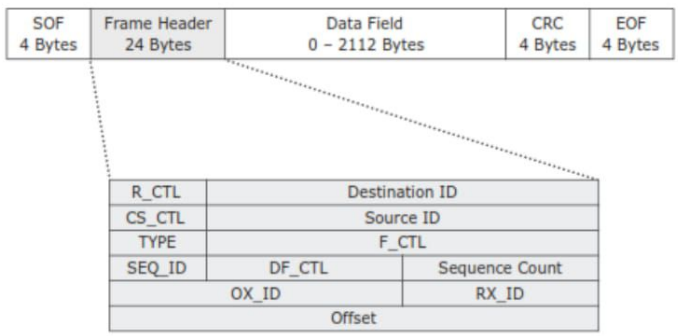


Figure 5-15: FC frame

- The frame header is **24 bytes long and contains addressing information for the frame.**
- It includes the following information: Source ID (S_ID), Destination ID (D_ID), Sequence ID (SEQ_ID), Sequence Count (SEQ_CNT), Originating Exchange ID (OX_ID), and Responder

Exchange ID (RX_ID), in addition to some control fields

- The frame header also defines the following fields:

1. Routing Control (R_CTL): This field denotes whether the frame is a link control frame or a data frame.

Link control frames are nondata frames that do not carry any payload. These frames are used for setup and messaging.

Data frames carry the payload and are used for data transmission.

2. Class Specific Control (CS_CTL): This field specifies link speeds for class 1 and class 4 data transmission.

3. TYPE: This field describes the upper layer protocol (ULP) to be carried on the frame if it is a data frame.

4. Data Field Control (DF_CTL): A 1-byte field that indicates the existence of any optional headers at the beginning of the data payload.

5. Frame Control (F_CTL): A 3-byte field that contains control information related to frame content. [First sequence]

Structure and Organization of FC Data

- A frame represents a word, a sequence represents a sentence, and an exchange represents a conversation.
    1. **Exchange operation:** An exchange operation enables two N_ports to identify and manage a set of information units. This unit maps to a sequence. Sequences can be both unidirectional and bidirectional.
    2. **Sequence:** A sequence refers to a contiguous set of frames that are sent from one port to another.
    3. **Frame:** A frame is the fundamental unit of data transfer at Layer 2. Each frame can contain up to 2,112 bytes of payload.

Flow Control

- Flow control defines the pace of the flow of data frames during data transmission.

1. BB_Credit: FC uses the *BB_Credit* mechanism for hardware-based flow control.

- BB_Credit controls the maximum number of frames that can be present over the link at any given point in time.
- In a switched fabric, BB_Credit management may take place between any two FC ports.
- The BB_Credit mechanism provides frame acknowledgment through the *Receiver Ready (R_RDY)* primitive.

2. EE_Credit: When an initiator and a target establish themselves as nodes communicating with each other, they exchange the EE_Credit parameters.

- Provides flow control class 1 and class 2 traffic

Classes of Service

- Classes of services to meet the requirements of wide range of applications

	CLASS 1	CLASS 2	CLASS 3
Communication type	Dedicated connection	Nondedicated connection	Nondedicated connection
Flow control	End-to-end credit	End-to-end credit B-to-B credit	B-to-B credit
Frame delivery	In order delivery	Order not guaranteed	Order not guaranteed
Frame acknowl-edgement	Acknowledged	Acknowledged	Not acknowledged
Multiplexing	No	Yes	Yes
Bandwidth utilization	Poor	Moderate	High

** *Explain zoning. Which are the different types of zoning***

5.9 Zoning

- Zoning is an FC switch function that enables nodes within the fabric to be logically segmented into groups that can communicate with each other.
- When a device (host or storage array) logs onto a fabric, it is registered with the name server.
- The zoning function controls the process by allowing only the members in the same zone to establish these link-level services.
- Multiple zone sets may be defined in a fabric, but only one zone set can be active at a time.
- A zone set is a set of zones and a zone is a set of members. A member may be in multiple zones. A member may be in multiple zones.

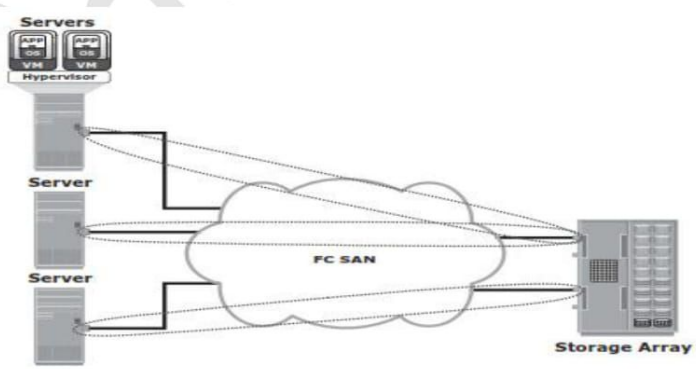
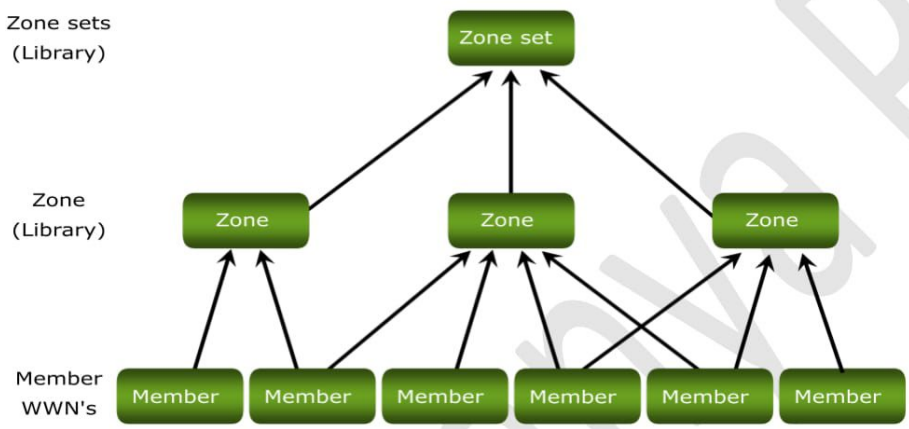


Figure 5-17: Zoning

- Zone members, zones, and zone sets form the hierarchy defined in the zoning process (see Figure 5-18).
- A zone set is composed of a group of zones that can be activated or deactivated as a

single entity in a fabric.

- Multiple zone sets may be defined in a fabric, but only one zone set can be active at a time.
- Members are nodes within the SAN that can be included in a zone.
- ***Switch ports, HBA ports, and storage device ports*** can be members of a zone.
- A port or node can be a member of multiple zones.
- Nodes distributed across multiple switches in a switched fabric may also be grouped into the same zone.
- Zone sets are also referred to as zone configurations.
- Zoning provides control by allowing only the members in the same zone to establish communication with each other.



Types of Zoning

Zoning can be categorized into three types:

1. Port zoning: It uses the FC addresses of the physical ports to define zones. The FC address is dynamically assigned when the port logs on to the fabric. Therefore, any change in the fabric configuration affects zoning.

Port zoning is also called *hard zoning*. Although this method is secure, it requires updating of zoning configuration information in the event of fabric reconfiguration.

2. WWN zoning: It uses World Wide Names to define zones. WWN zoning is also referred to as *soft zoning*.

A major advantage of WWN zoning is its flexibility. It allows the SAN to be recabled without reconfiguring the zone information.

