1. Definition of nas, benefits of nas?

It is a ip-based, dedicated, high-performance file sharing & nlw storage.

7.2 Benefits of NAS

NAS offers the following benefits:

- **Comprehensive access to information:** Enables efficient file sharing and supports many-to-one and one-to-many configurations. The many-to-one configuration enables a NAS device to serve many clients simultaneously. The one-to-many configuration enables one client to connect with many NAS devices simultaneously.
- **Improved efficiency:** NAS delivers better performance compared to a general-purpose file server because NAS uses an operating system specialized for file serving.
- **Improved flexibility:** Compatible with clients on both UNIX and Windows platforms using industry-standard protocols. NAS is flexible and can serve requests from different types of clients from the same source.
- **Centralized storage:** Centralizes data storage to minimize data duplication on client workstations, and ensure greater data protection
- **Simplified management:** Provides a centralized console that makes it possible to manage file systems efficiently

- **Scalability:** Scales well with different utilization profiles and types of business applications because of the high-performance and low-latency design
- **High availability:** Offers efficient replication and recovery options, enabling high data availability. NAS uses redundant components that provide maximum connectivity options. A NAS device supports clustering technology for failover.
- **Security:** Ensures security, user authentication, and file locking with industry-standard security schemas
- **Low cost:** NAS uses commonly available and inexpensive Ethernet components.
- **Ease of deployment:** Configuration at the client is minimal, because the clients have required NAS connection software built in.

2. Components of nos :

7.4 Components of NAS

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

- CPU and memory
- One or more network interface cards (NICs), which provide connectivity to the client network. Examples of network protocols supported by NIC include Gigabit Ethernet, Fast Ethernet, ATM, and Fiber Distributed Data Interface (FDDI).
- An optimized operating system for managing the NAS functionality. It translates file-level requests into block-storage requests and further converts the data supplied at the block level to file data.
- NFS, CIFS, and other protocols for file sharing
- Industry-standard storage protocols and ports to connect and manage physical disk resources

