Fibre Channel Storage Area

Components of SAN

- A SAN consists of three basic components: servers, network infrastructure, and storage.
- These components can be further broken down into the following key elements: node ports, cabling, interconnecting devices (such as FC switches or hubs), storage arrays, and SAN management software.

1 Node Ports

- In fibre channel, devices such as hosts, storage and tape libraries are all referred to as nodes.
- Each node is a source or destination of information for one or more nodes.
- Each node requires one or more ports to provide a physical interface for communicating with other nodes.
- These ports are integral components of an HBA and the storage front-end adapters.
- A port operates in full duplex data transmission mode with a transmit (Tx) link and a receive (Rx) link (see Figure 6-3).

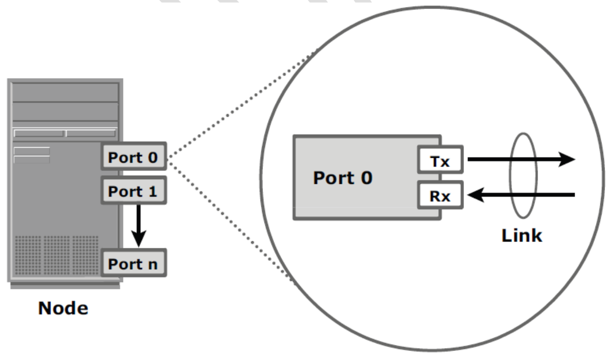


Figure 6-3: Nodes, ports, and links

2 Cabling

- SAN implementations use optical fiber cabling.
- Copper can be used for shorter distances for back-end connectivity, as it provides a better 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, multi-mode and single-mode.
- Multi-mode fiber (MMF) cable carries multiple beams of light projected at different angles simultaneously onto the core of the cable (see Figure 6-4 (a)).
- Based on the bandwidth, multi-mode fibers are classified as OM1 (62.5µm), OM2 (50µm) and laser optimized OM3 (50µm).

- 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 usually used for distances of up to 500 meters 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 (see Figure 6-4 (b)).
- These cables are available in diameters of 7–11 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 limits modal dispersion.
- Among all types of fibre cables, single-mode provides minimum signal attenuation over maximum distance (up to 10 km).
- A single-mode cable is used for long-distance cable runs, limited only by the power of the laser at the transmitter and sensitivity of the receiver.

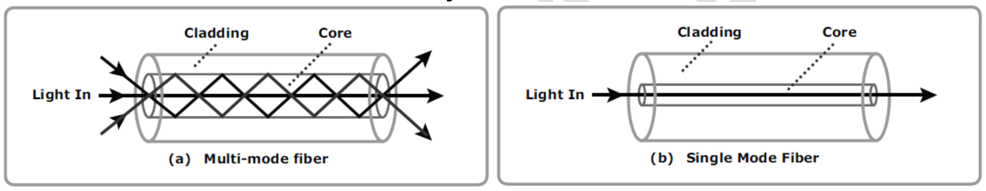


Figure 6-4: Multi-mode fiber and single-mode fiber

- MMFs are generally used within data centers for shorter distance runs, while SMFs are used for longer distances.
- MMF transceivers are less expensive as compared to SMF transceivers.
- A Standard connector (SC) (see Figure 6-5 (a)) and a Lucent connector (LC) (see Figure 6-5 (b)) are two commonly used connectors for fiber optic cables.
- An SC is used for data transmission speeds up to 1 Gb/s, whereas an LC is used for speeds up to 4 Gb/s.
- Figure 6-6 depicts a Lucent connector and a Standard connector.
- A Straight Tip (ST) is a fiber optic connector with a plug and a socket that is locked with a half-twisted bayonet lock (see Figure 6-5 (c)).
- In the early days of FC deployment, fiber optic cabling predominantly used ST connectors.
- This connector is often used with Fibre Channel patch panels.

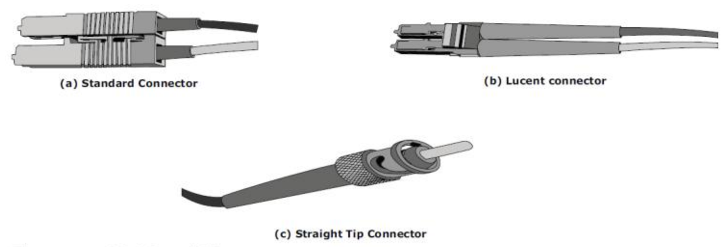


Figure 6-5: SC, LC, and ST connectors

- The Small Form-factor Pluggable (SFP) is an optical transceiver used in optical communication.
- The standard SFP+ transceivers support data rates up to 10 Gb/s.

3 Interconnect Devices

- Hubs, switches, and directors are the interconnect devices commonly used in 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 bandwidth because data travels through all the connection points.
- Because of availability of low cost and high performance switches, hubs are no longer used in 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, resulting in bandwidth aggregation.
- Directors are larger than switches and are deployed for data center implementations.
- The function of directors is similar to that of FC switches, but directors have higher port count and fault tolerance capabilities.

4 Storage Arrays

- The fundamental purpose of a SAN is to provide host access to storage resources.
- The capabilities of intelligent storage arrays are detailed in Chapter 4.
- The large storage capacities offered by modern storage arrays have been exploited in SAN environments for storage consolidation and centralization.
- SAN implementations complement the standard features of storage arrays by providing high availability and redundancy, improved performance, business continuity, and multiple host connectivity.

5 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 logical partitioning of the SAN, called zoning.
- In addition, the software provides management of typical SAN components such as HBAs, storage components, and interconnecting devices.

FC Connectivity

- The FC architecture supports three basic interconnectivity options: point-topoint, arbitrated loop (FC-AL), and fabric connect.

1 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, as only two devices can communicate with each other at a given time.
- Moreover, it cannot be scaled to accommodate a large number of network devices.
- Standard DAS uses point to point connectivity.

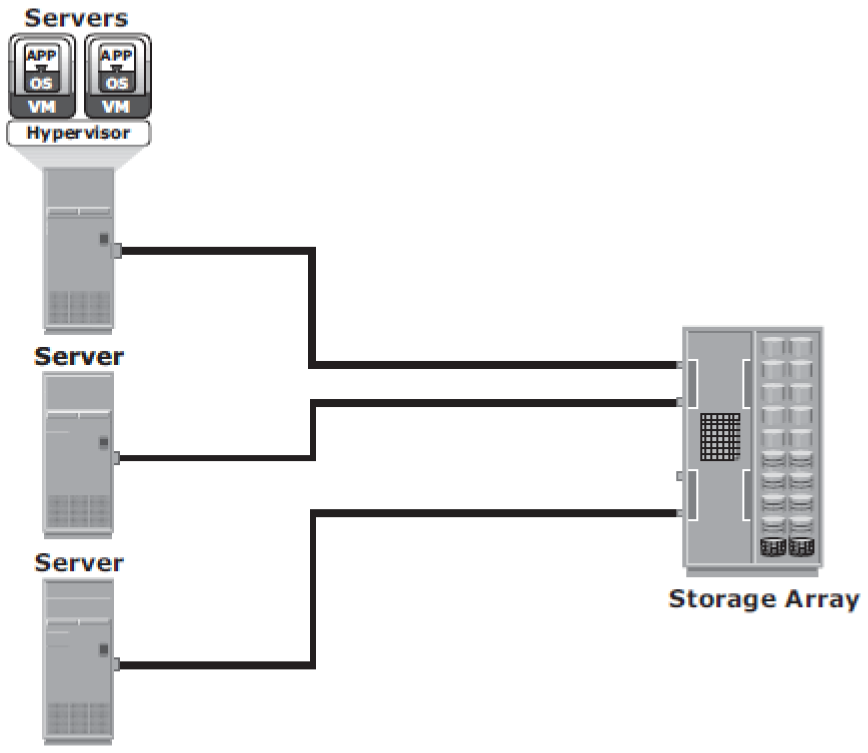


Figure 5-6: Point-to-point connectivity

2 Fibre Channel Arbitrated Loop

- In the FC-AL configuration, devices are attached to a shared loop, as shown in Figure 6-7.
- 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.
- As a loop configuration, FC-AL can be implemented without any interconnecting devices by directly connecting one device to another in a ring through cables.
- However, FC-AL implementations may also use hubs whereby the arbitrated loop is physically connected in a star topology.
- The FC-AL configuration has the following limitations in terms of scalability:
  - ➢ FC-AL shares the bandwidth in the loop.
  - ➢ Only one device can perform I/O operations at a time.
  - ➢ Because each device in a loop has to wait for its turn to process an I/O request, the speed of data transmission is low in an FC-AL topology.
  - ➢ . FC-AL uses 8-bit addressing. It can support up to 127 devices on a loop.
  - ➢ Adding or removing a device results in loop re-initialization, which can cause a momentary pause in loop traffic.

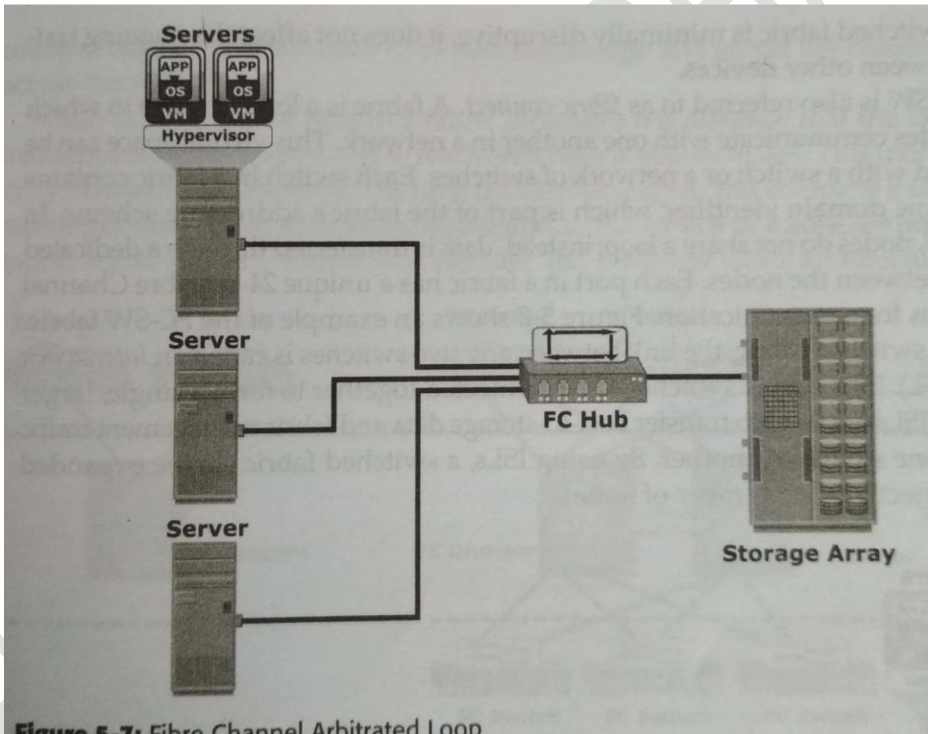


Figure 5-7: Fibre Channel Arbitrated Loop

3 Fibre Channel Switched Fabric

- Unlike a loop configuration, a Fibre Channel switched fabric (FC-SW) network provides interconnected devices, dedicated bandwidth, and scalability.

- The addition or removal of a device in a switched fabric is minimally disruptive; it does not affect the ongoing traffic between other devices.
- 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

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.

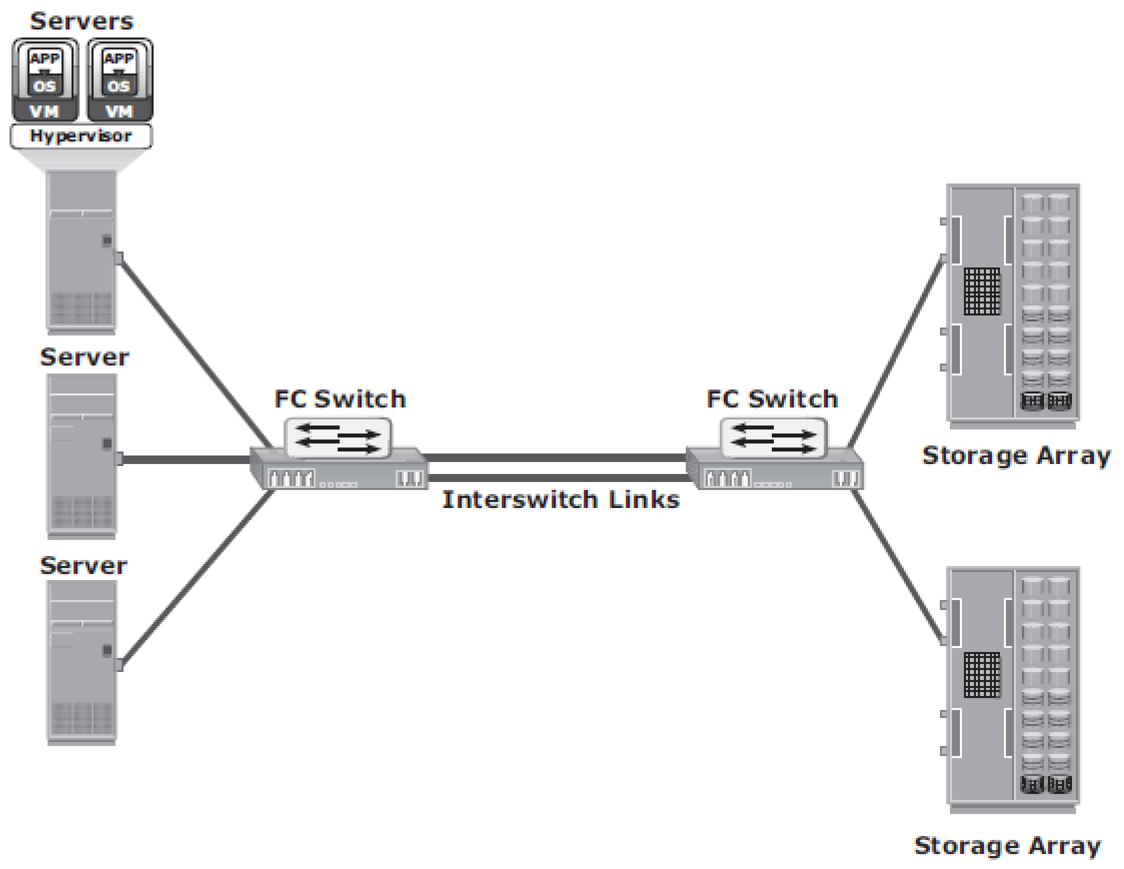


Figure 5-8: Fibre Channel switched fabric

A fabric can be described by the number of tiers it contains. The number of tiers in a fabric is based on the number of switches traversed between two points that are farthest from each other. This number is based on the infrastructure constructed by the fabric instead of how the storage and server are connected across the switches.

- When the number of tiers in a fabric increases, the distance that a fabric management message must travel to reach each switch in the fabric also increases.
- The 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 (detailed later in this chapter).
- Figure 5-9 illustrates two-tier and three-tier fabric architecture.