3. Mixed zoning: It combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific port to be tied to the WWN of a node.

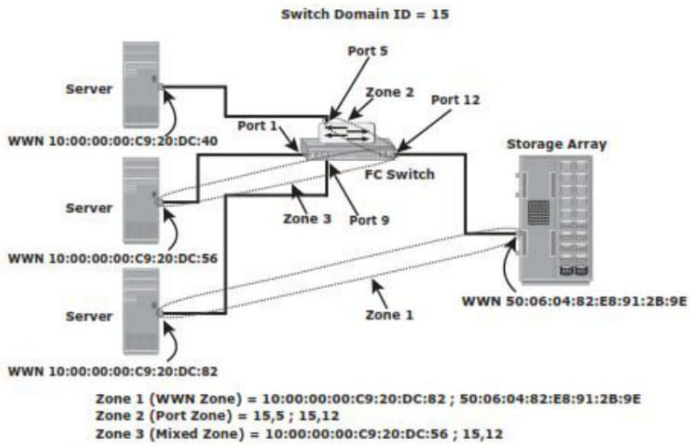


Figure 5-19: Types of zoning

Explain the FC topologies

5.10 FC SAN Topologies

Fabric design follows standard topologies to connect devices. Core-edge fabric is one of the popular topologies for fabric designs. Variations of core-edge fabric and mesh topologies are most commonly deployed in FC SAN implementations

Mesh Topology

- A mesh topology may be one of the two types: full mesh or partial mesh.
- In a full mesh, every switch is connected to every other switch in the topology.
- A full mesh topology may be appropriate when the number of switches involved is small
- A typical deployment would involve up to four switches or directors, with each of them servicing highly localized host-to-storage traffic.
- In a full mesh topology, a maximum of one ISL or hop is required for host-to-storage traffic.
- However, with the increase in the number of switches, the number of switch ports used for ISL also increases.
- This reduces the available switch ports for node connectivity.
- In a partial mesh topology, several hops or ISLs may be required for the traffic to reach its destination.
- Partial mesh offers more scalability than full mesh topology.

- However, without proper placement of host and storage devices, traffic c management in a partial mesh fabric might be complicated and ISLs could become overloaded due to excessive traffic aggregation.

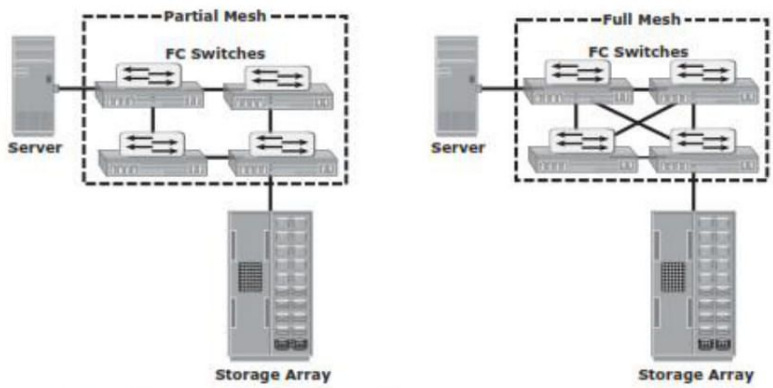


Figure 5-20: Partial mesh and full mesh topologies

Core-Edge Fabric

- The core-edge fabric topology has two types of switch tiers.
- The edge tier is usually composed of switches and offers an inexpensive approach to adding more hosts in a fabric.
- Each switch at the edge tier is attached to a switch at the core tier through ISLs.
- The core tier is usually composed of enterprise directors that ensure high fabric availability.
- In addition, typically all traffic must either traverse this tier or terminate at this tier.
- In this configuration, all storage devices are connected to the core tier, enabling host-to-storage traffic to traverse only one ISL.
- Hosts that require high performance may be connected directly to the core tier and consequently avoid ISL delays.
- In core-edge topology, the edge-tier switches are not connected to each other.
- The core edge fabric topology increases connectivity within the SAN while conserving the overall port utilization.
- If fabric expansion is required, additional edge switches are connected to the core.
- The core of the fabric is also extended by adding more switches or directors at the core tier.
- Based on the number of core-tier switches, this topology has different variations, such as, single-core topology (see Figure 5-21) and dual-core topology (see Figure 5-22).

- To transform a single-core topology to dual-core, new ISLs are created to connect each edge switch to the new core switch in the fabric.

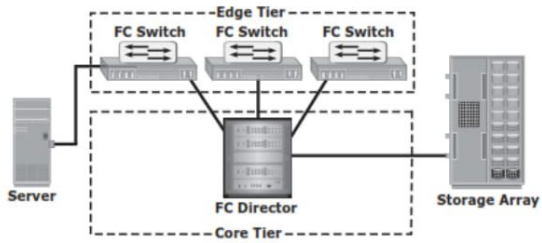


Figure 5-21: Single-core topology

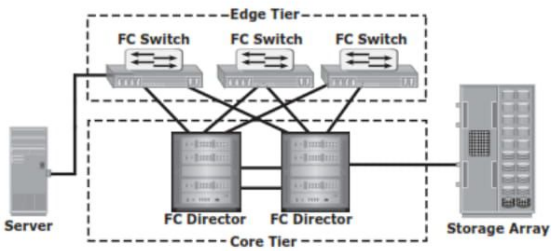


Figure 5-22: Dual-core topology

Benefits and Limitations of Core-Edge Fabric

- Benefits:
- The core-edge fabric provides maximum one-hop storage access to all storage devices in the system.
- Because traffic travels in a deterministic pattern (from the edge to the core and vice versa), a core-edge provides easier calculation of the ISL load and traffic patterns.
- Core-edge fabrics are scaled to larger environments by adding more core switches and linking them, or adding more edge switches.
- This method enables extending the existing simple core-edge model or expanding the fabric into a compound or complex core-edge model.
- Hop count represents the total number of ISLs traversed by a packet between its source and destination.
- A common best practice is to keep the number of host-to-storage hops unchanged, at one hop, in a core-edge.
- Limitations:
- Generally, a large hop count means a high data transmission delay between the source and destination.
- As the number of cores increases, it is prohibitive to continue to maintain ISLs from each core to each edge switch.

- When this happens, the fabric design is changed to a compound or complex core-edge design

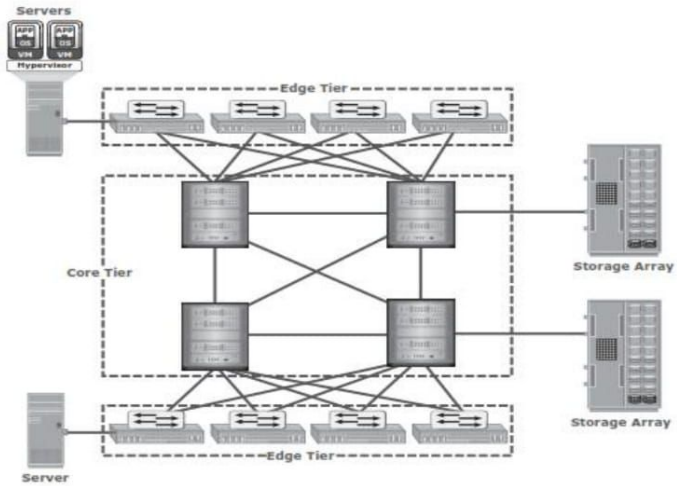


Figure 5-23: Compound core-edge topology

** *Discuss virtualization in SAN***

Virtualization in SAN

- Block-level storage virtualization aggregates block storage devices (LUNs) and enables provisioning of virtual storage volumes, independent of the underlying physical storage.
- A virtualization layer, which exists at the SAN, abstracts the identity of physical storage devices and creates a storage pool from heterogeneous storage devices.
- Virtual volumes are created from the storage pool and assigned to the hosts.
- Instead of being directed to the LUNs on the individual storage arrays, the hosts are directed to the virtual volumes provided by the virtualization layer.
- For hosts and storage arrays, the virtualization layer appears as the target and initiator devices, respectively.
- The virtualization layer maps the virtual volumes to the LUNs on the individual arrays.
- The hosts remain unaware of the mapping operation and access the virtual volumes as if they were accessing the physical storage attached to them.
- Typically, the virtualization layer is managed via a dedicated virtualization appliance to which the hosts and the storage arrays are connected.
- Figure 5 24 illustrates a virtualized environment. It shows two physical servers, each of which has one virtual volume assigned.
- Previously, block-level storage virtualization provided nondisruptive data migration only within a data center.
- The new generation of block-level storage virtualization enables non disruptive data

migration both within and between data centers.

- It provides the capability to connect the virtualization layers at multiple data centers.
- The connected virtualization layers are managed centrally and work as a single virtualization layer stretched across data centers (see Figure 5-25).
- This enables the federation of block storage resources both within and across data centers. The virtual volumes are created from the federated storage resources.