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

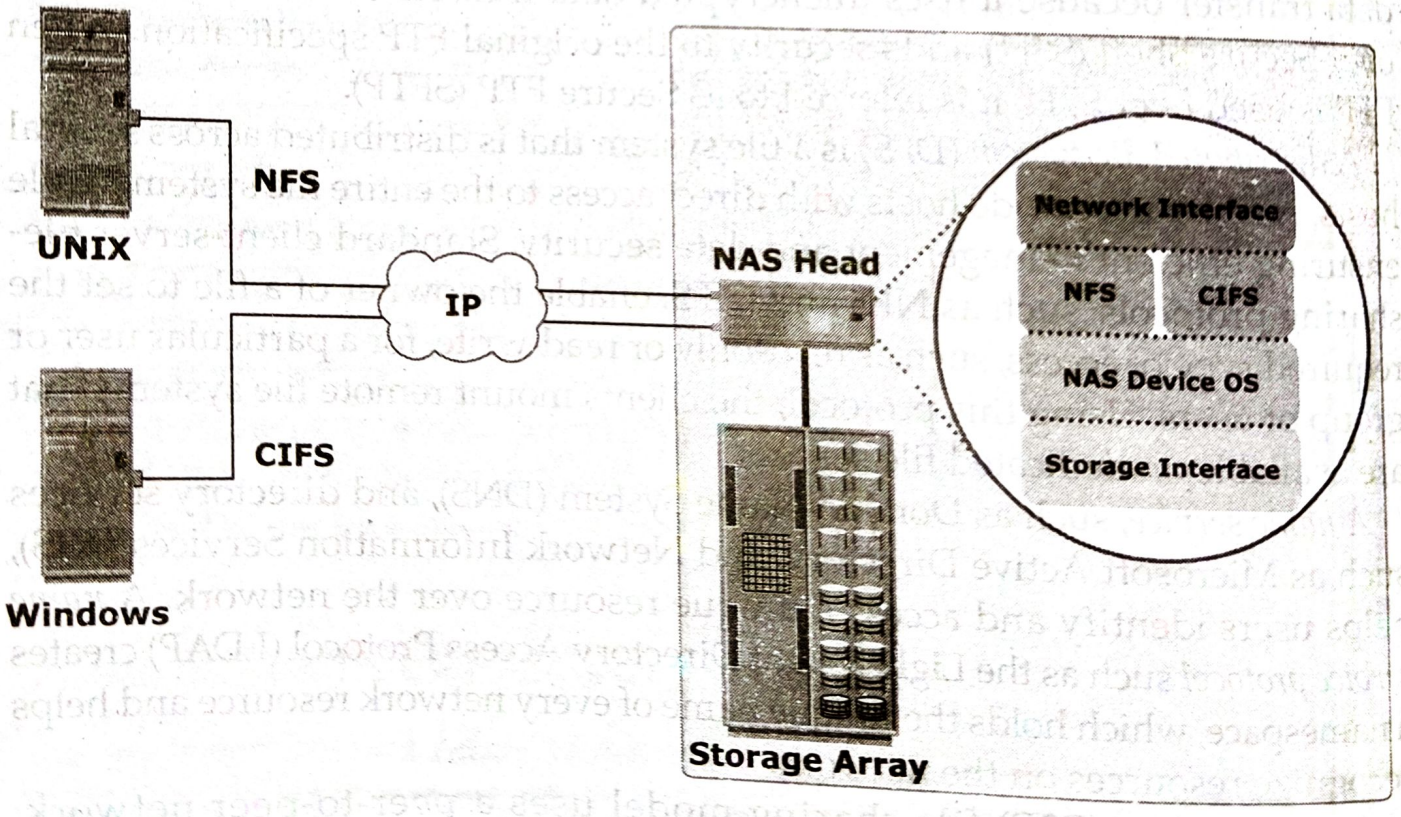


Figure 7-3: Components of NAS

3. implementation of nas ?

7.6 NAS Implementations

Three common NAS implementations are unified, gateway, and scale-out. The *unified* NAS consolidates NAS-based and SAN-based data access within a unified storage platform and provides a unified management interface for managing both the environments.

In a *gateway* implementation, the NAS device uses external storage to store and retrieve data, and unlike unified storage, there are separate administrative tasks for the NAS device and storage.

The *scale-out* NAS implementation pools multiple nodes together in a cluster. A node may consist of either the NAS head or storage or both. The cluster performs the NAS operation as a single entity.

7.6.1 Unified NAS

Unified NAS performs file serving and storing of file data, along with providing access to block-level data. It supports both CIFS and NFS protocols for file access and iSCSI and FC protocols for block level access. Due to consolidation of NAS-based and SAN-based access on a single storage platform, unified NAS reduces an organization's infrastructure and management costs.

A unified NAS contains one or more NAS heads and storage in a single system. NAS heads are connected to the storage controllers (SCs), which provide access to the storage. These storage controllers also provide connectivity to iSCSI and FC hosts. The storage may consist of different drive types, such as SAS, ATA, FC, and flash drives, to meet different workload requirements.

7.6.2 Unified NAS Connectivity

Each NAS head in a unified NAS has front-end Ethernet ports, which connect to the IP network. The front-end ports provide connectivity to the clients and service the file I/O requests. Each NAS head has back-end ports, to provide connectivity to the storage controllers.

iSCSI and FC ports on a storage controller enable hosts to access the storage directly or through a storage network at the block level. Figure 7-5 illustrates an example of unified NAS connectivity.

7.6.3 Gateway NAS

A gateway NAS device consists of one or more NAS heads and uses external and independently managed storage. Similar to unified NAS, the storage is shared with other applications that use block-level I/O. Management functions in this type of solution are more complex than those in a unified NAS environment because there are separate administrative tasks for the NAS head and the storage. A gateway solution can use the FC infrastructure, such as switches and directors for accessing SAN-attached storage arrays or direct-attached storage arrays.

The gateway NAS is more scalable compared to unified NAS because NAS heads and storage arrays can be independently scaled up when required.

For example, NAS heads can be added to scale up the NAS device performance. When the storage limit is reached, it can scale up, adding capacity on the SAN, independent of NAS heads. Similar to a unified NAS, a gateway NAS also enables high utilization of storage capacity by sharing it with the SAN environment.