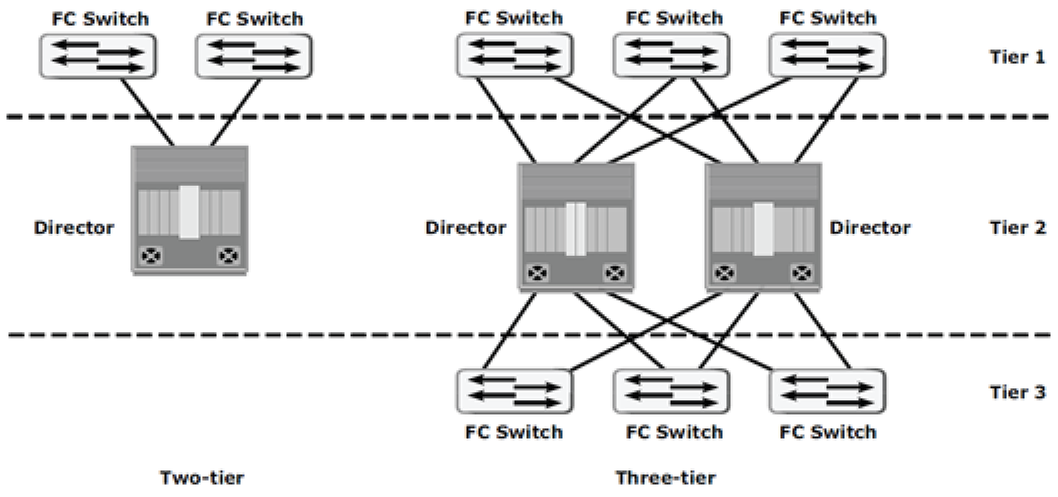


Figure 6-10: Tiered structure of FC-SW topology

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.

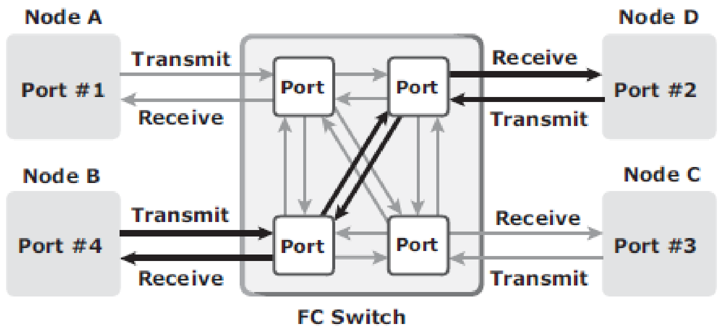


Figure 5-10: Data transmission in Fibre Channel switched fabric

Fibre Channel Architecture:

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. FCP forms the fundamental construct of the FC SAN infrastructure.

Fibre Channel provides a serial data transfer interface that operates over copper wire and optical fiber. FCP is the implementation of serial SCSI over an FC network. In FCP architecture, all external and remote storage devices attached to the SAN appear as local devices to the host operating system. The key advantages of FCP are as follows:

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

Fibre Channel Protocol Stack:

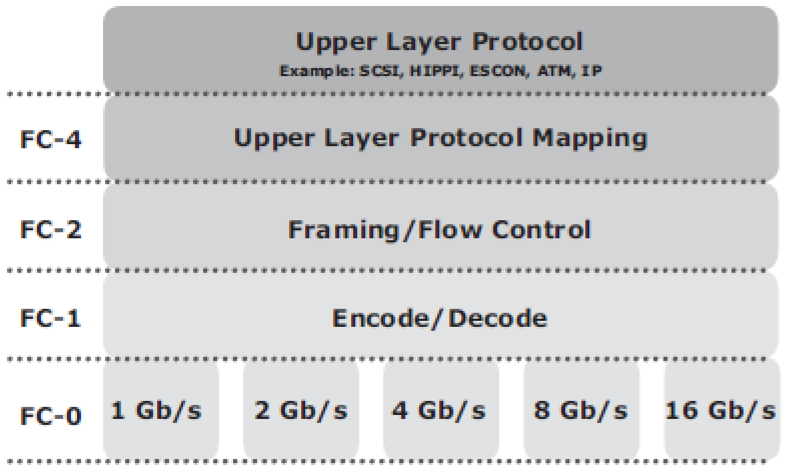


Figure 5-12: Fibre Channel protocol stack

- 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.

FC-4 Upper Layer Protocol

- 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 6-7).
- Some of the protocols include SCSI, HIPPI Framing Protocol, Enterprise Storage Connectivity (ESCON), ATM, and IP.

FC-2 Transport Layer

- The FC-2 is the transport layer that contains the payload, addresses of the source and destination ports, and link control information.
- 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 Transmission Protocol

- This layer defines the transmission protocol that includes serial encoding and decoding rules, special characters used, and error control.
- 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-0 Physical Interface

- FC-0 is the lowest layer in the FCP stack.
- This layer defines the physical interface, media, and transmission of raw 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.

2 Fibre Channel Addressing

- An FC address is dynamically assigned when a port logs on to the fabric.
- The FC address has a distinct format that varies according to the type of node port in the fabric.
- These ports can be an N_port and an NL_port in a public loop, or an NL_port in a private loop.
- The first field of the FC address of an N_port contains the domain ID of the switch (see Figure 6-14).
- This is an 8-bit field.
- Out of the possible 256 domain IDs, 239 are available for use; the remaining 17 addresses are reserved for specific services.

- For example, FFFFFC is reserved for the name server, and FFFFFE is reserved for the fabric login service.
- The maximum possible number of N_ports in a switched fabric is calculated as 239 domains × 256 areas × 256 ports = 15,663,104 Fibre Channel addresses.

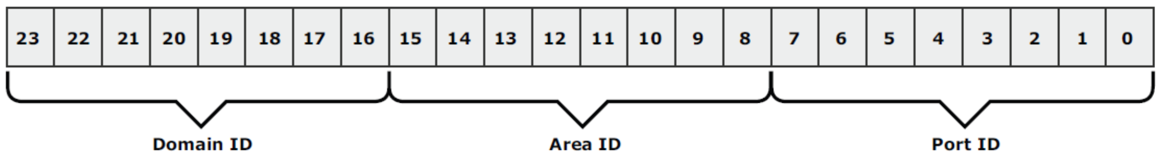


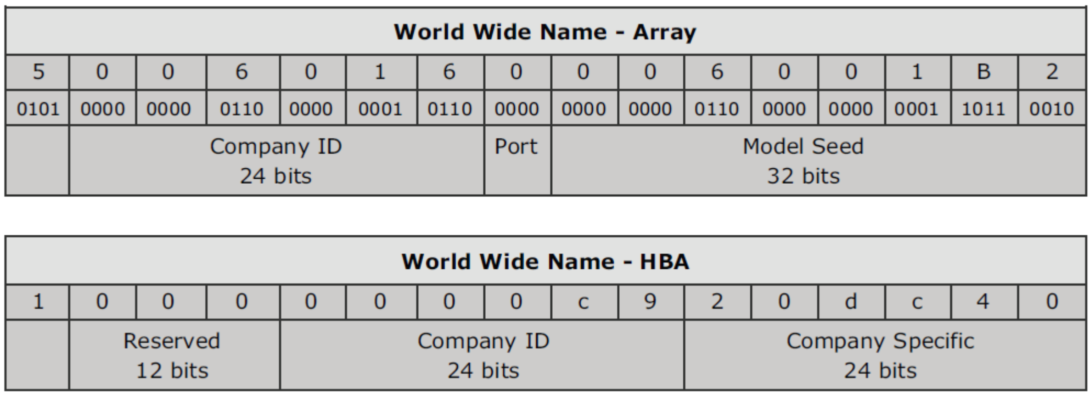
Figure 6-14: 24-bit FC address of N_port

- The area ID is used to identify a group of F_ports.
- An example of a group of F_ports would be a card on the switch with more than one port on it.

The last field in the FC address identifies the F_port within the group

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).
- 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.
- WWNs are burned into the hardware or assigned through software.
- 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 6-16 illustrates the WWN structure for an array and the 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
	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
	Reserved 12 bits			Company ID 24 bits						Company Specific 24 bits					

Figure 6-16: World Wide Names

FC Frame

- An FC frame (Figure 6-17) consists of five parts: start of frame (SOF), frame header, data field, cyclic redundancy check (CRC), and end of frame (EOF).
- The SOF and EOF act as delimiters.
- In addition to this role, the SOF is a flag that indicates whether the frame is the first frame in a sequence of frames.
- 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 S_ID and D_ID are standard FC addresses for the source port and the destination port, respectively.
- The SEQ_ID and OX_ID identify the frame as a component of a specific sequence and exchange, respectively.
- The frame header also defines the following fields:

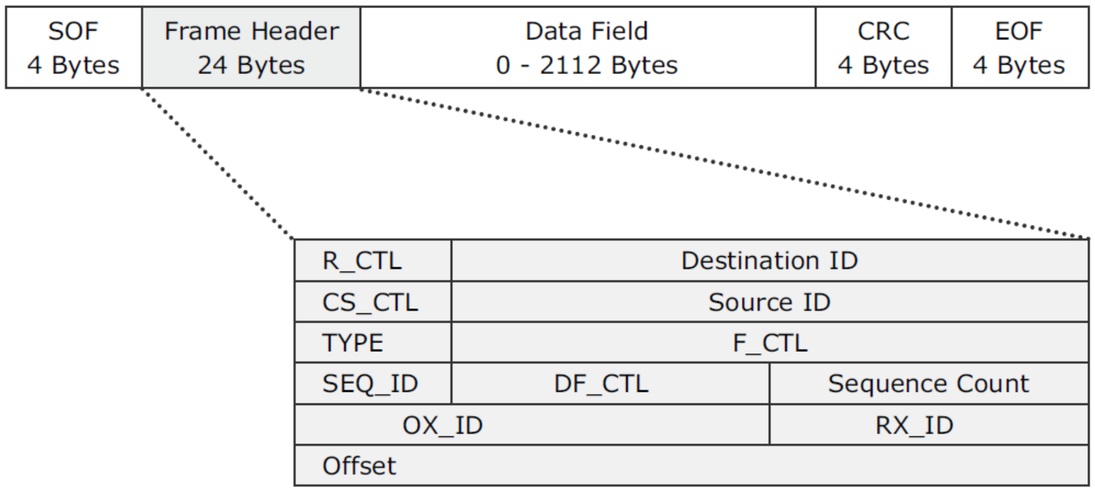


Figure 6-17: FC frame

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.
- In contrast, data frames carry the payload and are used for data transmission.

Class Specific Control (CS_CTL):

- This field specifies link speeds for class 1 and class 4 data transmission.

TYPE:

- This field describes the upper layer protocol (ULP) to be carried on the frame if it is a data frame.
- However, if it is a link control frame, this field is used to signal an event such as "fabric busy."
- For example, if the TYPE is 08, and the frame is a data frame, it means that the SCSI will be carried on an FC.

Data Field Control (DF_CTL):

- A 1-byte field that indicates the existence of any optional headers at the beginning of the data payload.
- It is a mechanism to extend header information into the payload.

Frame Control (F_CTL):

- A 3-byte field that contains control information related to frame content.
- For example, one of the bits in this field indicates whether this is the first sequence of the exchange.
- The data field in an FC frame contains the data payload, up to 2,112 bytes of original data — in most cases, SCSI data.
- The biggest possible payload an FC frame can deliver is 2,112 bytes of data with 36 bytes of fixed overhead.

- A link control frame, by definition, has a payload of 0 bytes.
- Only data frames carry a payload.
  - The CRC checksum facilitates error detection for the content of the frame.
  - This checksum verifies data integrity by checking whether the content of the frames was received correctly.
  - The CRC checksum is calculated by the sender before encoding at the FC-1 layer.
  - Similarly, it is calculated by the receiver after decoding at the FC-1 layer.

Structure and Organization of FC Data

- In an FC network, data transport is analogous to a conversation between two people, whereby a frame represents a word, a sequence represents a sentence, and an exchange represents a conversation.

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 depending upon the type of data sequence exchanged between the initiator and the target.

Sequence:

- A sequence refers to a contiguous set of frames that are sent from one port to another.
- A sequence corresponds to an information unit, as defined by the ULP.

Frame:

- A frame is the fundamental unit of data transfer at Layer 2.
- Each frame can contain up to 2,112 bytes of payload.

6.6.5 Flow Control

- Flow control defines the pace of the flow of data frames during data transmission.
- FC technology uses two flow-control mechanisms: buffer-to-buffer credit (BB_Credit) and end-to-end credit (EE_Credit).

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 transmitting port maintains a count of free receiver buffers and continues to send frames if the count is greater than 0.
- The BB_Credit mechanism provides frame acknowledgment through the Receiver Ready (R_RDY) primitive.

EE_Credit

- The function of end-to-end credit, known as EE_Credit, is similar to that of BB_Credit.
- When an initiator and a target establish themselves as nodes communicating with each other, they exchange the EE_Credit parameters (part of Port Login).
- The EE_Credit mechanism affects the flow control for class 1 and class 2 traffic only.

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 (see Figure 5-17).**

Whenever a change takes place in the name server database, the fabric controller sends a Registered State Change Notification (RSCN) to all the nodes impacted by the change.

If zoning is not configured, the fabric controller sends an RSCN to all the nodes in the fabric. Involving the nodes that are not impacted by the change results in increased fabric-management traffic.

sFor a large fabric, the amount of FC traffic generated due to this process can be significant and might impact the host-to-storage data traffic. Zoning helps to limit the number of RSCNs in a fabric. In the presence of zoning, a fabric sends the RSCN to only those nodes in a zone where the change has occurred.

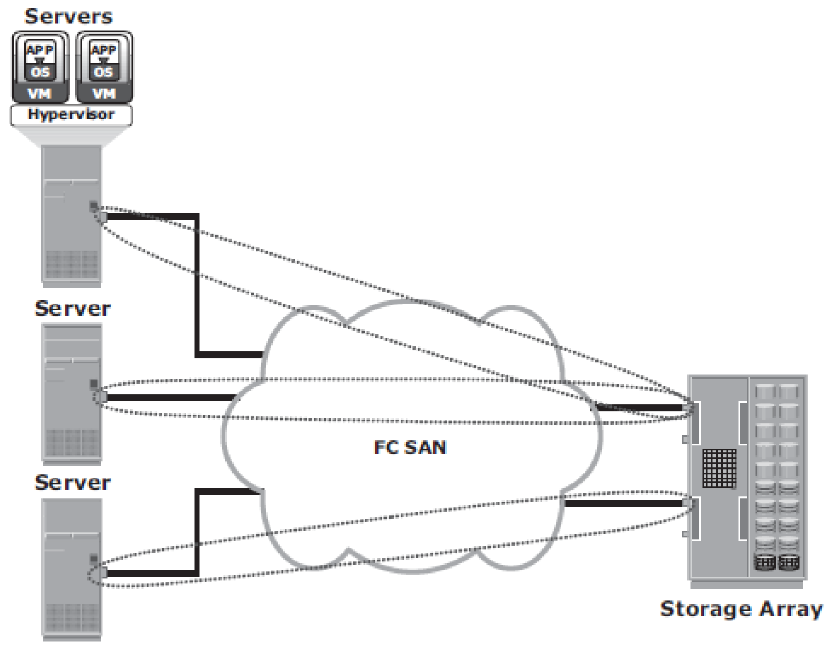


Figure 5-17: Zoning

- 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.