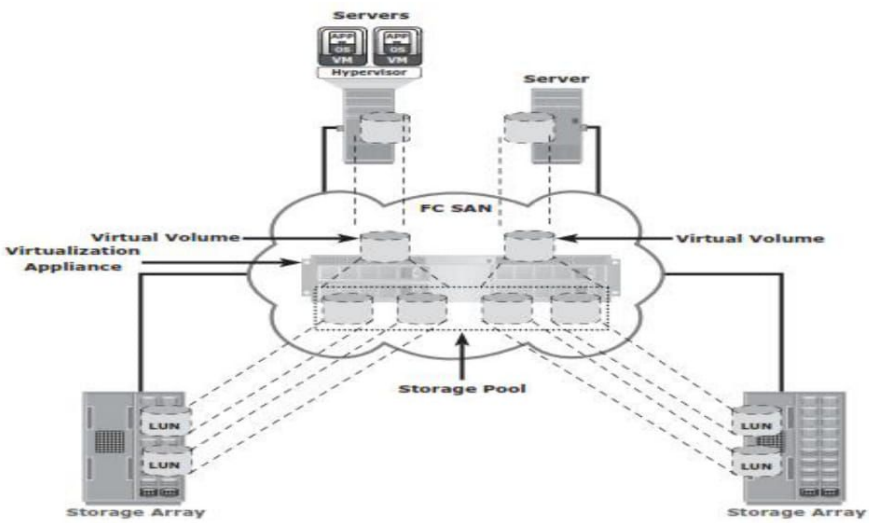


Figure 5-24: Block-level storage virtualization

Virtual SAN (VSAN)

- Virtual SAN (also called virtual fabric) is a logical fabric on an FC SAN, which enables communication among a group of nodes regardless of their physical location in the fabric.
- In a VSAN, a group of hosts or storage ports communicate with each other using a virtual topology defined on the physical SAN.
- Multiple VSANs may be created on a single physical SAN. Each VSAN acts as an independent fabric with its own set of fabric services, such as name server, and zoning.
- Fabric-related configurations in one VSAN do not affect the traffic in another.

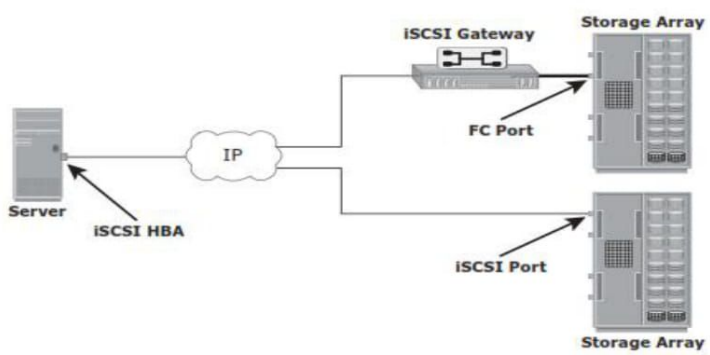


Figure 6-1: iSCSI implementation

Chapter 6: IP SAN and FCoE

6.1 iSCSI

iSCSIis an IP based protocol that establishes and manages connections between host and storage over IP, as shown in Figure 6-1. 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.

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

- A standard NIC with software iSCSI initiator, a TCP offload engine (TOE) NIC with software iSCSI initiator, and an iSCSI HBA are the three iSCSI host connectivity options.
- 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 alleviate 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.

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 Figure 6-2 (a), the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernet switch.

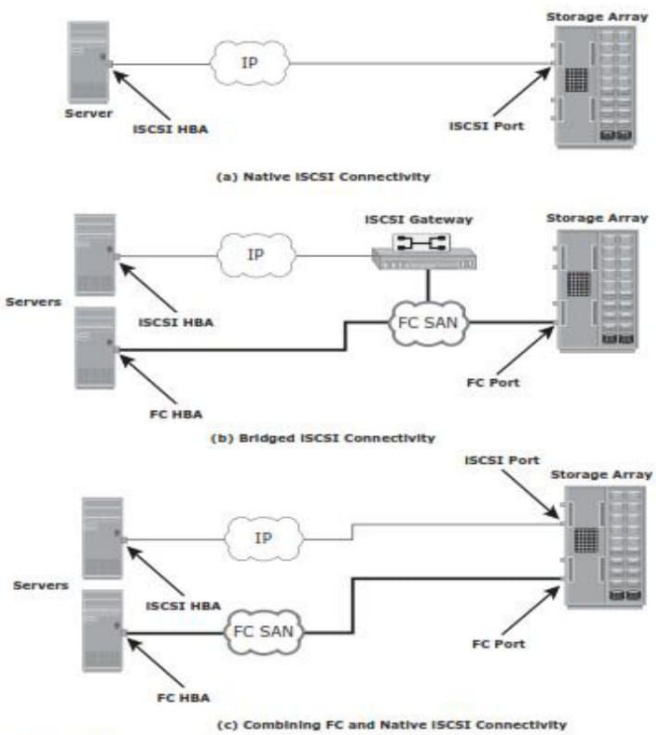


Figure 6-2: iSCSI Topologies

Bridged iSCSI Connectivity

A bridged iSCSI implementation includes FC components in its configuration. Figure 6-2 (b) illustrates iSCSI host connectivity to an FC storage array.

- In this case the array doesnot 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.

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 Figure 6-2 (c).

iSCSI Protocol Stack

Figure 6-3 displays a model of the iSCSI protocol layers and depicts the encapsulation order of the SCSI commands for their delivery through a physical carrier.

- 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.

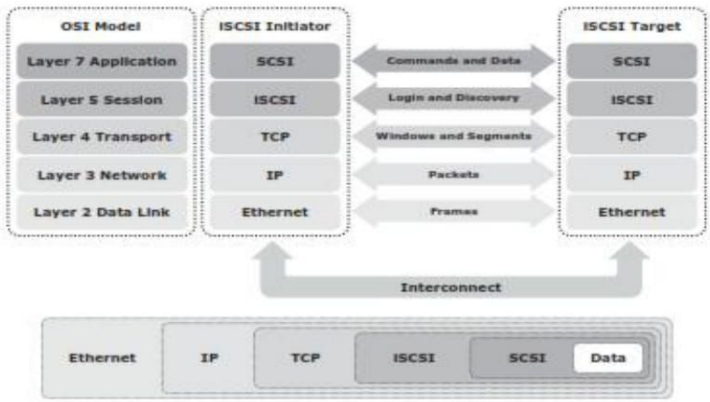


Figure 6-3: iSCSI protocol stack

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 Figure 6-4.
- 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.
- As shown in Figure 6-5, each iSCSI PDU does not correspond in a 1:1 relationship with an IP packet.
- Depending on its size, an iSCSI PDU can span an IP packet or even coexist with another PDU in the same packet.

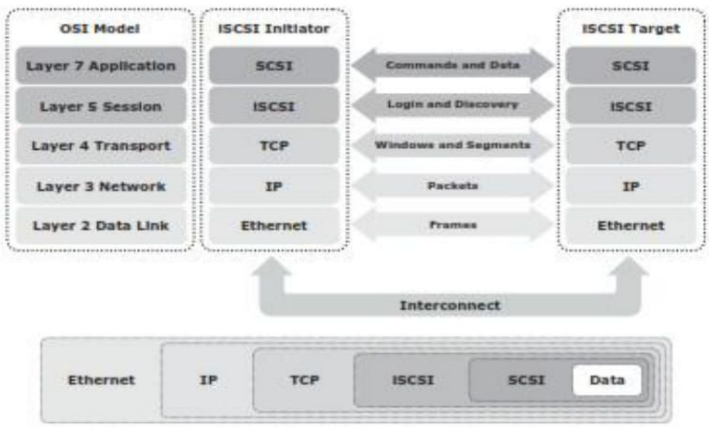


Figure 6-3: iSCSI protocol stack

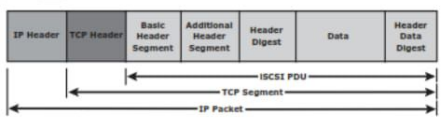


Figure 6-4: iSCSI PDU encapsulated in an IP packet

A message transmitted on a network is divided into a number of packets. If necessary, each packet can be sent by a different route across the network. Packets can arrive in a different order than the order in which they were sent. IP only delivers them; it is up to TCP to organize them in the right sequence. The target extracts the SCSI commands and data on the basis of the information in the iSCSI header.

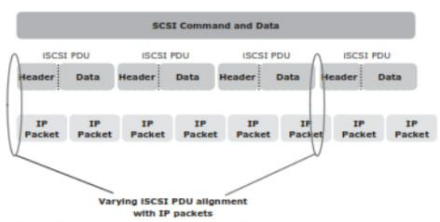


Figure 6-5: Alignment of iSCSI PDUs with IP packets

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: Send Targets discovery or internet Storage Name Service (iSNS).