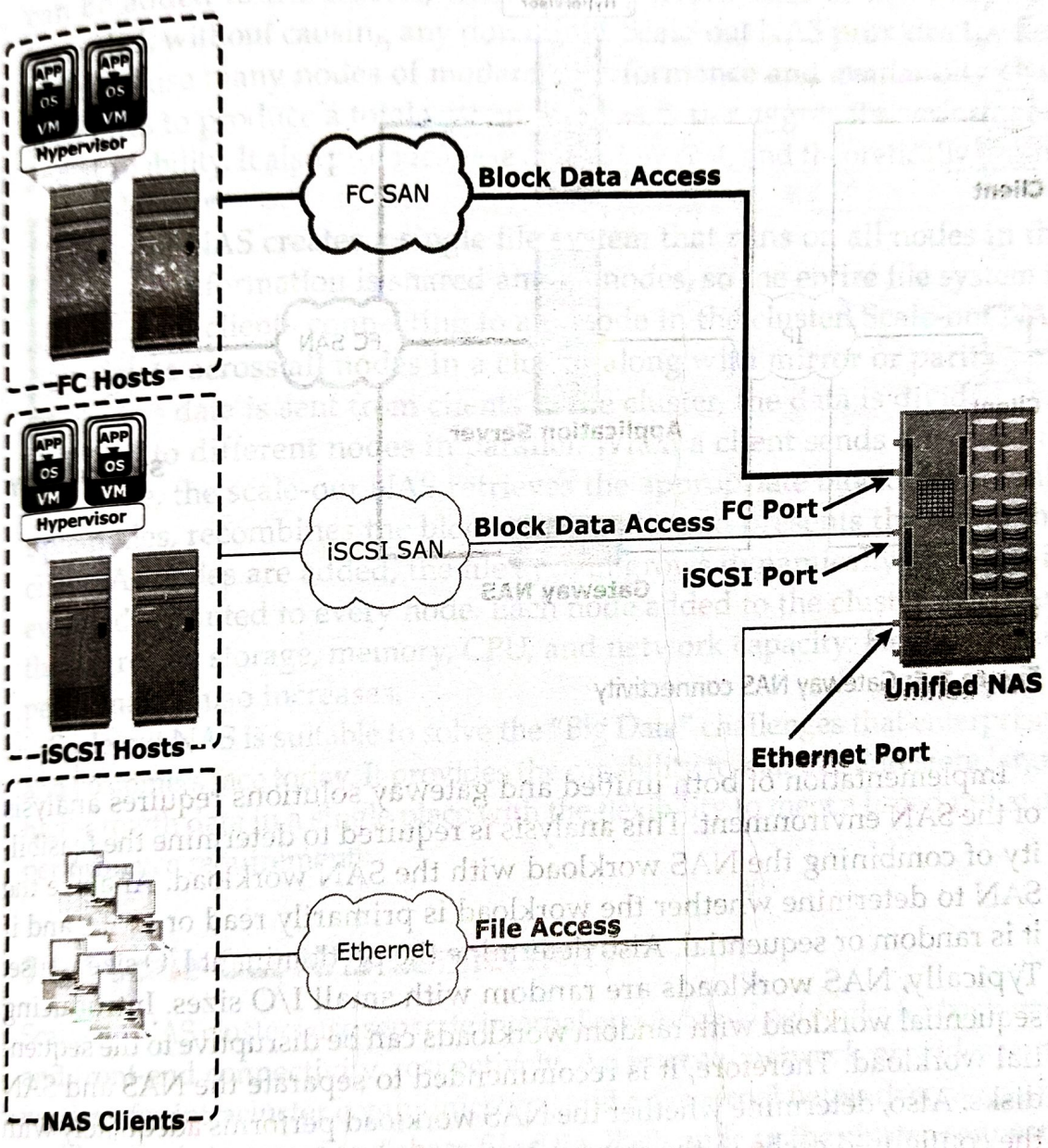


Figure 7-5: Unified NAS connectivity

7.6.4 Gateway NAS Connectivity

In a gateway solution, the front-end connectivity is similar to that in a unified storage solution. Communication between the NAS gateway and the storage system in a gateway solution is achieved through a traditional FC SAN. To deploy a gateway NAS solution, factors, such as multiple paths for data, redundant

fabrics, and load distribution, must be considered. Figure 7-6 illustrates an example of gateway NAS connectivity.