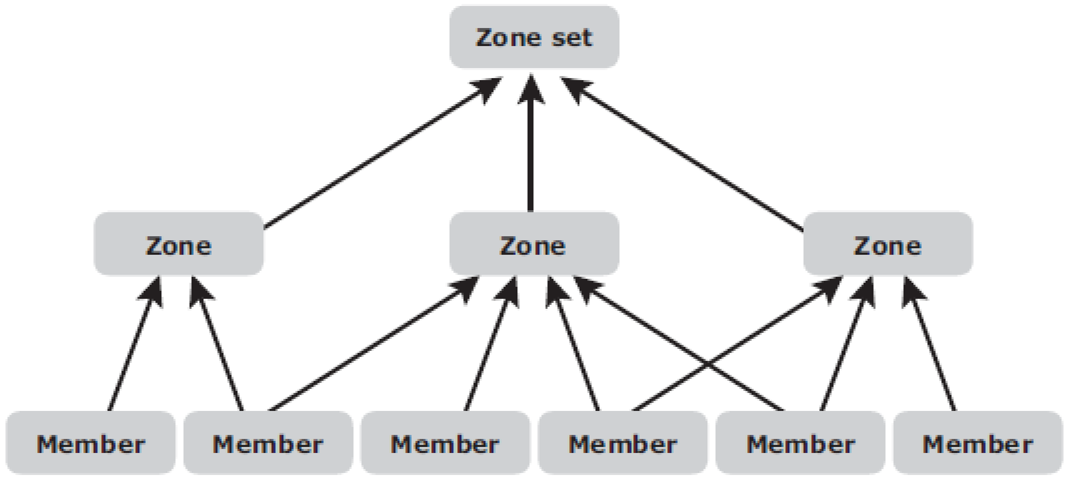


Figure 5-18: Members, zones, and zone sets

- 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:** Uses the physical address of switch ports to define zones. In port zoning, access to node is determined by the physical switch port to which a node is connected. The zone members are the port identifier (switch domain ID and port number) to which HBA and its targets (storage devices) are connected. If a node is moved to another switch port in the fabric, then zoning must be modified to allow the node, in its new port, to participate in its original zone. However, if an HBA or storage device port fails, an administrator just has to replace the failed device without changing the zoning configuration.

2) **WWN zoning:** Uses World Wide Names to define zones. The zone members are the unique WWN addresses of the HBA and its targets (storage devices). A major advantage of WWN zoning is its flexibility. WWN zoning allows nodes to be moved to another switch port in the fabric and maintain connectivity to its zone partners without having to modify the zone configuration. This is possible because the WWN is static to the node port.
3) **Mixed zoning:** Combines the qualities of both WWN zoning and port zoning. Using mixed zoning enables a specific node port to be tied to the WWN of another node.

Figure 5-19 shows the three types of zoning on an FC network.6.8 Fibre Channel Login Types

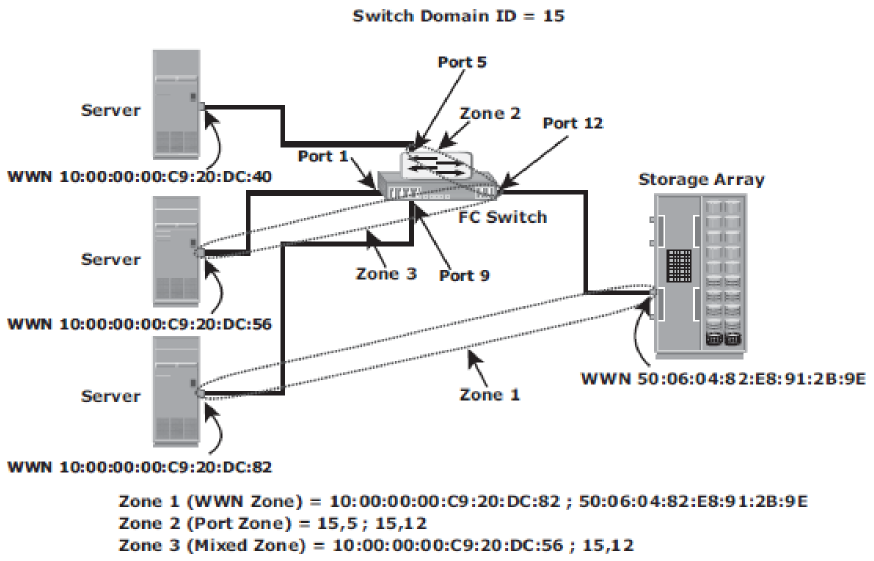

Switch Domain ID = 15
Server
Port 5
Zone 2
Port 12
Port 1
WWN 10:00:00:00:C9:20:DC:40
Storage Array
FC Switch
Server
Zone 3
Port 9
WWN 10:00:00:00:C9:20:DC:56
Server
Zone 1
WWN 50:06:04:82:E8:91:2B:9E
WWN 10:00:00:00:C9:20:DC:82
Zone 1 (WWN Zone) = 10:00:00:00:C9:20:DC:82 ; 50:06:04:82:E8:91:2B:9E
Zone 2 (Port Zone) = 15,5 ; 15,12
Zone 3 (Mixed Zone) = 10:00:00:00:C9:20:DC:56 ; 15,12

Figure 5-19: Types of zoning

FC SAN Topologies:

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 c to reach its destination. Partial mesh offers more scalability than full mesh topology. However, without proper placement of host and storage devices, traffic management in a partial mesh fabric might be complicated and ISLs could become overloaded due to excessive traffic aggregation. Figure 5-20 depicts both partial mesh and full mesh topologies.

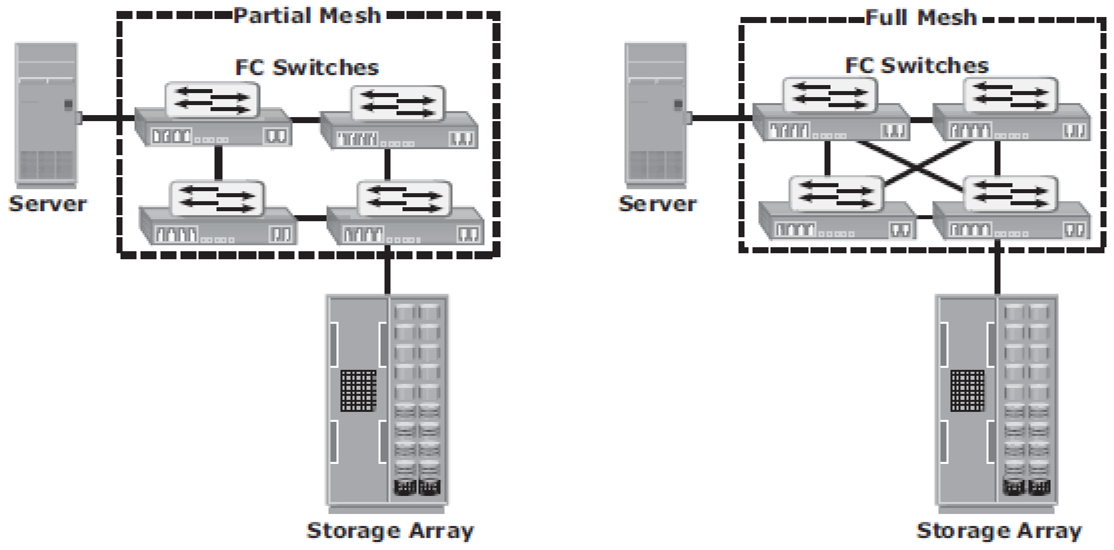


Figure 5-20: Partial mesh and full mesh topologies

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.

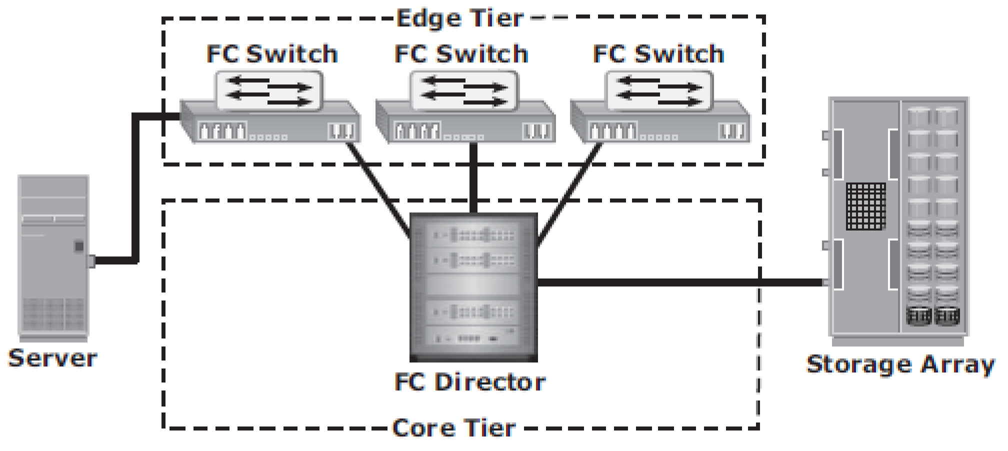


Figure 5-21: Single-core topology

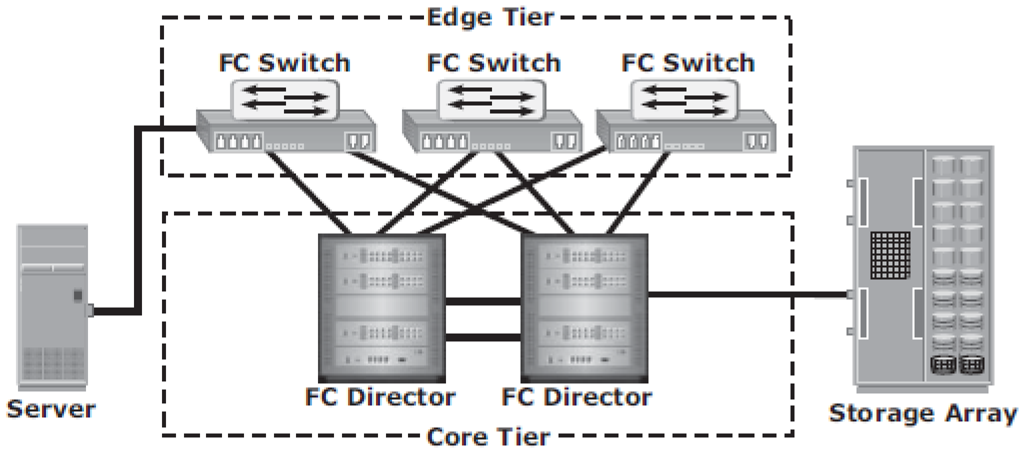


Figure 5-22: Dual-core topology

Benefits and Limitations of Core-Edge Fabric

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.

In this topology, because each tier's switch port is used for either storage or hosts, it's easy to identify which network resources are approaching their capacity, making it easier to develop a set of rules for scaling and apportioning.

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.

However, the core-edge fabric might lead to some performance-related problems because scaling a core-edge topology involves increasing the number of hop counts in the fabric. 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. Generally, a large hop count means a high data transmission delay between the source and destination.

Virtualization in SAN:

Block-level Storage Virtualization

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.

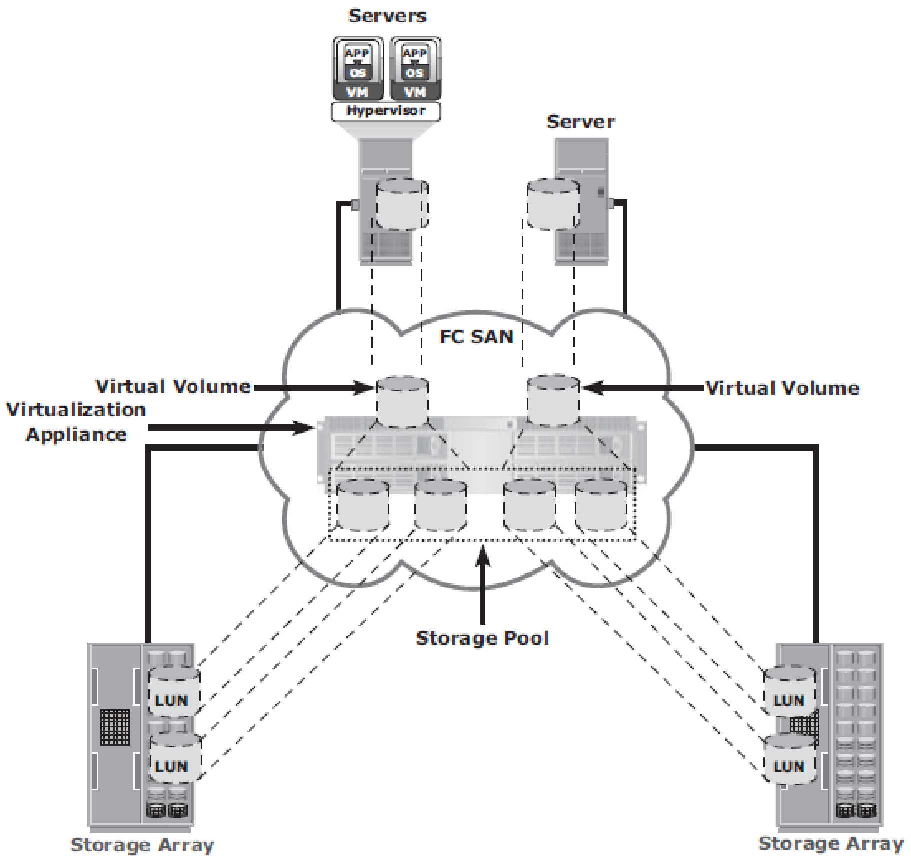


Figure 5-24: Block-level storage virtualization

Figure 5-24 illustrates a virtualized environment. It shows two physical servers, each of which has one virtual volume assigned. These virtual volumes are used by the servers. These virtual volumes are mapped to the LUNs in the storage arrays. When an I/O is sent to a virtual volume, it is redirected through the virtualization layer at the storage network to the mapped LUNs.

Depending on the capabilities of the virtualization appliance, the architecture may allow for more complex mapping between array LUNs and virtual volumes.

Block-level storage virtualization enables extending the storage volumes online to meet application growth requirements. It consolidates heterogeneous storage arrays and enables transparent volume access.

Block-level storage virtualization also provides the advantage of nondisruptive data migration. In a traditional SAN environment, LUN migration from one array to another is an offline event because the hosts needed to be updated to reflect the new array confi guration.

In other instances, host CPU cycles were required to migrate data from one array to the other, especially in a multivendor environment.

With a block-level virtualization solution in place, the virtualization layer handlesthe back-end migration of data, which enables LUNs to remain online and accessible while data is migrating. No physical changes are required because the host still points to the same virtual targets on the virtualization layer. However, the mappings information on the virtualization layer should be changed. These changes can be executed dynamically and are transparent to the end user.

Previously, block-level storage virtualization provided nondisruptive data migration only within a data center. The new generation of block-level storage virtualization enables nondisruptive 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 (seeFigure 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.