- iSNS (see Figure 6-6) 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 know the targets that it can access, it can query the iSNS server for a list of available targets.

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.

Following are two types of iSCSI names commonly used:

- iSCSI Qualified Name (IQN): An organization must own a registered domain name to generate iSCSI Qualified 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 other device identifiers. An iSCSI Qualified Name enables storage administrators to assign meaningful names to iSCSI devices, and therefore, manage those devices more easily.
- Extended Unique Identifier (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 as eui.0300732A32598D26.

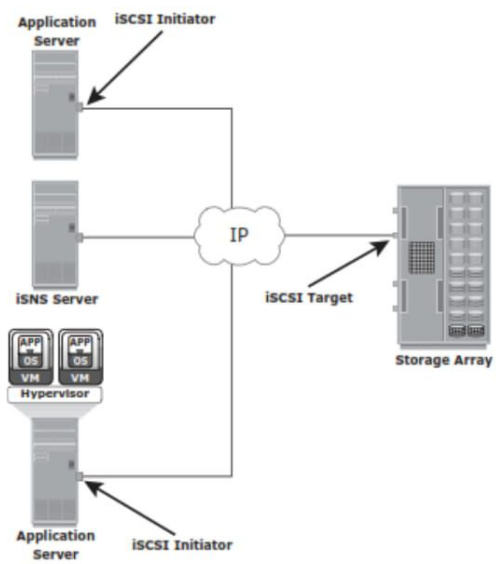


Figure 6-6: Discovery using iSNS

iSCSI Session

An iSCSI session is established between an initiator and a target, as shown in Figure 6-7. 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)

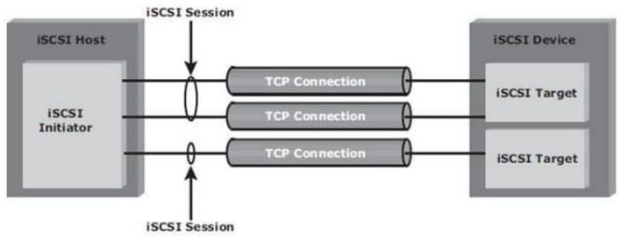


Figure 6-7: iSCSI session

iSCSI 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.

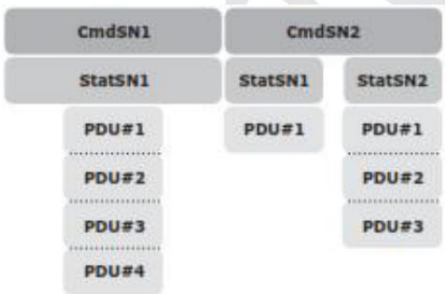


Figure 6-8: Command and status sequence number

6.2 FCIP

FC SAN provides a high-performance infrastructure for localized data movement. Organizations are now looking for ways to transport data over a long distance between their disparate SANs at multiple geographic locations. One of the best ways to achieve this goal is to interconnect geographically dispersed SANs through reliable, high- speed links.

FCIP Protocol Stack

The FCIP protocol stack is shown in Figure: 6-9. Applications generate SCSI commands and data, which are processed by various layers of the protocol stack.

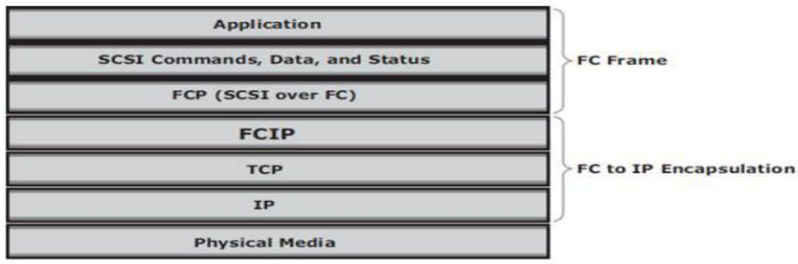


Figure 6-9: FCIP protocol stack

The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Figure 6-10). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IP traffic.

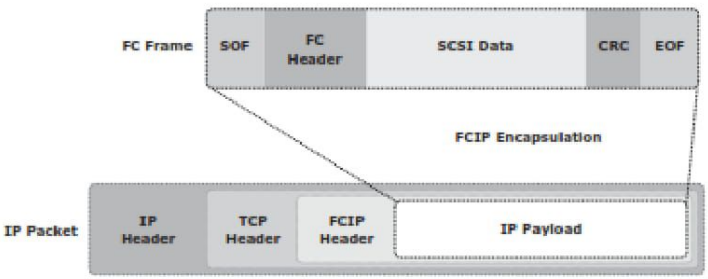


Figure 6-10: FCIP encapsulation

FCIP Topology

In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (see Figure 6-11). 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.

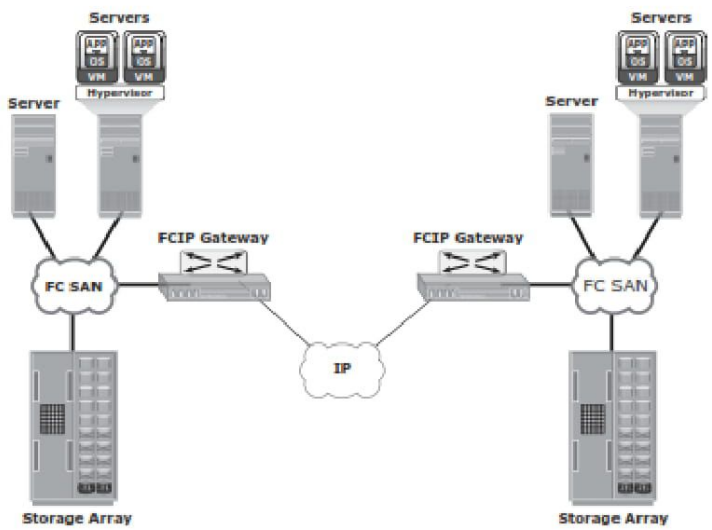


Figure 6-11: FCIP topology

FCIP Performance and Security

Performance, reliability, and security should always be taken into consideration when implementing storage solutions. The implementation of FCIP is also subject to the same considerations.

- Configuring multiple paths between FCIP gateways eliminates single point of failure and provides increased bandwidth
- In the case of extended distance the IP network might be a bottleneck is sufficient bandwidth is not available
- Because FCIP creates a unified fabric disruption in the underlying network can cause instabilities in the SAN environment.

6.3 FCoE

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.

Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffi c over a single physical interface infrastructure. FCoE helps organizations address the challenges of having multiple discrete network infrastructures.

I/O Consolidation Using FCoE

The key benefit of FCoE is I/O consolidation. Figure 6-12 represents the infrastructure before FCoE deployment. Here, the storage resources are accessed using HBAs, and the IP network resources are accessed using NICs by the servers. Typically, in a data center, a server is configured with 2 to 4 NIC cards and redundant HBA cards. If the data center has hundreds of servers, it would require a large number of adapters, cables, and switches. This leads to a complex environment, which is difficult to manage and scale

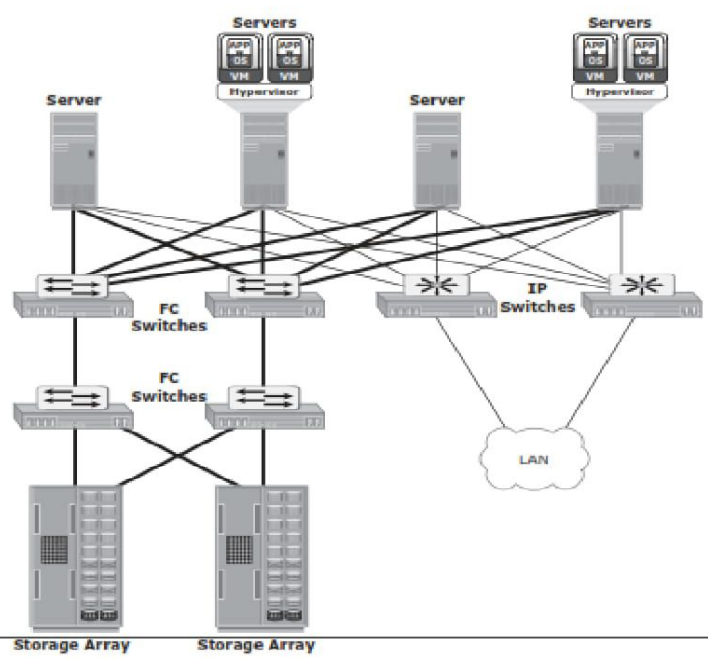


Figure 6-12: Infrastructure before using FCoE

Figure 6-13 shows the I/O consolidation with FCoE using FCoE **switches and Converged Network Adapters (CNAs).** A CNA replaces both HBAs and NICs in the server and consolidates both the IP and FC traffic. This reduces the requirement of multiple network adapters at the server to connect to different networks. Overall, this reduces the requirement of adapters, cables, and switches. This also considerably reduces the cost and management overhead.