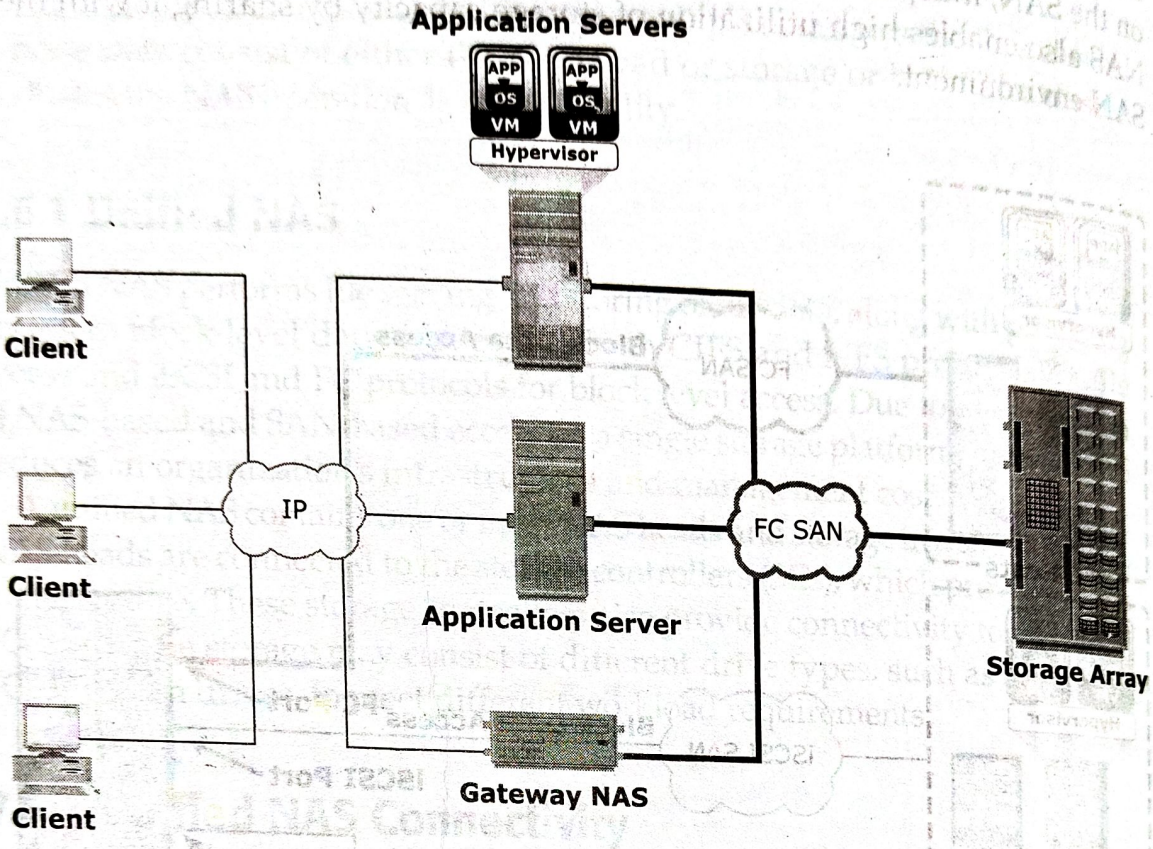


Figure 7-6: Gateway NAS connectivity

Implementation of both unified and gateway solutions requires analysis of the SAN environment. This analysis is required to determine the feasibility of combining the NAS workload with the SAN workload. Analyze the SAN to determine whether the workload is primarily read or write, and if it is random or sequential. Also determine the predominant I/O size in use. Typically, NAS workloads are random with small I/O sizes. Introducing sequential workload with random workloads can be disruptive to the sequential workload. Therefore, it is recommended to separate the NAS and SAN disks. Also, determine whether the NAS workload performs adequately with the configured cache in the storage system.