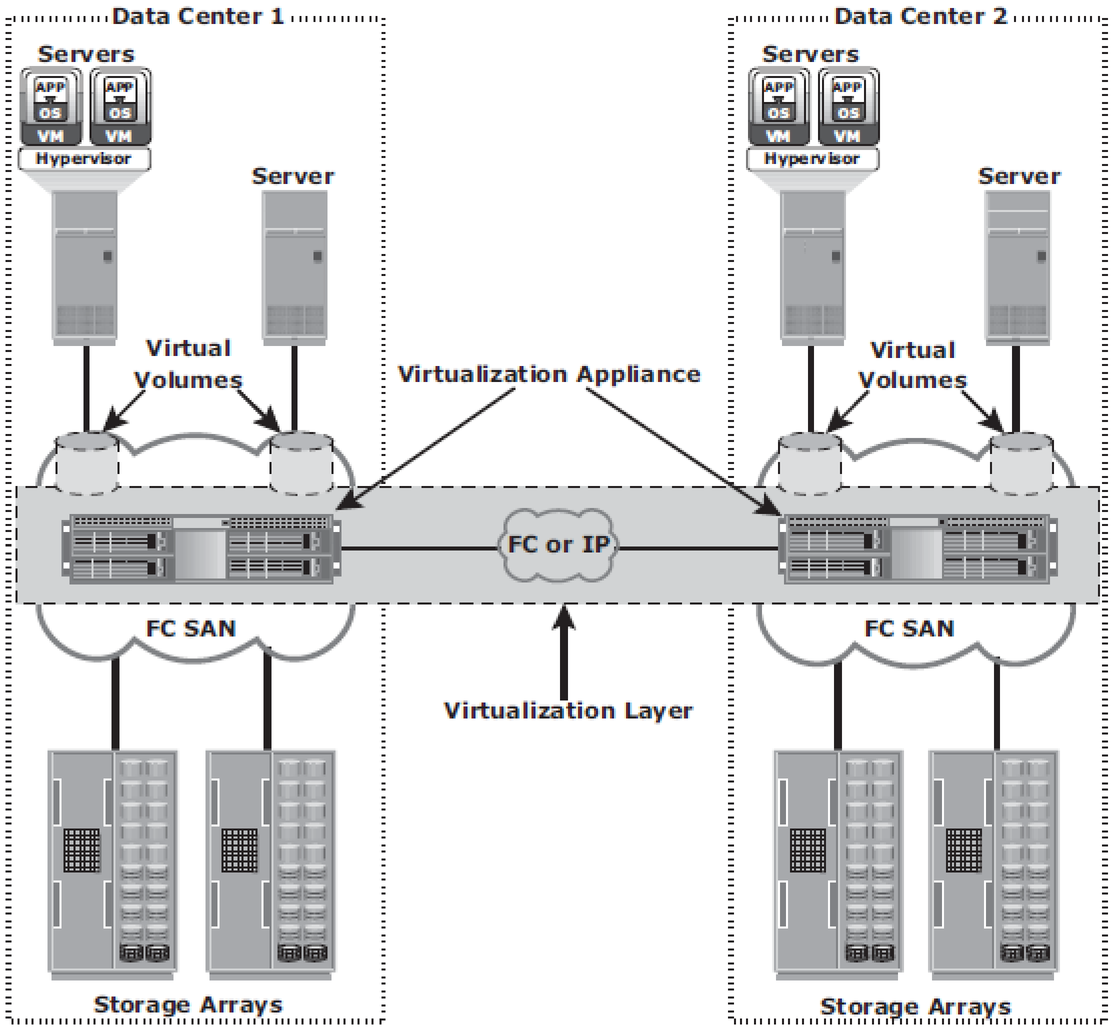


Figure 5-25: Federation of block storage across data centers

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. VSANs improve SAN security, scalability, availability, and manageability. VSANs provide enhanced security by isolating the sensitive data in a VSAN and by restricting access to the resources located within that VSAN.

The same Fibre Channel address can be assigned to nodes in different VSANs, thus increasing the fabric scalability. Events causing traffic disruptions in one VSAN are contained within that VSAN and are not propagated to other VSANs.

VSANs facilitate an easy, flexible, and less expensive way to manage networks. Configuring VSANs is easier and quicker compared to building separate physical FC SANs for various node groups.

IP SAN and FCoE

Two primary protocols that leverage IP as the transport mechanism are Internet **SCSI (iSCSI)** and **Fibre Channel over IP (FCIP).**

iSCSI is encapsulation of SCSI I/O over IP. FCIP is a protocol in which an FCIP entity such as an FCIP gateway is used to tunnel FC fabrics through an IP network.

In FCIP, FC frames are encapsulated onto the IP payload. An FCIP implementation is capable of merging interconnected fabrics into a single fabric

iSCSI

iSCSI is 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.

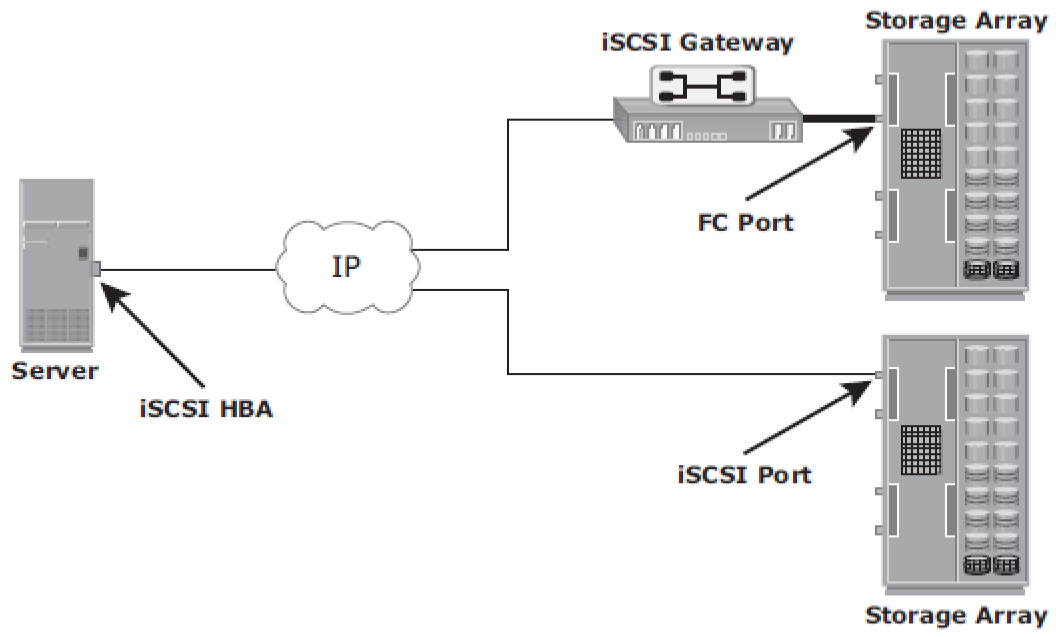


Figure 6-1: iSCSI implementation

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

- **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.
- 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.

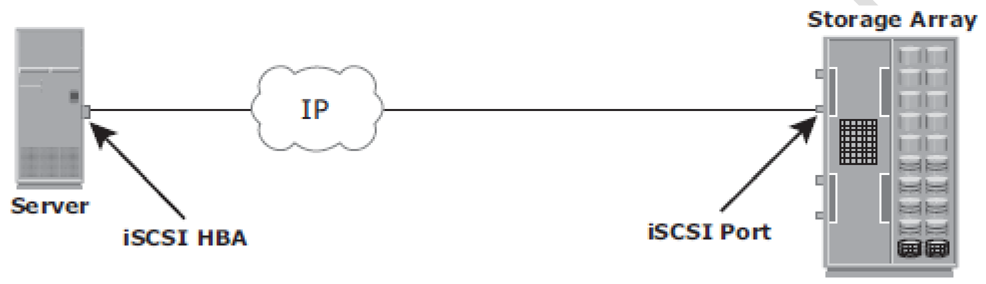
iSCSI Topologies:

Two topologies of iSCSI implementations are native and bridged.

Native iSCSI Connectivity

Native topology does not have FC components. The initiators may be either directly attached to targets or connected through the IP network.

FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed. In the following Figure, the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernet switch.



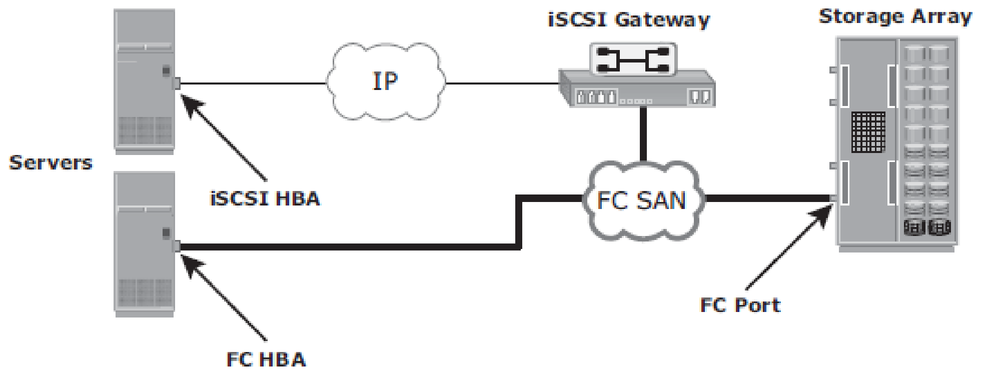
(a) Native iSCSI Connectivity

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

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.

A bridged iSCSI implementation includes FC components in its configuration. The following figure illustrates iSCSI host connectivity to an FC storage array.



(b) Bridged iSCSI Connectivity

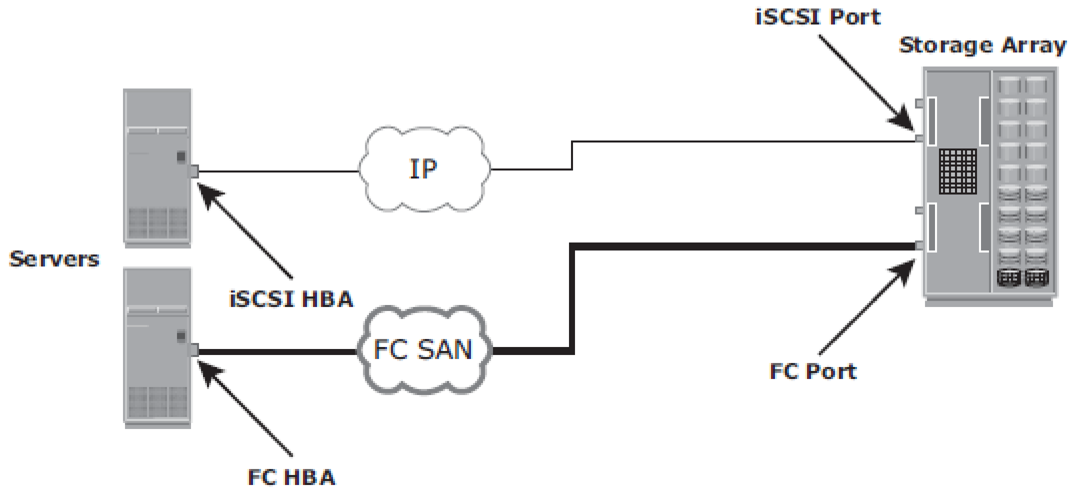
In this case, the array does not have any iSCSI ports. Therefore, an external device, called a gateway or a multiprotocol router, must be used to facilitate the communication between the iSCSI host and FC storage.

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 the following Figure.



(c) Combining FC and Native iSCSI Connectivity

Figure 6-2: iSCSI Topologies

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.

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.

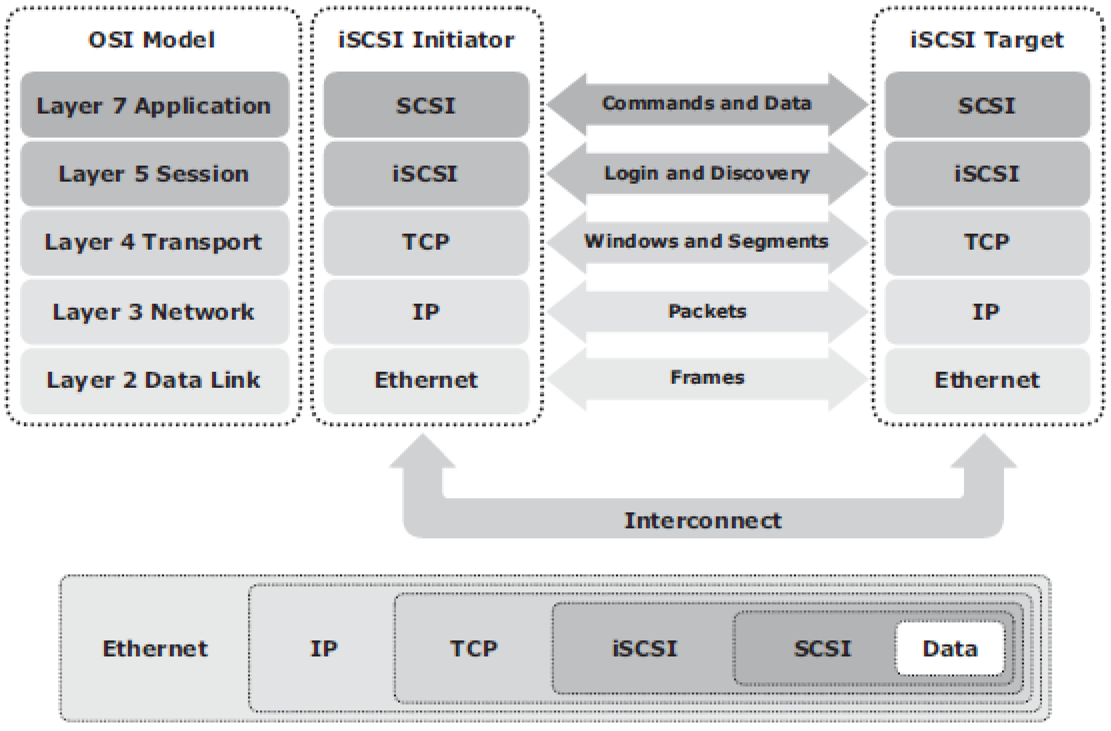


Figure 6-3: iSCSI protocol stack

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

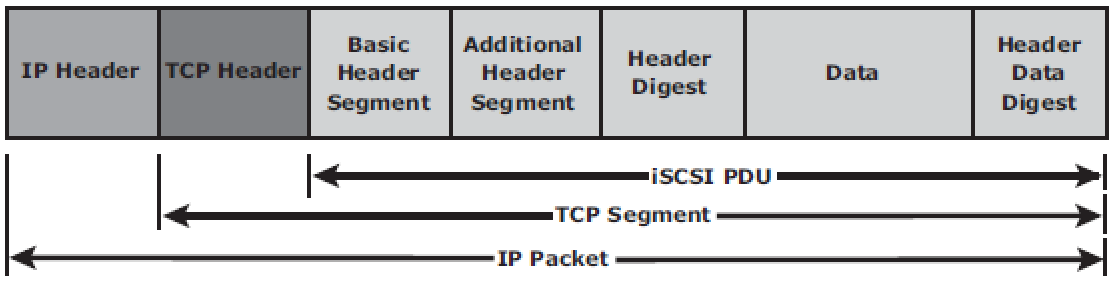


Figure 6-4: iSCSI PDU encapsulated in an IP packet

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.