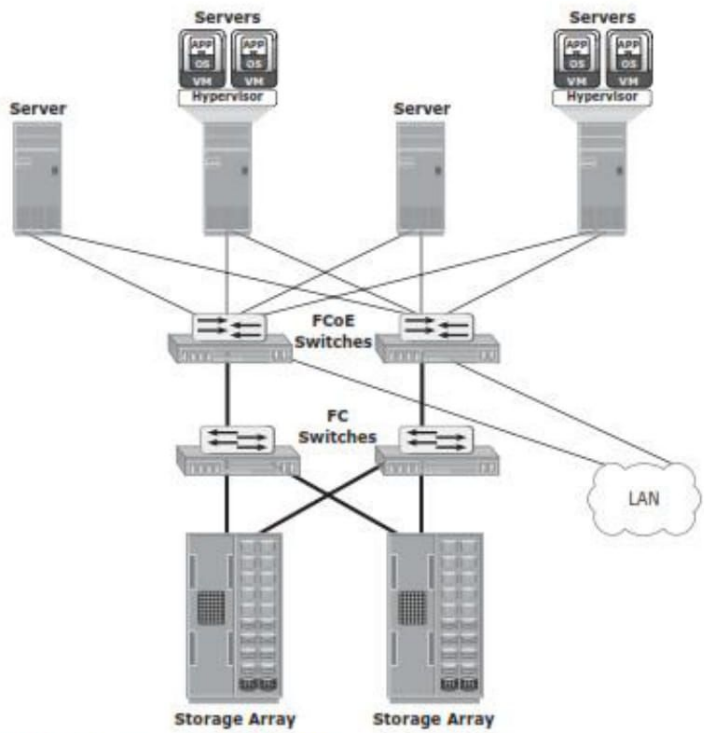


Figure 6-13: Infrastructure after using FCoE

Components of an FCoE Network

- Converged Network Adapter (CNA)
- Cables
- FCoE switches

Converged Network Adapter

- 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 Figure 6-14, 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.

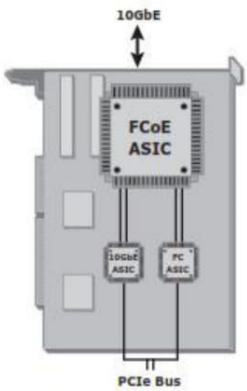


Figure 6-14: Converged Network Adapter

Cables

Currently two options are available for FCoE cabling: Copper based Twinax and 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 fiber optic cables.

FCoE Switches

An FCoE switch has both Ethernet switch and Fibre Channel switch functionalities.

- The FCoE switch has a Fibre Channel Forwarder (FCF), Ethernet Bridge, and set of Ethernet ports and optional FC ports, as shown in Figure 6-15.
- 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.

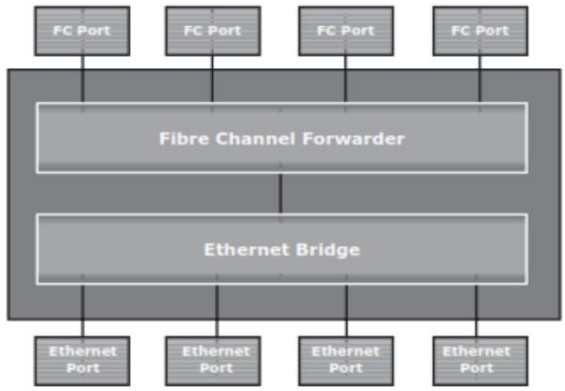


Figure 6-15: FCoE switch generic architecture

- 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 traffi c as usual Ethernet traffi c and forwards it over the Ethernet ports.

FCoE Frame Structure

- An FCoE frame is an Ethernet frame that contains an FCoE Protocol Data Unit. Figure 6- 16 shows the FCoE frame structure.
- The first 48-bits in the frame are used to specify the destination MAC address, and the next 48-bits specify the source MAC address.
- The 32- bit IEEE 802.1Q tag supports the creation of multiple virtual networks (VLANs) across a single physical infrastructure.
- FCoE has its own Ethertype, as designated by the next 16 bits, followed by the 4-bit version field.
- The next 100-bits are reserved and are followed by the 8-bit Start of Frame and then the actual FC frame.
- The 8-bit End of Frame delimiter is followed by 24 reserved bits. The frame ends with the final 32-bits dedicated to the Frame Check Sequence (FCS) function that provides error detection for the Ethernet frame.

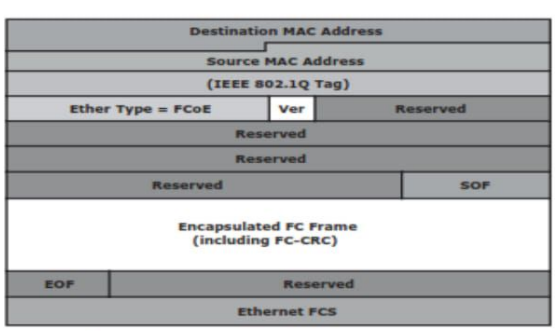


Figure 6-16: FCoE frame structure

FCoE Frame Mapping

The encapsulation of the Fibre Channel frame occurs through the mapping of the FC frames onto Ethernet, as shown in Figure 6-17. Fibre Channel and traditional networks have stacks of layers where each layer in the stack represents a set of functionalities. The FC stack consists of fi ve layers: FC-0 through FC-4. Ethernet is typically considered as a set of protocols that operates at the physical and data link layers in the seven layer OSI stack. The FCoE protocol specification replaces the FC-0 and FC-1 layers of the FC stack with Ethernet.

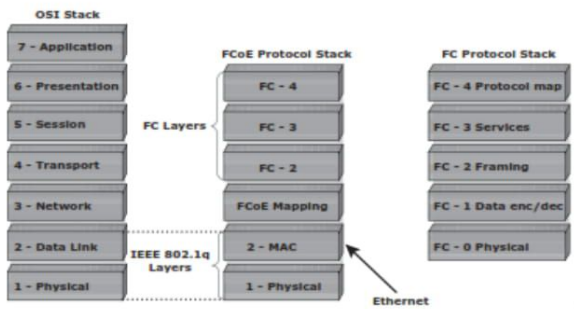


Figure 6-17: FCoE frame mapping

FCoE Enabling Technologies

- Conventional Ethernet is lossy in nature, which means that frames might be dropped or lost during transmission.
- Converged Enhanced Ethernet (CEE), or lossless Ethernet, provides a new specification to the existing Ethernet standard that eliminates the lossy nature of Ethernet.
- This makes 10 Gb Ethernet a viable storage networking option, similar to FC.
- Lossless Ethernet requires certain functionalities, they are:
    - Priority-based flow control
    - Enhanced transmission selection
    - Congestion Notification
    - Data center bridging exchange protocol

Priority-Based Flow Control (PFC)

- PFC provides a link level flow control mechanism.
- PFC creates eight separate virtual links on a single physical link and allows any of these links to be paused and restarted independently.
- PFC enables the pause mechanism based on user priorities or classes of service.
- Enabling the pause based on priority allows creating lossless links for traffic, such as FCoE traffic.
- This PAUSE mechanism is typically implemented for FCoE while regular TCP/IP traffic continues to drop frames.
- Figure 6-18 illustrates how a physical Ethernet link is divided into eight virtual links and

allows a PAUSE for a single virtual link without affecting the traffic for the others.