7.6.5 Scale-Out NAS

Both unified and gateway NAS implementations provide the capability to scale-up their resources based on data growth and rise in performance requirements. Scaling up these NAS devices involves adding CPUs, memory, and storage to

the NAS device. Scalability is limited by the capacity of the NAS device to house and use additional NAS heads and storage.

Scale-out NAS enables grouping multiple nodes together to construct a clustered NAS system. A scale-out NAS provides the capability to scale its resources by simply adding nodes to a clustered NAS architecture. The cluster works as a single NAS device and is managed centrally. Nodes can be added to the cluster, when more performance or more capacity is needed, without causing any downtime. Scale-out NAS provides the flexibility to use many nodes of moderate performance and availability characteristics to produce a total system that has better aggregate performance and availability. It also provides ease of use, low cost, and theoretically unlimited scalability.

Scale-out NAS creates a single file system that runs on all nodes in the cluster. All information is shared among nodes, so the entire file system is accessible by clients connecting to any node in the cluster. Scale-out NAS stripes data across all nodes in a cluster along with mirror or parity protection. As data is sent from clients to the cluster, the data is divided and allocated to different nodes in parallel. When a client sends a request to read a file, the scale-out NAS retrieves the appropriate blocks from multiple nodes, recombines the blocks into a file, and presents the file to the client. As nodes are added, the file system grows dynamically and data is evenly distributed to every node. Each node added to the cluster increases the aggregate storage, memory, CPU, and network capacity. Hence, cluster performance also increases.

Scale-out NAS is suitable to solve the "Big Data" challenges that enterprises and customers face today. It provides the capability to manage and store large, high-growth data in a single place with the flexibility to meet a broad range of performance requirements.

7.6.6 Scale-Out NAS Connectivity

Scale-out NAS clusters use separate internal and external networks for back-end and front-end connectivity, respectively. An internal network provides connections for intracluster communication, and an external network connection enables clients to access and share file data. Each node in the cluster connects to the internal network. The internal network offers high throughput and low latency and uses high-speed networking technology, such as InfiniBand or Gigabit Ethernet. To enable clients to access a node, the node must be connected to the external Ethernet network. Redundant internal or external networks may be used for high availability. Figure 7-7 illustrates an example of scale-out NAS connectivity.