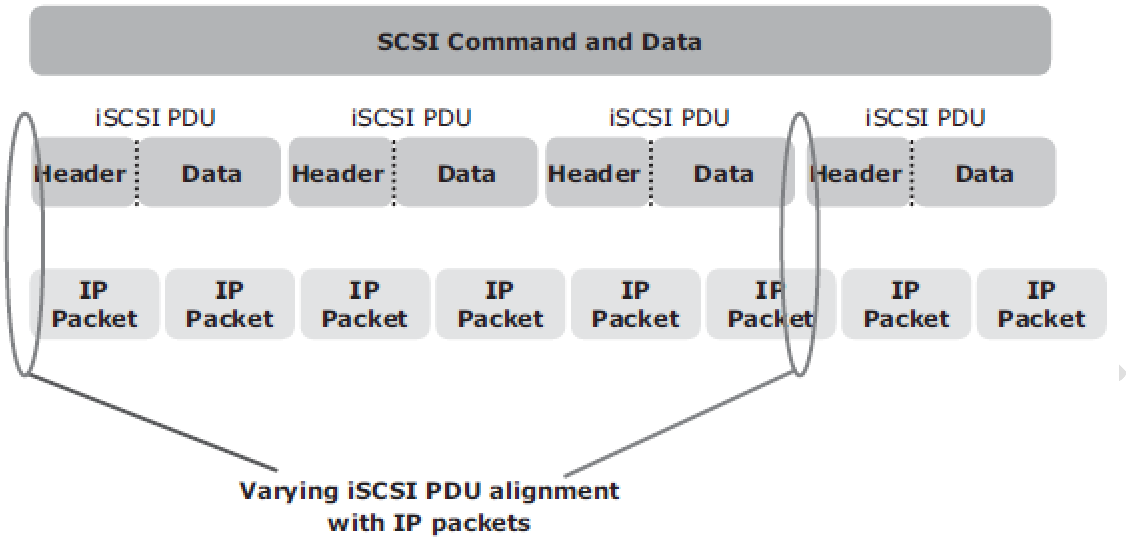


Figure 6-5: Alignment of iSCSI PDUs with IP packets

To achieve the 1:1 relationship between the IP packet and the iSCSI PDU, the maximum transmission unit (MTU) size of the IP packet is modified. This eliminates fragmentation of the IP packet, which improves the transmission efficiency.

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 or 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.

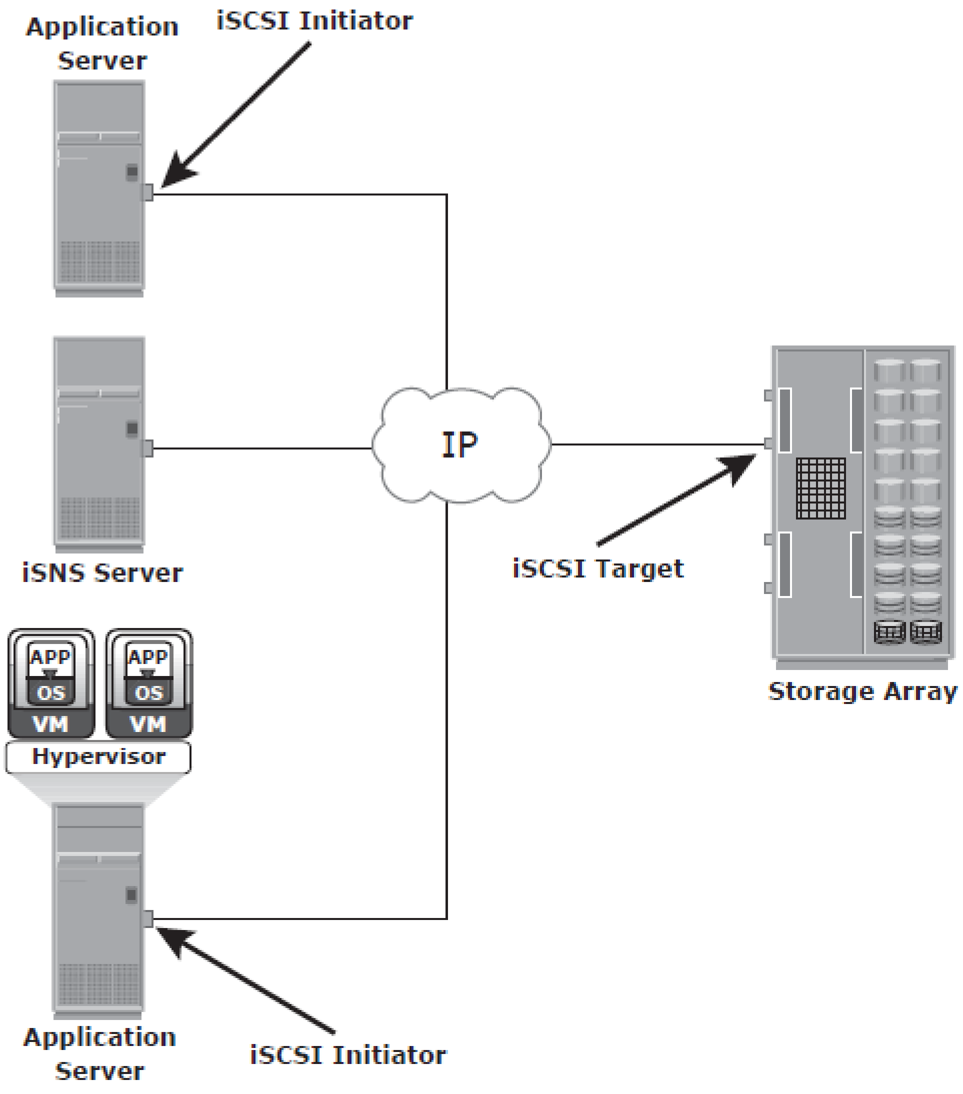


Figure 6-6: Discovery using iSNS

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.

The discovery can also take place by using service location protocol (SLP). However, this is less commonly used than SendTargets discovery and 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 iSCSInames commonly used:

1) **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 conf icts 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.

2) **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.

iSCSI Session:

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

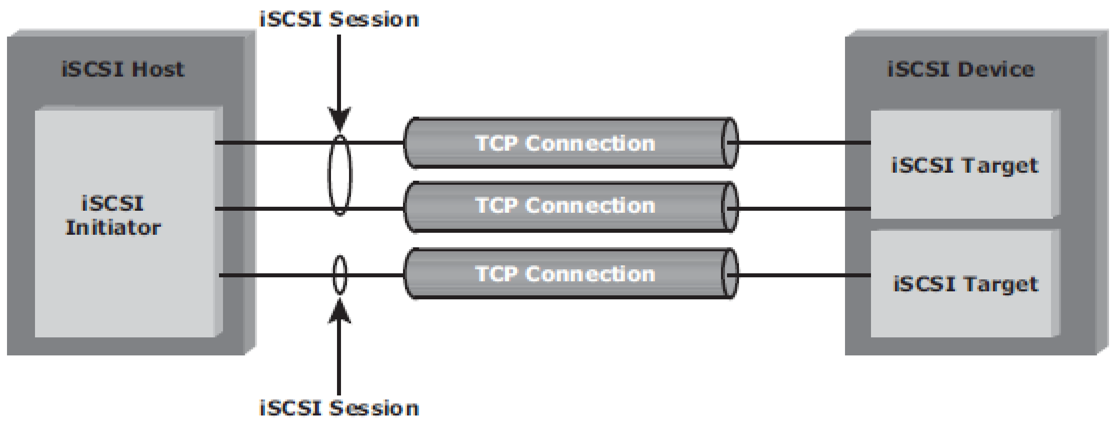


Figure 6-7: iSCSI session

A session is identified by a session ID (SSID), which includes part of aninitiator ID and a target ID. The session can be intended for one of the following:

1) The discovery of the available targets by the initiators and the location of a specific target on a network
2) 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 by encapsulating them in iSCSI PDUs that travel over the established TCP connection.

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.

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.

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. This ensures the correct order of data and commands at a target even when there are multiple TCP connections between an initiator and the target that use portal groups.

Similar to command numbering, a status sequence number (StatSN) is used to sequentially number status responses, as shown in Figure 6-8. 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. Each of these sequence numbers is stored locally as an unsigned 32-bit integer counter defined by iSCSI. These numbers are communicated between the initiator and target in the appropriate iSCSI PDU fields during command, status, and data exchanges.

For read operations, the DataSN begins at zero and is incremented by one for each subsequent data PDU in that command sequence. For a write operation, the first unsolicited data PDU or the first data PDU in response to an R2T begins with a DataSN of zero and increments by one for each subsequent data PDU.

R2TSN is set to zero at the initiation of the command and incremented by one for each subsequent R2T sent by the target for that command.

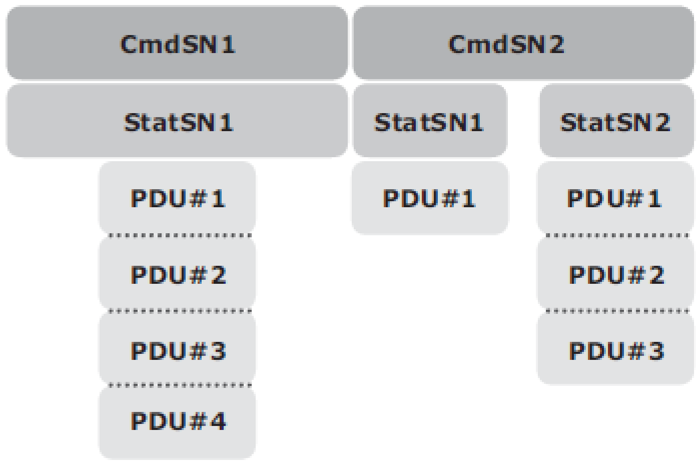


Figure 6-8: Command and status sequence number

FCIP

FCIP is a tunneling protocol that enables distributed FC SAN islands to be interconnected over the existing IP-based networks.

FCIP Protocol Stack

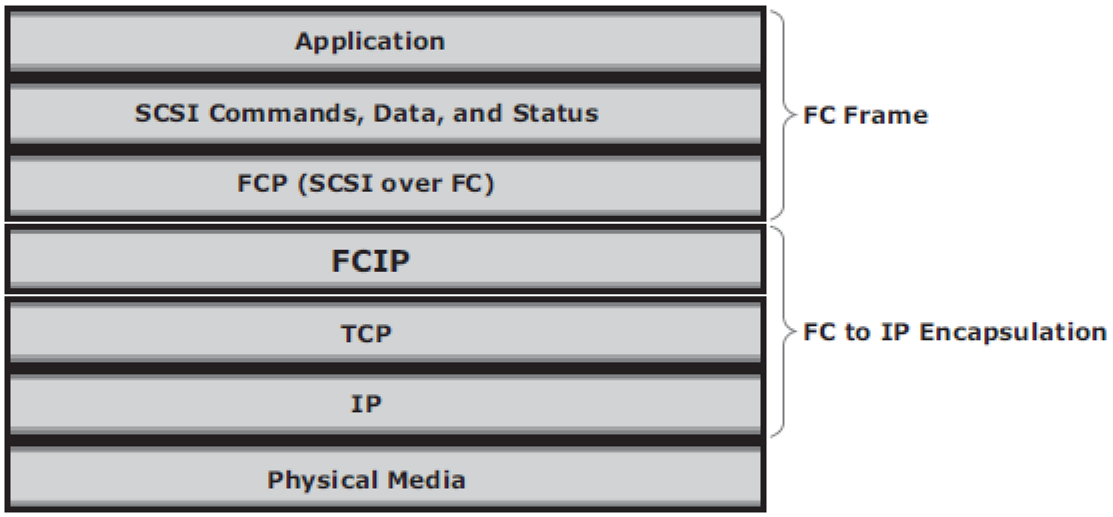


Figure 6-9: FCIP protocol stack

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.

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.

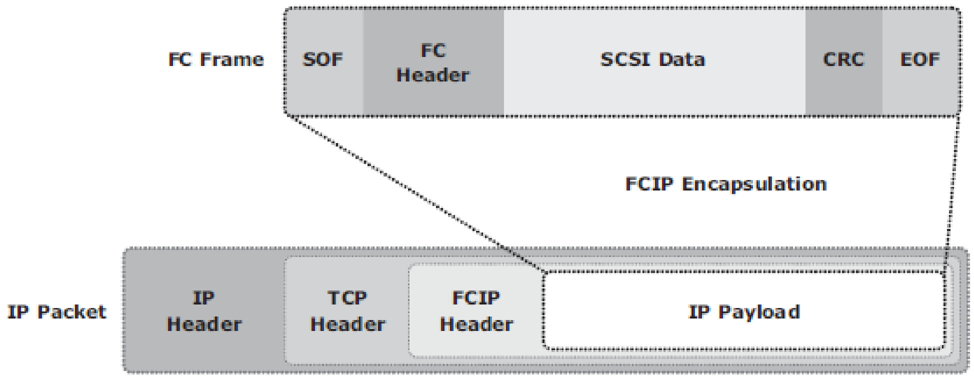


Figure 6-10: FCIP encapsulation

The FC frames can be encapsulated into the IP packet and sent to a remote SAN over the IP. 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.

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.

FCIP Topology:

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

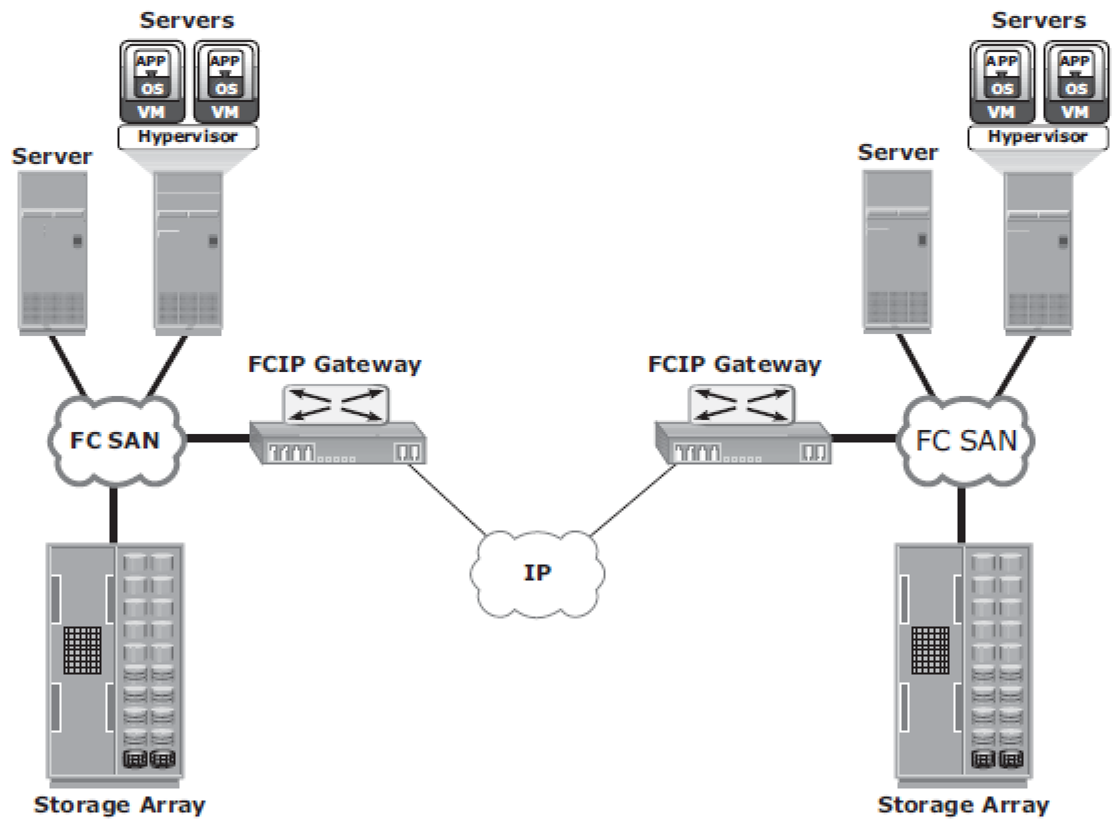


Figure 6-11: FCIP topology

- 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.