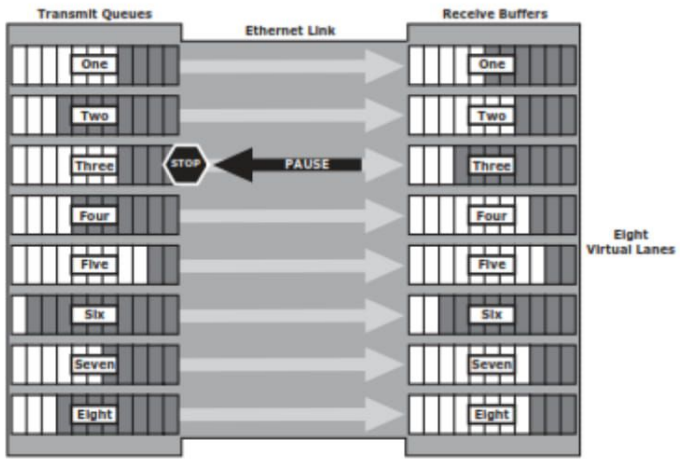


Figure 6-18: Priority-based flow control

Enhanced Transmission Selection (ETS)

- Enhanced transmission selection provides a common management framework for the assignment of bandwidth to different traffic classes, such as LAN, SAN, and Inter Process Communication (IPC).
- When a particular class of traffic does not use its allocated bandwidth, ETS enables other traffic classes to use the available bandwidth.

Congestion Notification (CN)

- Congestion notification provides end-to-end congestion management for protocols, such as FCoE, that do not have built-in congestion control mechanisms.
- Link level congestion notification provides a mechanism for detecting congestion and notifying the source to move the traffic flow away from the congested links.
- Link level congestion notification enables a switch to send a signal to other ports that need to stop or slow down their transmissions.
- The process of congestion notification and its management is shown in Figure 6-19, which represents the communication between the nodes A (sender) and B (receiver).

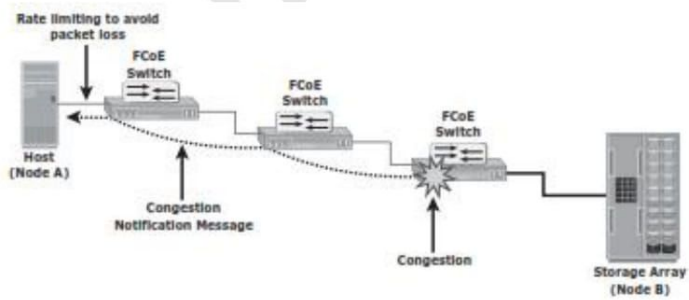


Figure 6-19: Congestion Notification

Chapter 7: Network-Attached Storage

Network-attached storage (NAS) is a dedicated, high-performance file sharing and storage device. NAS enables its clients to share files over an IP network.

- NAS provides the advantages of server consolidation by eliminating the need for multiple file servers.
- It also consolidates the storage used by the clients onto a single system, making it easier to manage the storage.
- A 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 and, therefore, performs file I/O better than a general-purpose server.
- As a result, a NAS device can serve more clients than general-purpose servers and provide the benefit of server consolidation.

7.4 Components of NAS

A NAS device has two key components: NAS head and storage (see Figure 7-3). In some NAS implementations, the storage could be external to the NAS device and shared with other hosts. The NAS head includes the following components:

- CPU and memory
- One or more network interface cards (NICs), which provide connectivity to the client network. Examples of network protocols supported by NIC include Gigabit Ethernet, Fast Ethernet, ATM, and Fiber Distributed Data Interface (FDDI).
- 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

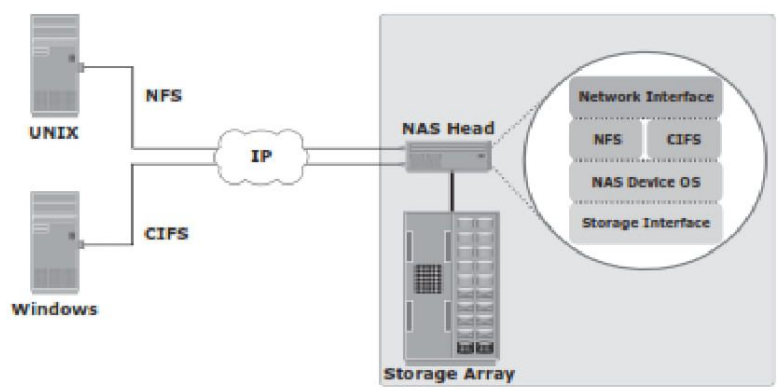


Figure 7-3: Components of NAS

7.5 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. For example, 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. Figure 7-4 illustrates this process.

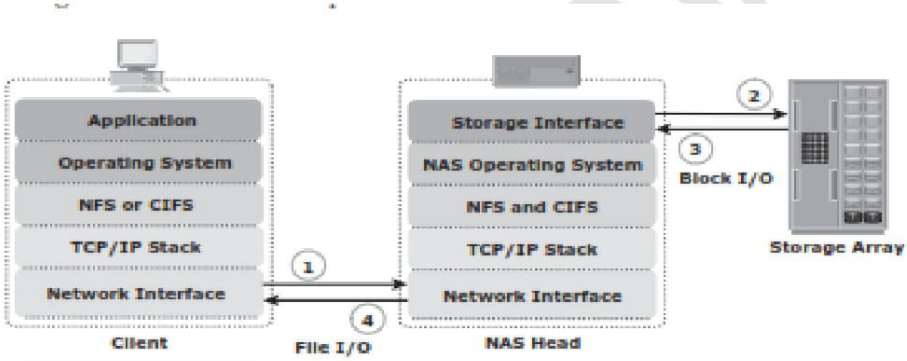


Figure 7-4: NAS I/O operation

7.7 NAS File-Sharing Protocols

Most NAS devices support multiple file-service protocols to handle file I/O requests to a remote file system. ***NFS and CIFS are the common protocols for file sharing***. NAS devices enable users to share file data across different operating environments and provide a means for users to migrate transparently from one operating system to another.

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 a machine-independent model to represent user data.

- It also 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 fi le 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.
- NFS (NFSv3 and earlier) is a *stateless protocol*, which means that it does not maintain any kind of table to store information about open fi les and associated pointers.
- Therefore, each call provides a full set of arguments 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):*** The most commonly used version, which 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 fi le 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.

CIFS

CIFS is a client-server application protocol that enables client programs to make requests for files and services on remote computers over TCP/IP. It is a public, or open, variation of Server Message Block (SMB) protocol.

- 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. The fault tolerance features of CIFS depend on whether an application is written to take advantage of these features.

7.9 File-Level Virtualization

- File-level virtualization eliminates the dependencies between the data accessed at the file level and the location where the files are physically stored.
- Implementation of file-level virtualization is common in NAS or file-server environments.

- It provides non-disruptive file mobility to optimize storage utilization.
- Before virtualization, each host knows exactly where its file resources are located.
- This environment leads to underutilized storage resources and capacity problems because files are bound to a specific NAS device or file server.
- It may be required to move the files from one server to another because of performance reasons or when the file server fills up.
- Moving files across the environment is not easy and may make files inaccessible during file movement.
- Moreover, hosts and applications need to be reconfigured to access the file at the new location.
- This makes it difficult for storage administrators to improve storage efficiency while maintaining the required service level.
- File-level virtualization simplifies file mobility.
- It provides user or application independence from the location where the files are stored. File-level virtualization creates a logical pool of storage, enabling users to use a logical path, rather than a physical path, to access files.
- File-level virtualization facilitates the movement of files across the online file servers or NAS devices.
- This means that while the files are being moved, clients can access their files non-disruptively.
- Clients can also read their files from the old location and write them back to the new location without realizing that the physical location has changed.
- A global namespace is used to map the logical path of a file to the physical path names. Figure 7-9 illustrates a file-serving environment before and after the implementation of file-level virtualization.

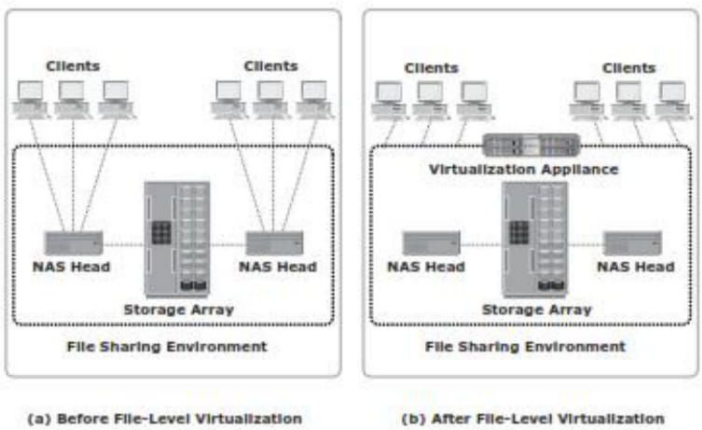


Figure 7-9: File-serving environment before and after file-level virtualization

Chapter 8: Object-Based and Unified Storage

Object-based storage is a way to store file data in the form of objects based on its content and other attributes rather than the name and location.