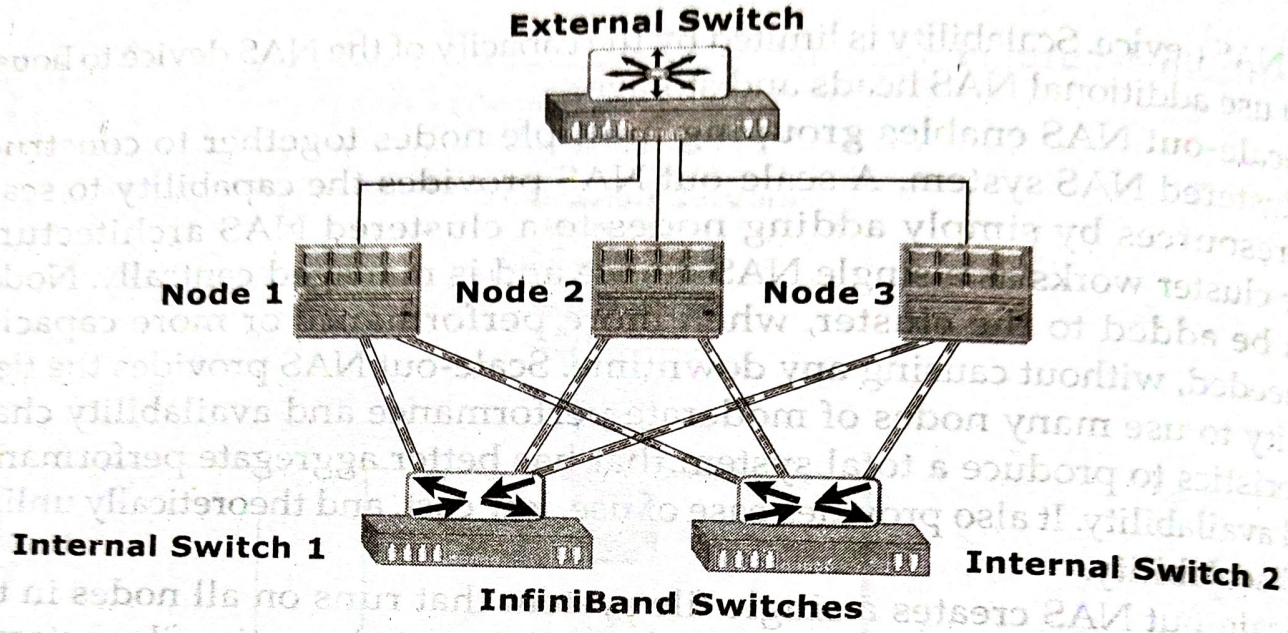


Figure 7-7: Scale-out NAS with dual internal and single external networks

4. Illustrate with neat I/O operations.

7.5 NAS I/O Operation

NAS provides file-level data access to its clients. File I/O is a high-level request that specifies the file to be accessed. For example, a client may request a file by specifying its name, location, or other attributes. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrieve data. The process of handling I/Os in a NAS environment is as follows:

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

Figure 7-4 illustrates this process.

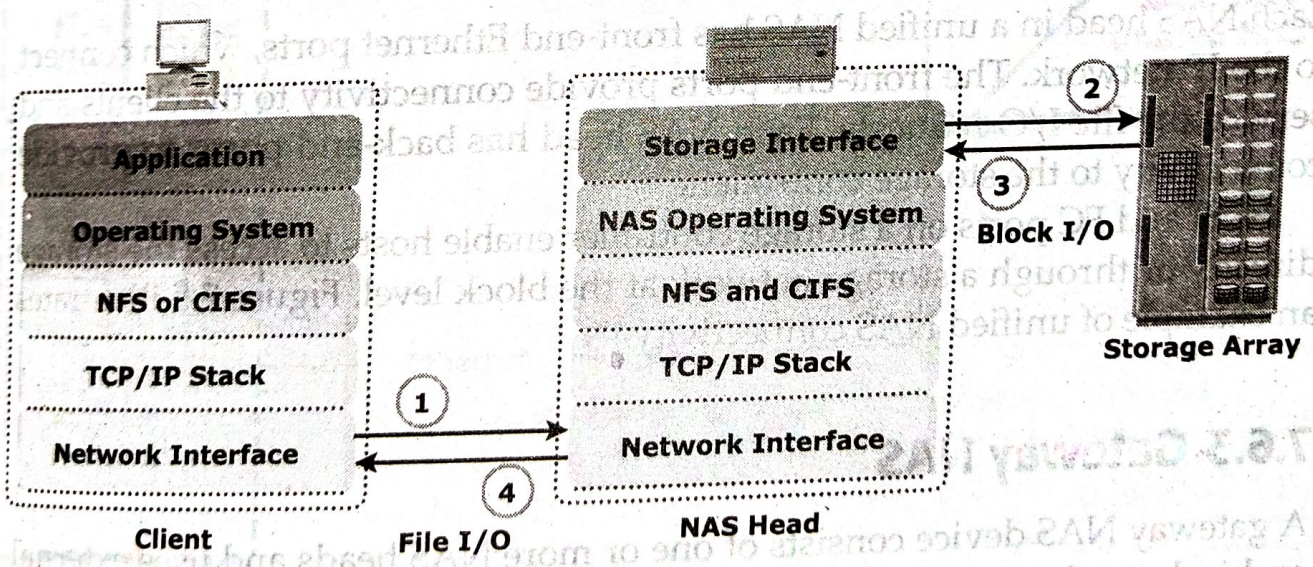


Figure 7-4: NAS I/O operation

5. protocols (NFS, CIFS)

7.7 NAS File-Sharing Protocols

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

7.7.1 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:

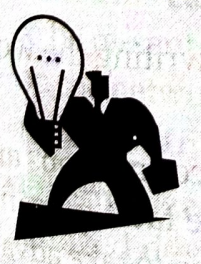
- Searching files and directories
- Opening, reading, writing to, and closing a file
- Changing file attributes
- Modifying file links and directories

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.