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.

From the perspective of performance, configuring multiple paths between FCIP gateways eliminates single points of failure and provides increased bandwidth. In a scenario of extended distance, the IP network might be a bottleneck if sufficient bandwidth is not available. In addition, because FCIP creates a unifiedfabric, disruption in the underlying IP network can cause instabilities in the SAN environment. These instabilities include a segmented fabric, excessive RSCNs, and host timeouts.

The vendors of FC switches have recognized some of the drawbacks related to FCIP and have implemented features to enhance stability, such as the capability to segregate the FCIP traffic into a separate virtual fabric.

Security is also a consideration in an FCIP solution because the data is transmitted over public IP channels. Various security options are available to protect the data based on the router’s support. IPSec is one such security measure that can be implemented in the FCIP environment.

FCoE

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.

I/O Consolidation Using FCoE

The key benefit of FCoE is I/O consolidation.

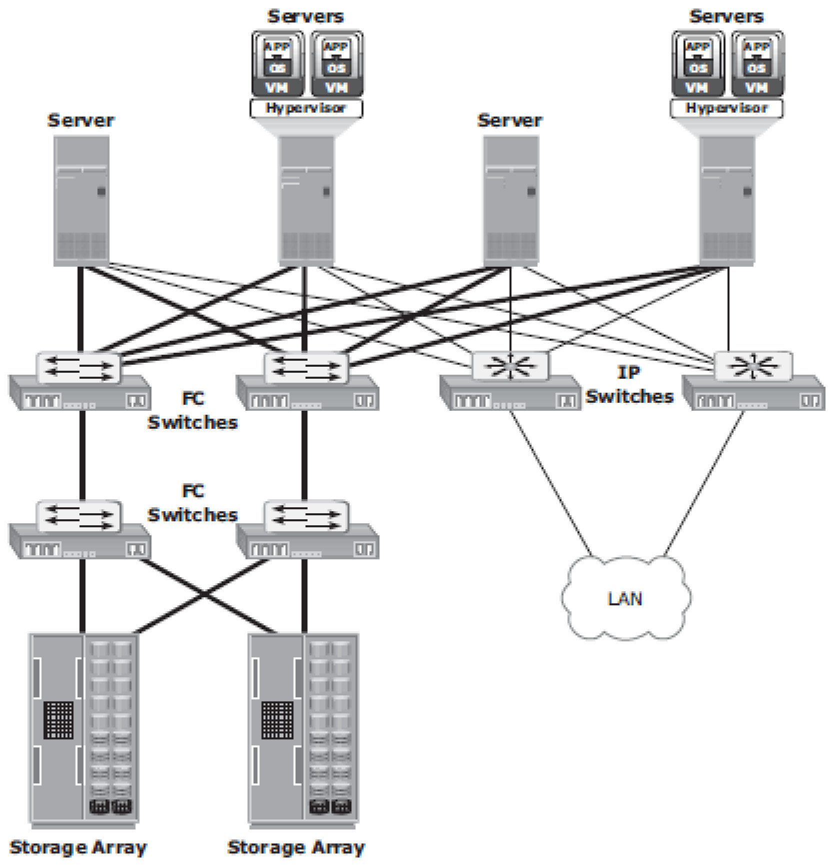


Figure 6-12: Infrastructure before using FCoE

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. The cost of power, cooling, and floor space further adds to the challenge.

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.

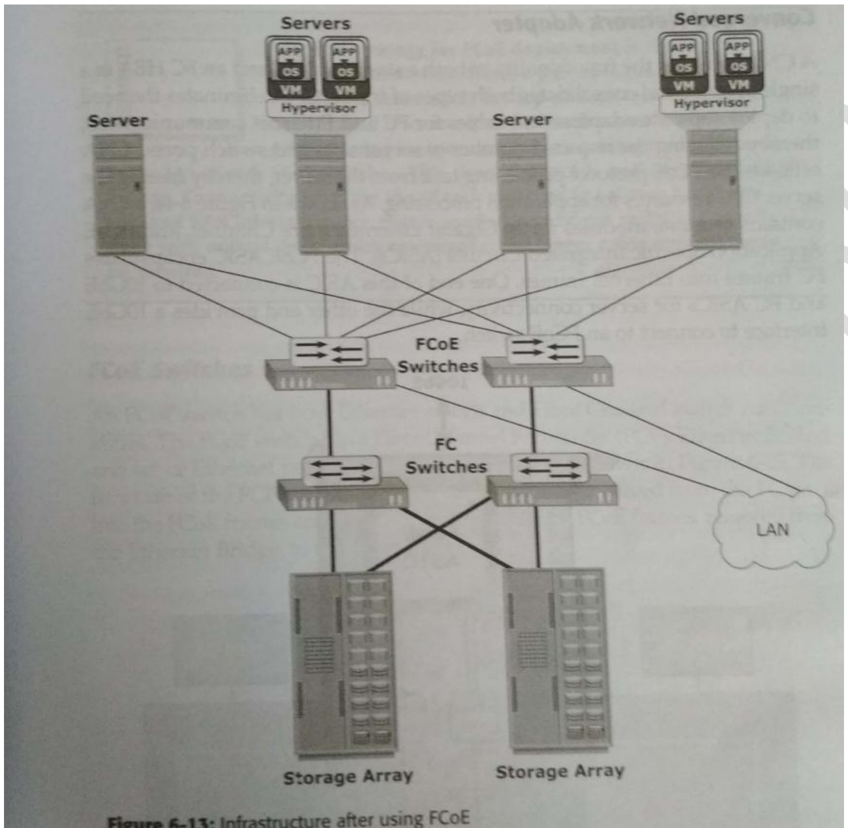


Figure 6-13: Infrastructure after using FCoE

Components of an FCoE Network:

1) *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.
- CNA offloads the FCoE protocol processing task from the server, thereby freeing the server CPU resources for application processing.

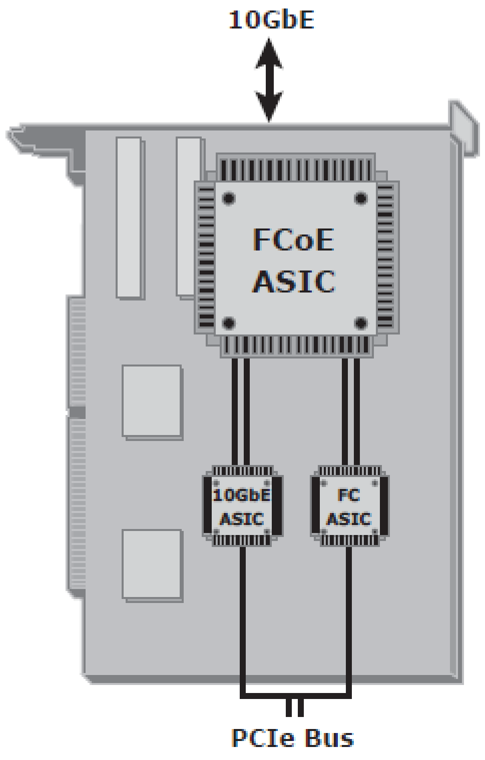


Figure 6-14: Converged Network Adapter

- As shown in Figure 6-14, a CAN 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.

2) ***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. 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.

3) ***FCoE Switches***

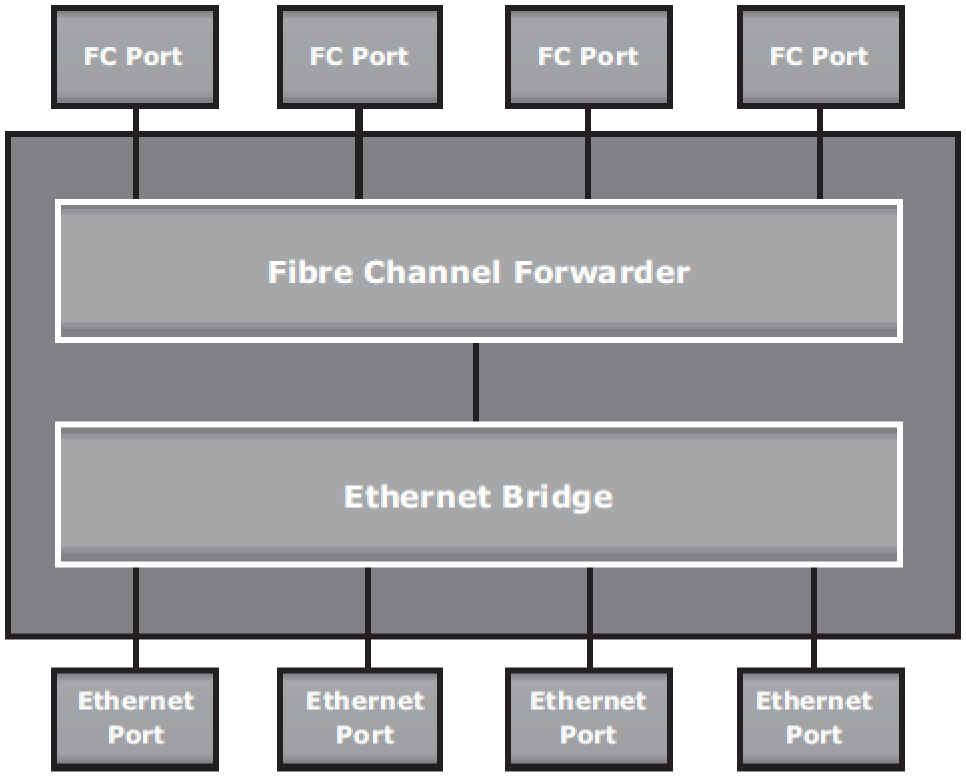


Figure 6-15: FCoE switch generic architecture

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.

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.

FCoE Frame Structure

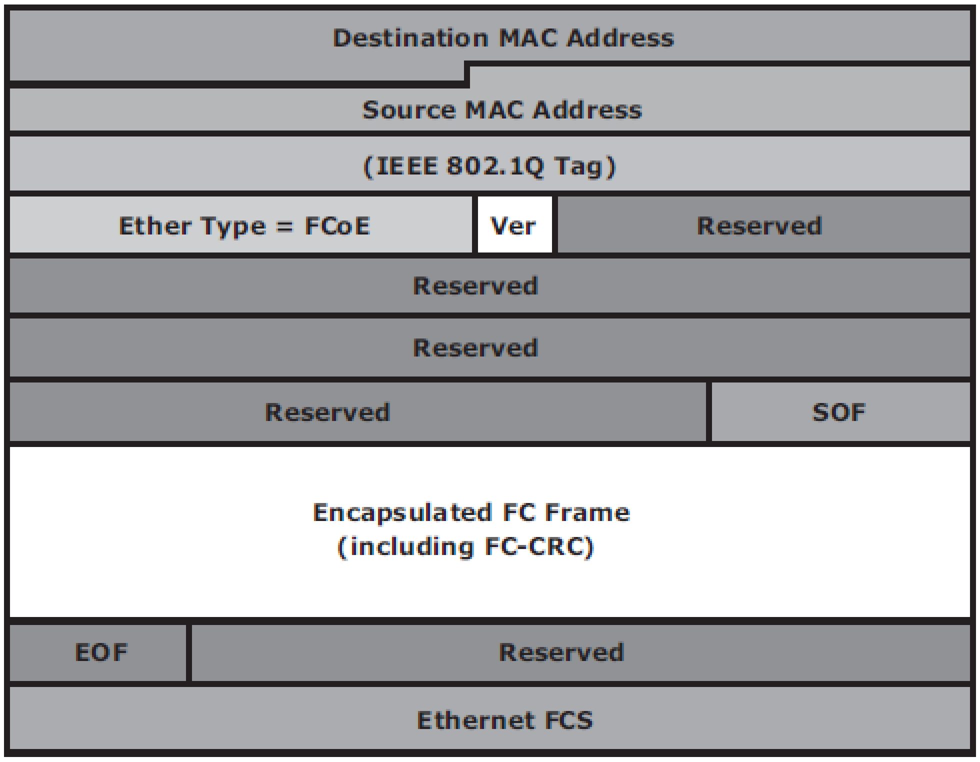


Figure 6-16: FCoE frame structure

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.

The encapsulated Fibre Channel frame consists of the original 24-byte FC header and the data being transported (including the Fibre Channel CRC).

The FC frame structure is maintained such that when a traditional FC SAN is connected to an FCoE capable switch, the FC frame is de-encapsulated from the FCoE frame and transported to FC SAN seamlessly. This capability enables FCoE to integrate with the existing FC SANs without the need for a gateway.

Frame size is also an important factor in FCoE. A typical Fibre Channel data frame has a 2,112-byte payload, a 24-byte header, and an FCS. A standard Ethernet frame has a default payload capacity of 1,500 bytes. To maintain good performance, FCoE must use jumbo frames to prevent a Fibre Channel frame from being split into two Ethernet frames.

FCoE Frame Mapping

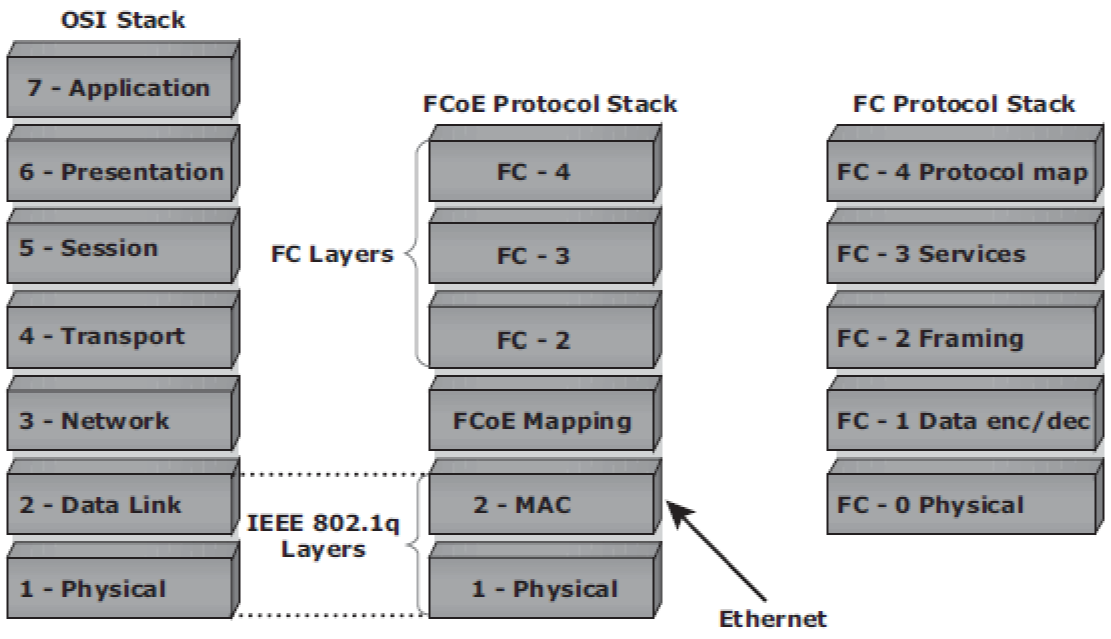


Figure 6-17: FCoE frame mapping

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 five 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. This provides the capability to carry the FC-2 to the FC-4 layer over the Ethernet layer.