8.1 Object-Based Storage Devices

- An OSD is a device that organizes and stores unstructured data, such as movies, office documents, and graphics, as objects.
- Object-based storage provides a scalable, self- managed, protected, and shared storage option.
- OSD stores data in the form of objects.
- OSD uses flat address space to store data. Therefore there is no hierarchy of directories and files; as a result, a large number of objects can be stored in an OSD system (see Figure 8-1).

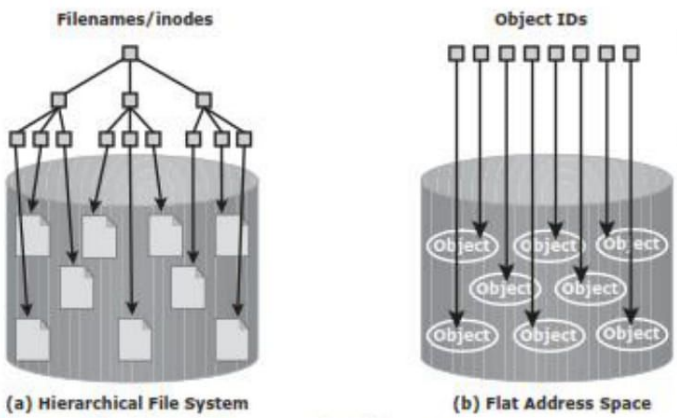


Figure 8-1: Hierarchical file system versus flat address space

- An object might contain user data, related metadata (size, date, ownership, and so on), and other attributes of data (retention, access pattern, and so on); see Figure 8-2.
- Each object stored in the system is identified by a unique ID called the object ID.
- The object ID is generated using specialized algorithms such as hash function on the data and guarantees that every object is uniquely identified.

Figure 8-2: Object structure

Object-Based Storage Architecture

- An I/O in the traditional block access method passes through various layers in the I/O path.
- The I/O generated by an application passes through the file system, the channel, or network and reaches the disk drive.
- When the file system receives the I/O from an application, the file system maps the incoming I/O to the disk blocks.
- The block interface is used for sending the I/O over the channel or network to the storage device.
- The I/O is then written to the block allocated on the disk drive. Figure 8-3 (a) illustrates the block- level access.

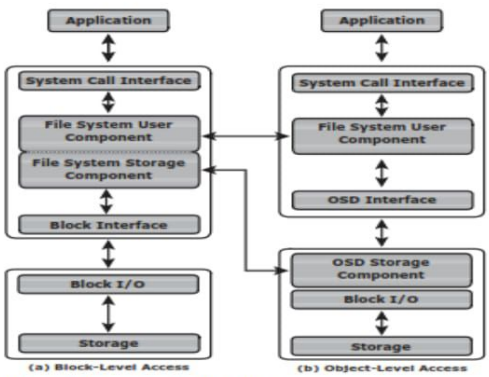


Figure 8-3: Block-level access versus object-level access

- The file system has two components: user component and storage component.
- The user component of the file system performs functions such as hierarchy management, naming, and user access control.
- The storage component maps the files to the physical location on the disk drive.
- When an application accesses data stored in OSD, the request is sent to the file system user component.
- The file system user component communicates to the OSD interface, which in turn sends the request to the storage device.
- The storage device has the OSD storage component responsible for managing the access to the object on a storage device. Figure 8-3 (b) illustrates the object-level access.
- After the object is stored, the OSD sends an acknowledgment to the application server.

Components of OSD

The OSD system is typically composed of three key components:

- Nodes
- private network and
- storage

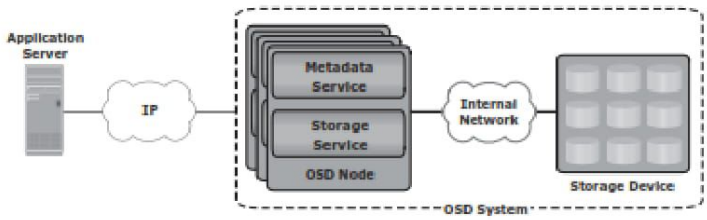


Figure 8-4: OSD components

- The OSD system is composed of one or more *nodes*.
- A node is a server that runs the OSD operating environment and provides services to store, retrieve, and manage data in the system.
- The OSD node has two key services: metadata service and storage service.
- The metadata service is responsible for generating the object ID from the contents (and can also include other attributes of data) of a file.
- It also maintains the mapping of the object IDs and the file system namespace.
- The storage service manages a set of disks on which the user data is stored.
- The OSD nodes connect to the storage via an internal network.
- The internal network provides node-to-node connectivity and node-to-storage connectivity.

Object Storage and Retrieval in OSD

The process of storing objects in OSD is illustrated in Figure 8-5. The data storage process in an OSD system is as follows:

1. The application server presents the file to be stored to the OSD node.
2. The OSD node divides the file into two parts: user data and metadata.
3. The OSD node generates the object ID using a specialized algorithm. The algorithm is executed against the contents of the user data to derive an ID unique to this data.
4. For future access, the OSD node stores the metadata and object ID using the metadata service.
5. The OSD node stores the user data (objects) in the storage device using the storage service.
6. An acknowledgment is sent to the application server stating that the object is stored

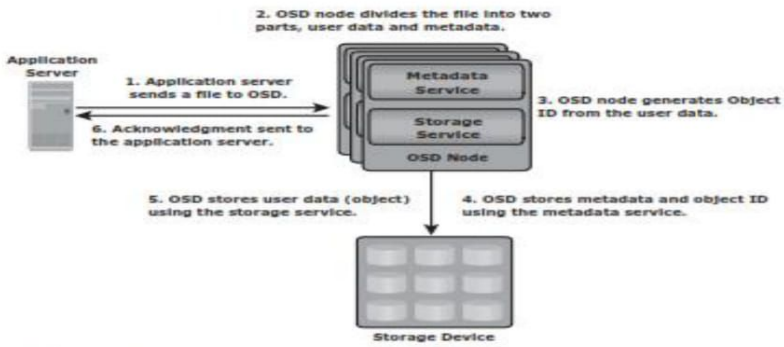


Figure 8-5: Storing objects on OSD

The process of retrieving objects in OSD is illustrated in Figures 8-6. The process of data retrieval from OSD is as follows:

1. The application server sends a read request to the OSD system.
2. The metadata service retrieves the object ID for the requested file.
3. The metadata service sends the object ID to the application server.
4. The application server sends the object ID to the OSD storage service for object retrieval.
5. The OSD storage service retrieves the object from the storage device.
6. The OSD storage service sends the file to the application server.

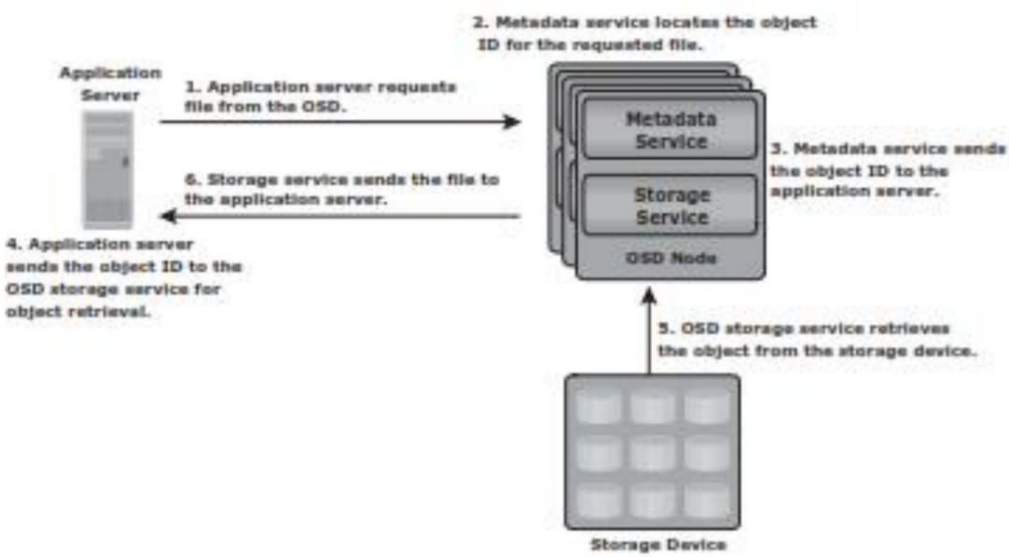


Figure 8-6: Object retrieval from an OSD system

Benefits of Object-Based Storage

The key benefits of object-based storage are as follows:

- Security and reliability: Data integrity and content authenticity are the key features of object-based storage devices. OSD uses specialized algorithms to create objects that provide strong data encryption capability. In OSD, request authentication is performed at the storage device rather than with an external authentication mechanism.