PNFS AND MPFS



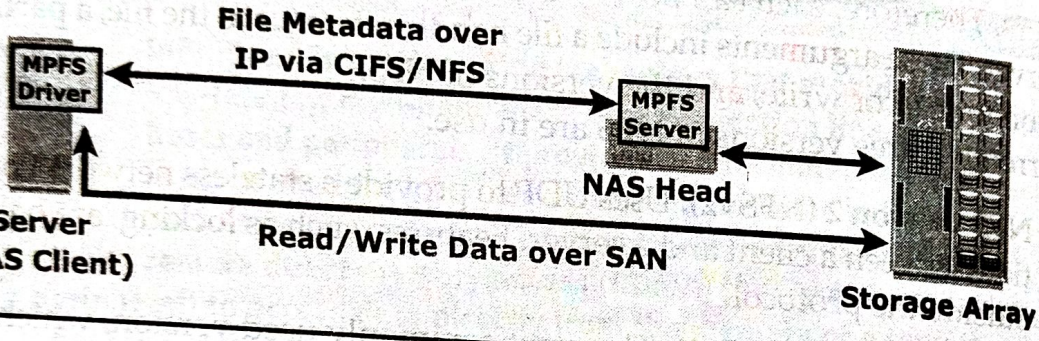
pNFS, as part of NFSv4.1, separates the file system protocol processing into two parts: metadata processing and data processing. The metadata includes information about a file system object, such as its name, location within the namespace, owner, access control list (ACL), and other attributes. The pNFS server, also called a metadata server,

(Continued)

PNFS AND MPFS *(continued)*

does the metadata processing and is kept out of the data path. pNFS clients send the metadata information to the pNFS server. The pNFS clients access storage devices directly using multiple parallel data paths. The pNFS client uses a storage network protocol, such as iSCSI or FC, to perform I/O to storage devices. The pNFS clients get information about the storage devices from the metadata server. Because the pNFS server is relieved of data processing and pNFS clients can access the storage devices directly using parallel paths, the pNFS mechanism significantly improves the pNFS client performance.

The EMC-patented Multi-Path File System (MPFS) protocol works similar to pNFS. The MPFS driver software, installed at the NAS clients, sends the file's metadata to the NAS device (MPFS server) via the IP network. The MPFS driver obtains information about the location of the data from the NAS device over the IP network. After knowing the data location, the MPFS driver communicates directly to the storage devices and enables the NAS clients to access the data over SAN. The following Figure shows the MPFS architecture that provides different paths for transferring a file's metadata and data.



7.7.2 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`.

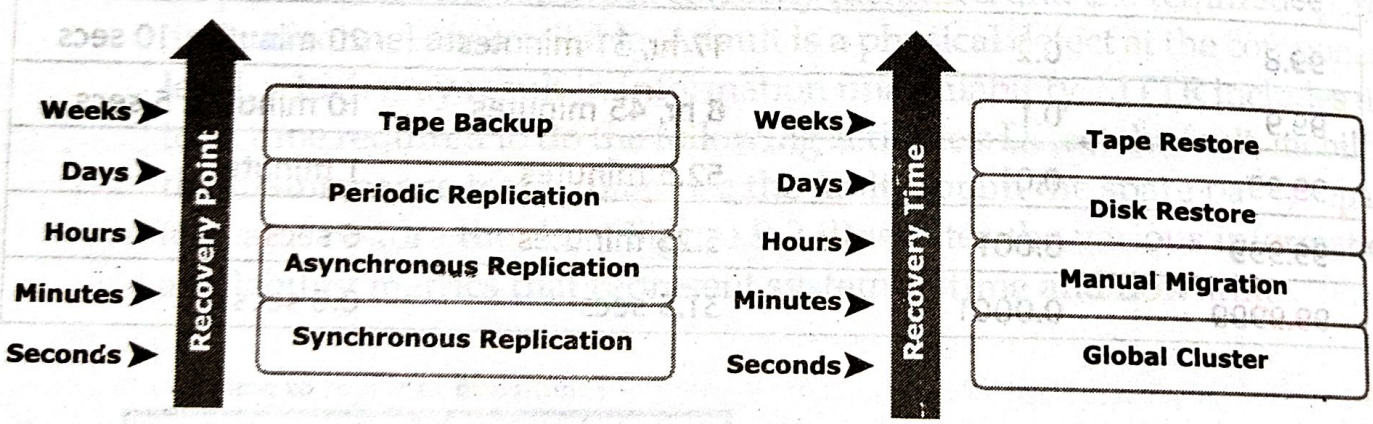
6. Explain business continuity terminology, also explain business continuity planning life cycle.