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. These functionalities are defined and maintained by the data center bridging (DCB) task group, which is a part of the IEEE 802.1 working group, and they are:

1) Priority-based flow control
2) Enhanced transmission selection
3) Congestion Notification
4) 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.

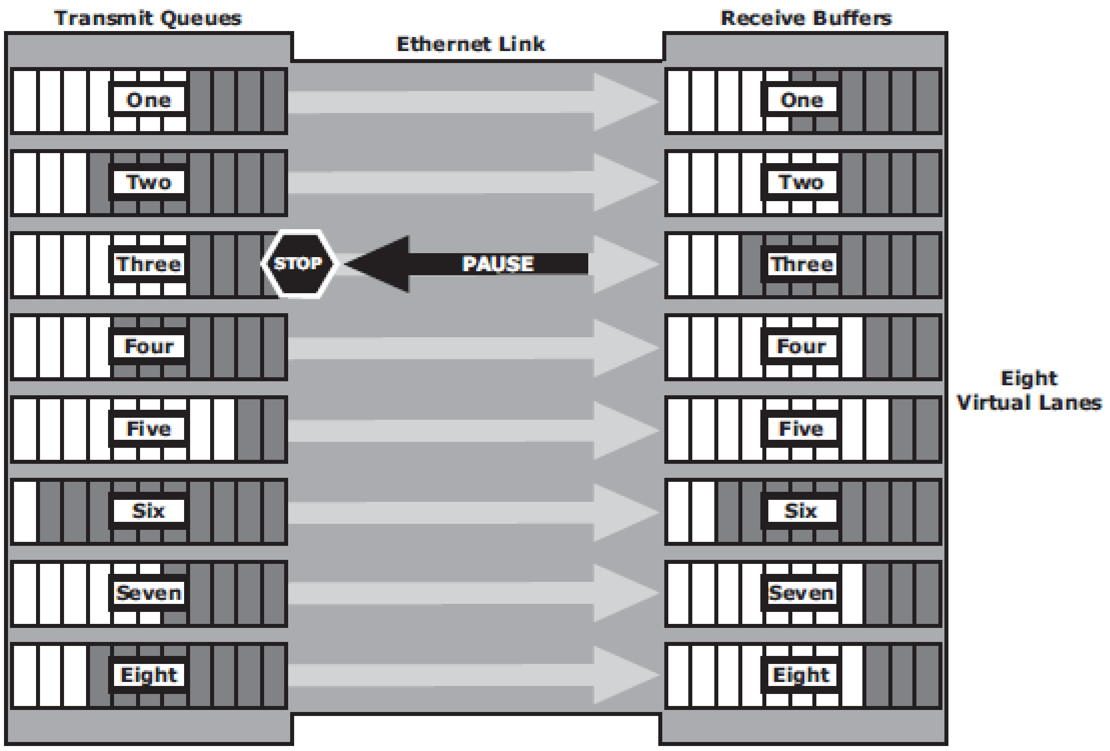


Figure 6-18: Priority-based flow control

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.

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).

If congestion at the receiving end occurs, the algorithm running on the switch generates a congestion notification message to the sending node (Node A). In response to the CN message, the sending end limits the rate of data transfer.

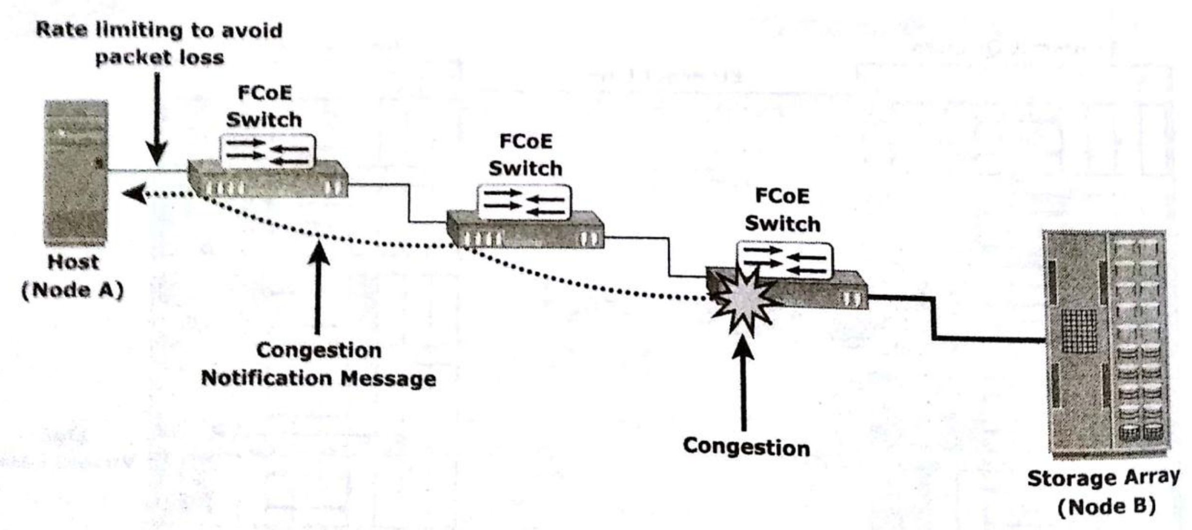


Figure 6-19: Congestion Notification

Data Center Bridging Exchange Protocol (DCBX)

DBBX protocol is a discovery and capability exchange protocol, which helps Converged Enhanced Ethernet devices to convey and configure their features with the other CEE devices in the network.

DCBX is used to negotiate capabilities between the switches and the adapters, and it allows the switch to distribute the configuration values to all the attached adapters. This helps to ensure consistent configuration across the entire network.

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.

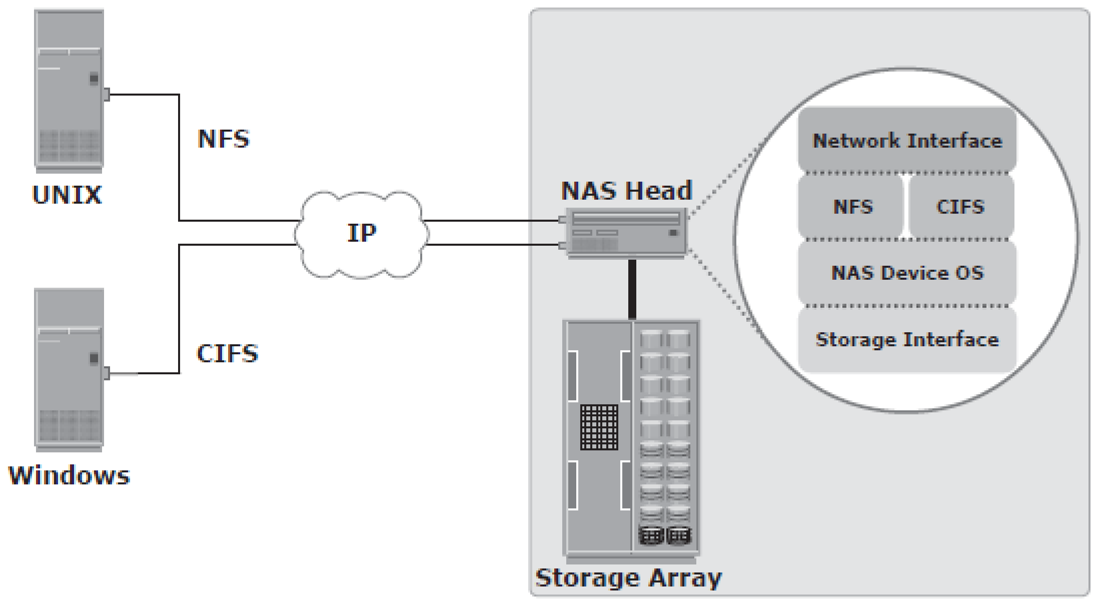


Figure 7-3: Components of NAS

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

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

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.

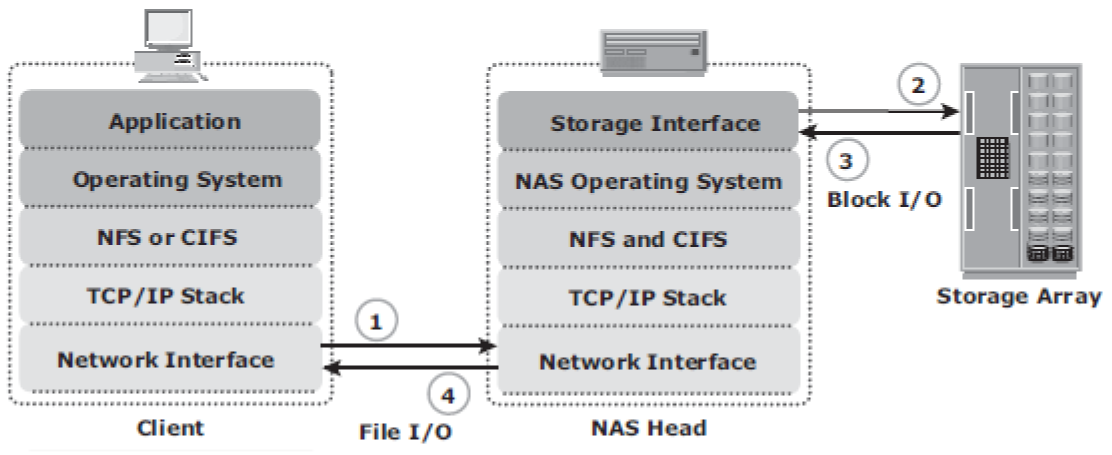


Figure 7-4: NAS I/O operation

NAS File-Sharing Protocols:

Most NAS devices support multiple file-service protocols to handle file I/O requests to a remote file system. As discussed earlier, 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.

1 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 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. NFS (NFSv3 and earlier) is a stateless protocol, which means that it does not maintain any kind of table to store information about open files and associated pointers. Therefore, each call provides a full set of arguments to access files on the server. These arguments include a file handle reference to the file, a particular position to read or write, and the versions of NFS.

Currently, three versions of NFS are in use:

- **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.
- **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 file attributes to reduce refetching.
- 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.

2 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.

The CIFS protocol enables remote clients to gain access to files on a server. CIFS enables file sharing with other clients by using special locks. Filenames in CIFS are encoded using unicode characters. CIFS provides the 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. The fault tolerance features of CIFS depend on whether an application is written to take advantage of these features. Moreover, 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:

\\server\share or \\servername.domain.suffix\share.

File-Level Virtualization

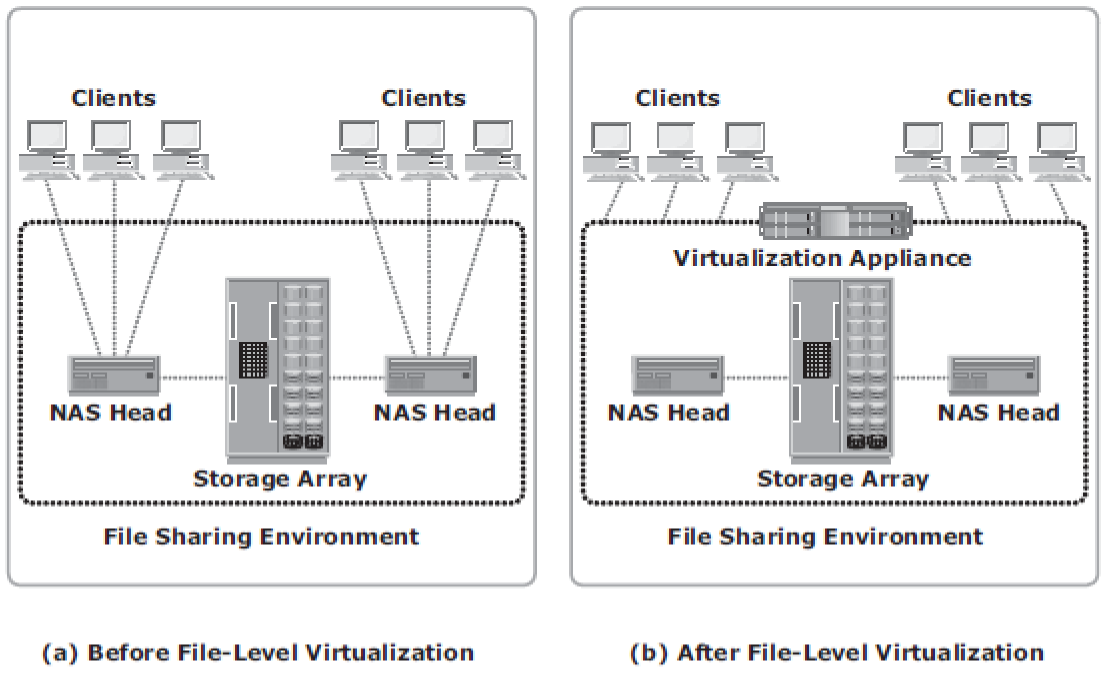


Figure 7-9: File-serving environment before and after file-level virtualization

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.

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).

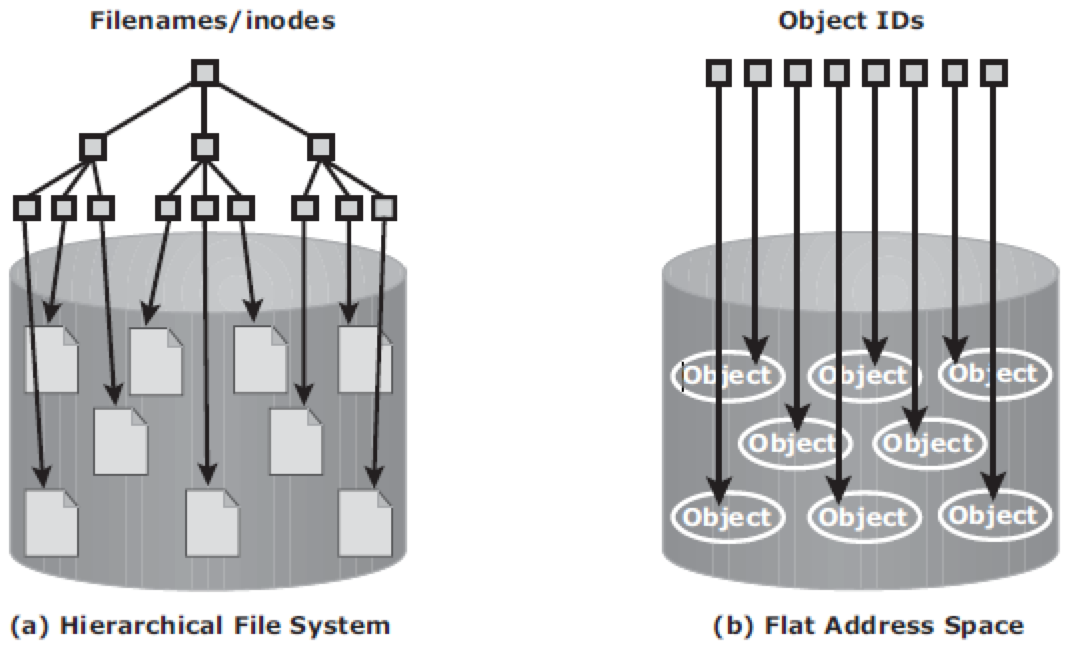


Figure 8-1: Hierarchical file system versus flat address space

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.