- Platform independence: Objects are abstract containers of data, including metadata and attributes. This feature allows objects to be shared across heterogeneous platforms locally or remotely. This platform-independence capability makes object-based storage the best candidate for cloud computing environments.
- Scalability: Due to the use of flat address space, object-based storage can handle large amounts of data without impacting performance. Both storage and OSD nodes can be scaled independently in terms of performance and capacity.
- Manageability: Object-based storage has an inherent intelligence to manage and protect objects. It uses self-healing capability to protect and replicate objects. Policy-based management capability helps OSD to handle routine jobs automatically.

Common Use Cases for Object-Based Storage

- A **data archival solution** is a promising use case for OSD. Data integrity and protection is the primary requirement for any data archiving solution. Traditional archival solutions — CD and DVD-ROM — do not provide scalability and performance. OSD stores data in the form of objects, associates them with a unique object ID, and ensures high data integrity. Along with integrity, it provides scalability and data protection. These capabilities make OSD a viable option for long term data archiving for fixed content.
- **Content addressed storage (CAS)** is a special type of object-based storage device purposely built for storing fixed content.
- Another use case for OSD is **cloud-based storage. OSD** uses a web interface to access storage resources. OSD provides inherent security, scalability, and automated data management. It also enables data sharing across heterogeneous platforms or tenants while ensuring integrity of data. These capabilities make OSD a strong option for cloud-based storage. Cloud service providers can leverage OSD to offer storage-as-a-service.

OSD supports web service access via representational state transfer (REST) and simple object access protocol (SOAP).

8.2 Content-Addressed Storage

CAS is an object-based storage device designed for secure online storage and retrieval of fixed content. CAS stores user data and its attributes as an object. The stored object is assigned a globally unique address, known as a content address (CA).

CAS provides all the features required for storing fixed content. The key features of CAS are as follows:

- Content authenticity: It assures the genuineness of stored content. This is achieved by generating a unique content address for each object and validating the content address for stored objects at regular intervals. Content authenticity is assured because the address assigned to each object is as unique as a fingerprint. Every time

an object is read, CAS uses a hashing algorithm to recalculate the object's content address as a validation step and compares the result to its original content address. If the object fails validation, CAS rebuilds the object using a mirror or parity protection scheme.

- Content integrity: It provides assurance that the stored content has not been altered. CAS uses a hashing algorithm for content authenticity and integrity. If the fixed content is altered, CAS generates a new address for the altered content, rather than overwrite the original fixed content.
- Location independence: CAS uses a unique content address, rather than directory path names or URLs, to retrieve data. This makes the physical location of the stored data irrelevant to the application that requests the data.

  Single-instance storage (SIS): CAS uses a unique content address to guarantee the storage of only a single instance of an object. When a new object is written, the CAS system is polled to see whether an object is already available with the same content address. If the object is available in the system, it is not stored; instead, only a pointer to that object is created.
- Retention enforcement: Protecting and retaining objects is a core requirement of an archive storage system. After an object is stored in the CAS system and the retention policy is defined, CAS does not make the object available for deletion until the policy expires.
- Data protection: CAS ensures that the content stored on the CAS system is available even if a disk or a node fails. CAS provides both local and remote protection to the data objects stored on it.
- Fast record retrieval: CAS stores all objects on disks, which provides faster access to the objects compared to tapes and optical discs.
- Load balancing: CAS distributes objects across multiple nodes to provide maximum throughput and availability.
- Scalability: CAS allows the addition of more nodes to the cluster without any interruption to data access and with minimum administrative overhead.
- Event notification: CAS continuously monitors the state of the system and raises an alert for any event that requires the administrator's attention. The event notification is communicated to the administrator through SNMP, SMTP, or e-mail.
- Self-diagnosis and repair: CAS automatically detect and repairs corrupted objects and alerts the administrator about the potential problem. CAS systems can be configured to alert remote support teams who can diagnose and repair the system remotely.
- Audit trails: CAS keeps track of management activities and any access or disposition of data. Audit trails are mandated by compliance requirements.

8.4 Unified Storage

Unified storage consolidates block, file, and object access into one storage solution. It supports multiple protocols, such as CIFS, NFS, iSCSI, FC, FCoE, REST (representational state transfer), and SOAP (simple object access protocol).

Components of Unified Storage

A unified storage system consists of the following key components: storage controller, NAS head, OSD node, and storage. Figure 8-9 illustrates the block diagram of a unified storage platform.

- **The storage controller** provides block-level access to application servers through iSCSI, FC, or FCoE protocols.
- It contains iSCSI, FC, and FCoE front-end ports for direct block access.
- The storage controller is also responsible for managing the back-end storage pool in the storage system.
- The controller configures LUNs and presents them to application servers, NAS heads, and OSD nodes.
- A **NAS head** is a dedicated file server that provides file access to NAS clients.
- A *NAS head* is a dedicated fi le server that provides file access to NAS clients.
- The NAS head is connected to the storage via the storage controller typically using a FC or FCoE connection.
- The system typically has two or more NAS heads for redundancy
- The **OSD node** accesses the storage through the storage controller using a FCor FCoE connection.

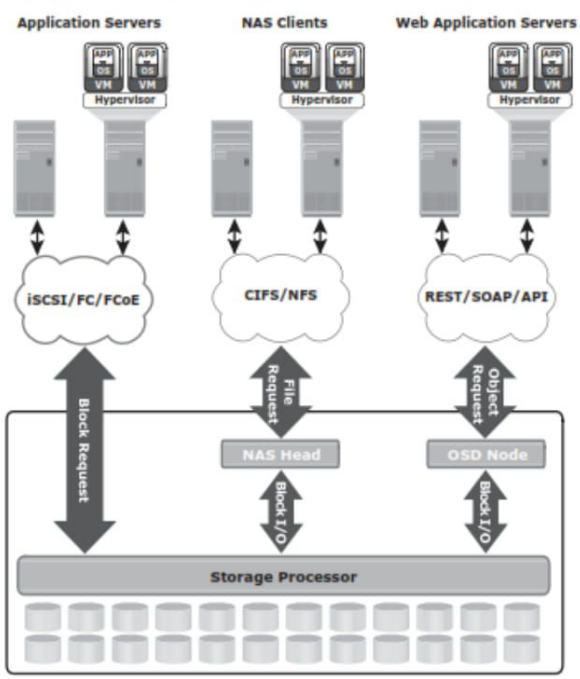


Figure 8-9: Unified storage platform

Data Access from Unified Storage

In a unified storage system, block, fi le, and object requests to the storage travel through different I/O paths. Figure 8-9 illustrates the different I/O paths for block, file, and object access.

- ***Block I/O request:*** The application servers are connected to an FC, iSCSI, or FCoE port on the storage controller. The server sends a block request over an FC, iSCSI, or FCoE connection. The storage processor (SP) processes the I/O and responds to the application server.
- ***File I/O request:*** The NAS clients (where the NAS share is mounted or mapped) send a file request to the NAS head using the NFS or CIFS protocol. The NAS head receives the request, converts it into a block request, and forwards it to the storage controller. Upon receiving the block data from the storage controller, the NAS head again converts the block request back to the file request and sends it to the clients.
- ***Object I/O request:*** The web application servers send an object request, typically using REST or SOAP protocols, to the OSD node. The OSD node receives the request, converts it into a block request, and sends it to the disk through the storage controller.

****