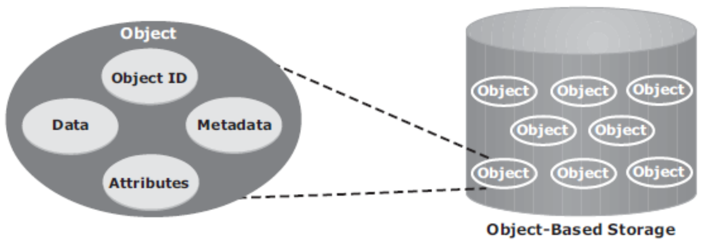


Figure 8-2: Object structure

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.

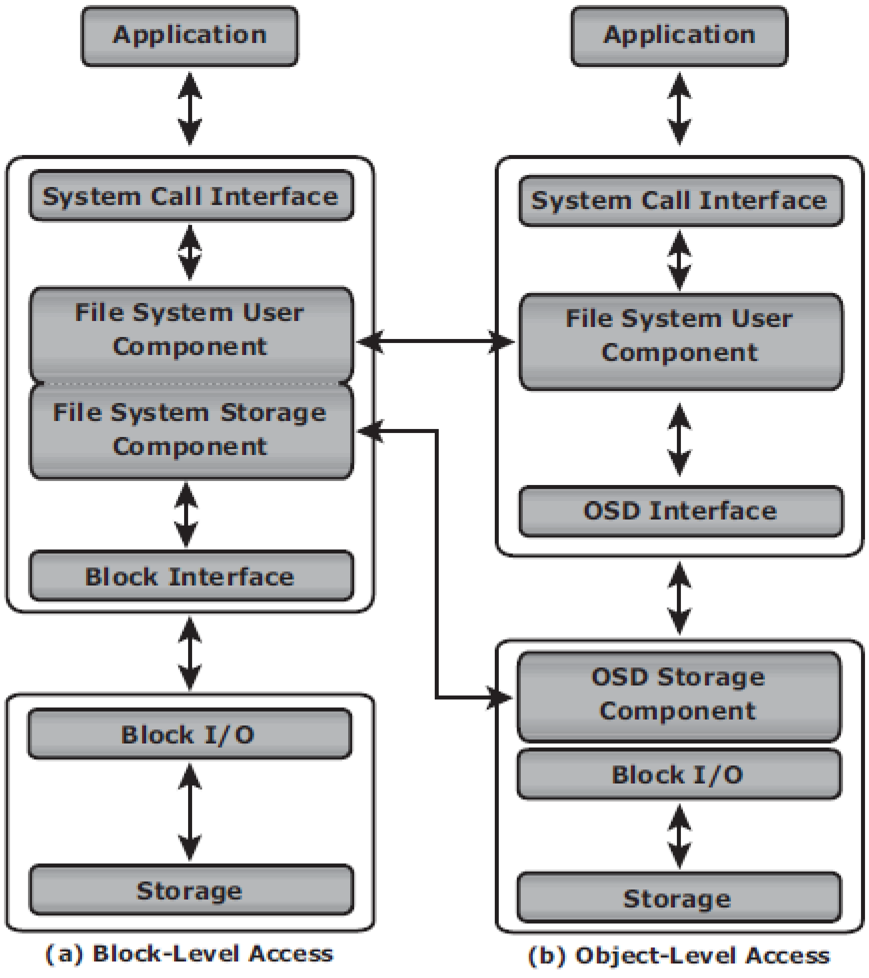


Figure 8-3: Block-level access versus object-level access

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. The OSD storage component manages all the required low-level storage and space management functions. It also manages security and access control functions for the objects.

Components of OSD

The OSD system is typically composed of three key components: nodes, private network, and storage. Figure 8-4 illustrates the components of OSD.

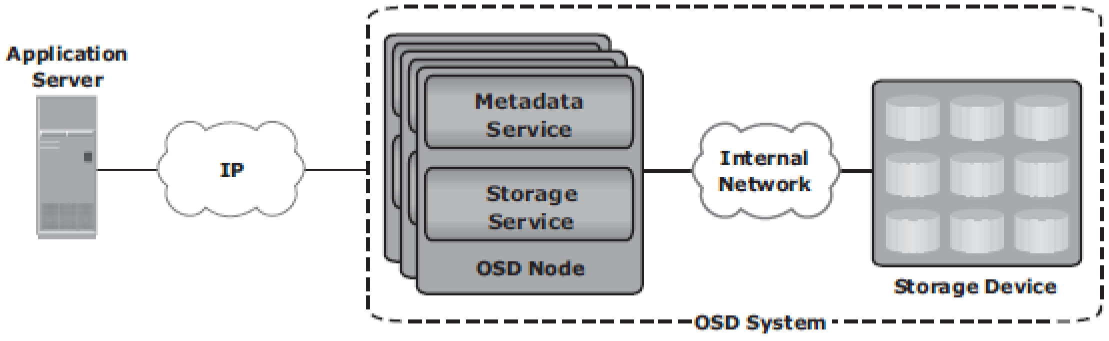


Figure 8-4: OSD components

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. The application server accesses the node to store and retrieve data over an external network.

In some implementations, such as CAS, the metadata service might reside on the application server or on a separate server.

OSD typically uses low-cost and high-density disk drives to store the objects. As more capacity is required, more disk drives can be added to the system.

Object Storage and Retrieval in OSD

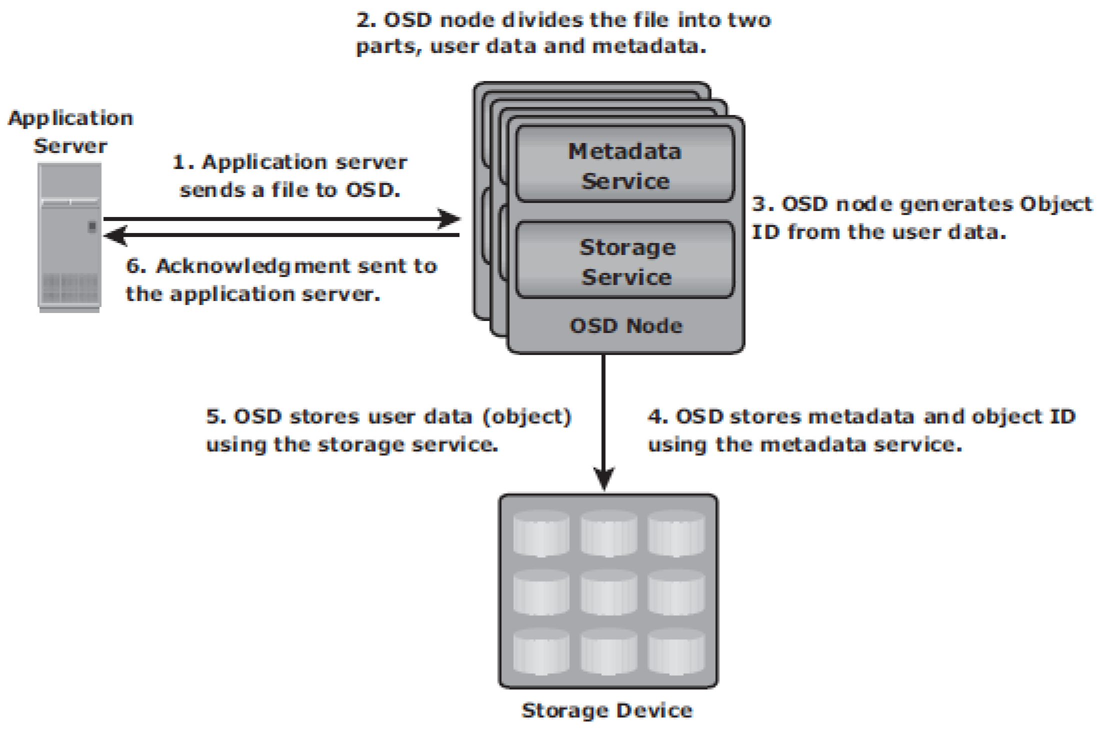

2. OSD node divides the file into two parts, user data and metadata.
Application Server
1. Application server sends a file to OSD.
6. Acknowledgment sent to the application server.
Metadata Service
Storage Service
OSD Node
3. OSD node generates Object ID from the user data.
5. OSD stores user data (object) using the storage service.
4. OSD stores metadata and object ID using the metadata service.
Storage Device

Figure 8-5: Storing objects on 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.

After an object is stored successfully, it is available for retrieval. A user accesses the data stored on OSD by the same filename. The application server retrieves the stored content using the object ID. This process is transparent to the user.

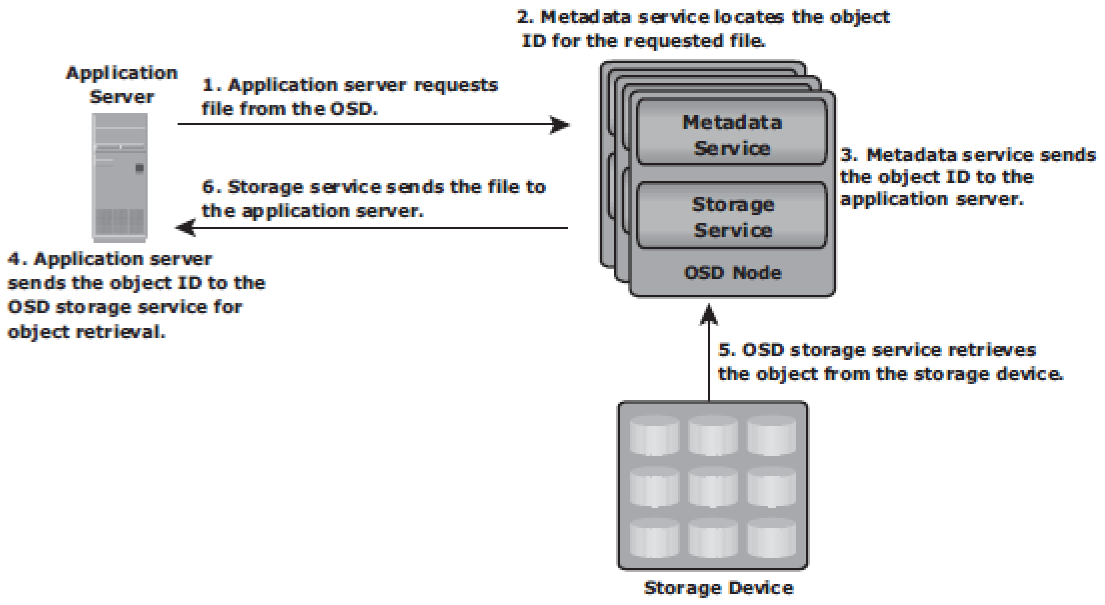

2. Metadata service locates the object ID for the requested file.
Application Server
1. Application server requests file from the OSD.
Metadata Service
3. Metadata service sends the object ID to the application server.
6. Storage service sends the file to the application server.
Storage Service
OSD Node
4. Application server sends the object ID to the OSD storage service for object retrieval.
5. OSD storage service retrieves the object from the storage device.
Storage Device

Figure 8-6: Object retrieval from an OSD system

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.

Benefits of Object-Based Storage:

1. **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.
2. **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.
3. **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.
4. **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. CAS is covered in the following section.

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). REST and SOAP APIs can be easily integrated with business applications that access OSD over the web.

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). This address is derived from the object's binary representation. CAS provides an optimized and centrally managed storage solution. Data access in

CAS differs from other OSD devices. In CAS, the application server access the CAS device only via the CAS API running on the application server. However, the way CAS stores data is similar to the other OSD systems. CAS provides all the features required for storing fixed content.

The key features of CAS are as follows:

1. **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.
2. **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.
3. **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.

4. **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.
5. **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.
6. **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. In the local protection option, data objects are either mirrored or parity protected. In mirror protection, two copies of the data object are stored on two different nodes in the same cluster. This decreases the total available capacity by 50 percent. In parity protection, the data object is split in multiple parts and parity is generated from them. Each part of the data and its parity are stored on a different node. This method consumes less capacity to protect the stored data, but takes slightly longer to regenerate the data if corruption of data occurs. In the remote replication option, data objects are copied to a secondary CAS at the remote location. In this case, the objects remain accessible from the secondary CAS if the primary CAS system fails.
7. **Fast record retrieval:** CAS stores all objects on disks, which provides faster access to the objects compared to tapes and optical discs.
8. **Load balancing:** CAS distributes objects across multiple nodes to provide maximum throughput and availability.
9. **Scalability:** CAS allows the addition of more nodes to the cluster without any interruption to data access and with minimum administrative overhead.
10. **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.
11. **Self diagnosis and repair:** CAS automatically detects 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.
12. **Audit trails:** CAS keeps track of management activities and any access or disposition of data. Audit trails are mandated by compliance requirements.

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.

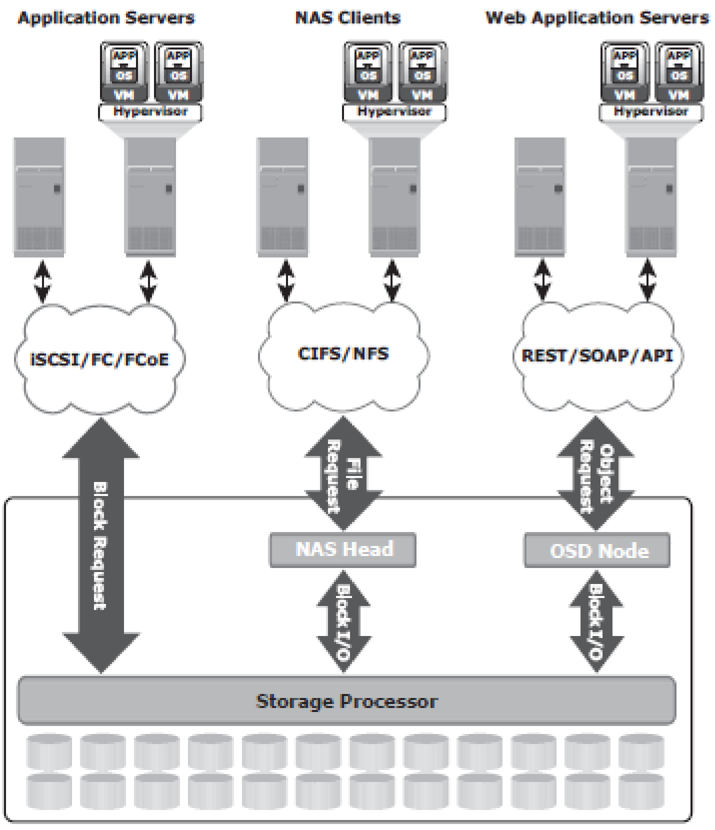


Figure 8-9: Unified storage platform

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.

The LUNs presented to the application server appear as local physical disks. A file system is configured on these LUNs and is made available to applications for storing data.

A NAS head is a dedicated file 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 LUNs presented to the NAS head appear as physical disks. The NAS head configures the file systems on these disks, creates a NFS,CIFS, or mixed share, and exports the share to the NAS clients.

The OSD node accesses the storage through the storage controller using a FC or FCoE connection. The LUNs assigned to the OSD node appear as physical disks. These disks are configured by the OSD nodes, enabling them to store the data from the web application servers.

Data Access from Unified Storage

In a unified storage system, block, file, 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 ormapped) 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. The controller in turn processes the block request and responds back to the OSD node, which in turn provides the requested object to the web application server.