9.2 BC Terminology

This section introduces and defines common terms related to BC operations, which are used in the next few chapters to explain advanced concepts:

- **Disaster recovery:** This is the coordinated process of restoring systems, data, and the infrastructure required to support ongoing business operations after a disaster occurs. It is the process of restoring a previous copy of the data and applying logs or other necessary processes to that copy to bring it to a known point of consistency. After all recovery efforts are completed, the data is validated to ensure that it is correct.

- **Disaster restart:** This is the process of restarting business operations with mirrored consistent copies of data and applications.
- **Recovery-Point Objective (RPO):** This is the point in time to which systems and data must be recovered after an outage. It defines the amount of data loss that a business can endure. A large RPO signifies high tolerance to information loss in a business. Based on the RPO, organizations plan for the frequency with which a backup or replica must be made. For example, if the RPO is 6 hours, backups or replicas must be made at least once in 6 hours. Figure 9-3 (a) shows various RPOs and their corresponding ideal recovery strategies. An organization can plan for an appropriate BC technology solution on the basis of the RPO it sets. For example:
  - **RPO of 24 hours:** Backups are created at an offsite tape library every midnight. The corresponding recovery strategy is to restore data from the set of last backup tapes.
  - **RPO of 1 hour:** Shipping database logs to the remote site every hour. The corresponding recovery strategy is to recover the database to the point of the last log shipment.
  - **RPO in the order of minutes:** Mirroring data asynchronously to a remote site
  - **Near zero RPO:** Mirroring data synchronously to a remote site



(a) Recovery-point objective

(b) Recovery-time objective

Figure 9-3: Strategies to meet RPO and RTO targets

- **Recovery-Time Objective (RTO):** The time within which systems and applications must be recovered after an outage. It defines the amount of downtime that a business can endure and survive. Businesses can optimize disaster recovery plans after defining the RTO for a given system. For example, if the RTO is 2 hours, it requires disk-based backup because it enables a faster restore than a tape backup. However, for an RTO of 1 week, tape backup will likely meet the requirements. Some examples

of RTOs and the recovery strategies to ensure data availability are listed here (refer to Figure 9-3 [b]):

- **RTO of 72 hours:** Restore from tapes available at a cold site.
- **RTO of 12 hours:** Restore from tapes available at a hot site.
- **RTO of few hours:** Use of data vault at a hot site
- **RTO of a few seconds:** Cluster production servers with bidirectional mirroring, enabling the applications to run at both sites simultaneously.

- **Data vault:** A repository at a remote site where data can be periodically or continuously copied (either to tape drives or disks) so that there is always a copy at another site
- **Hot site:** A site where an enterprise's operations can be moved in the event of disaster. It is a site with the required hardware, operating system, application, and network support to perform business operations, where the equipment is available and running at all times.
- **Cold site:** A site where an enterprise's operations can be moved in the event of disaster, with minimum IT infrastructure and environmental facilities in place, but not activated
- **Server Clustering:** A group of servers and other necessary resources coupled to operate as a single system. Clusters can ensure high availability and load balancing. Typically, in failover clusters, one server runs an application and updates the data, and another server is kept as standby to take over completely, as required. In more sophisticated clusters, multiple servers may access data, and typically one server is kept as standby. Server clustering provides load balancing by distributing the application load evenly among multiple servers within the cluster.

9.3 BC Planning Life Cycle

BC planning must follow a disciplined approach like any other planning process. Organizations today dedicate specialized resources to develop and maintain BC plans. From the conceptualization to the realization of the BC plan, a life cycle of activities can be defined for the BC process. The BC planning life cycle includes five stages (see Figure 9-4):

1. Establishing objectives
2. Analyzing
3. Designing and developing
4. Implementing
5. Training, testing, assessing, and maintaining

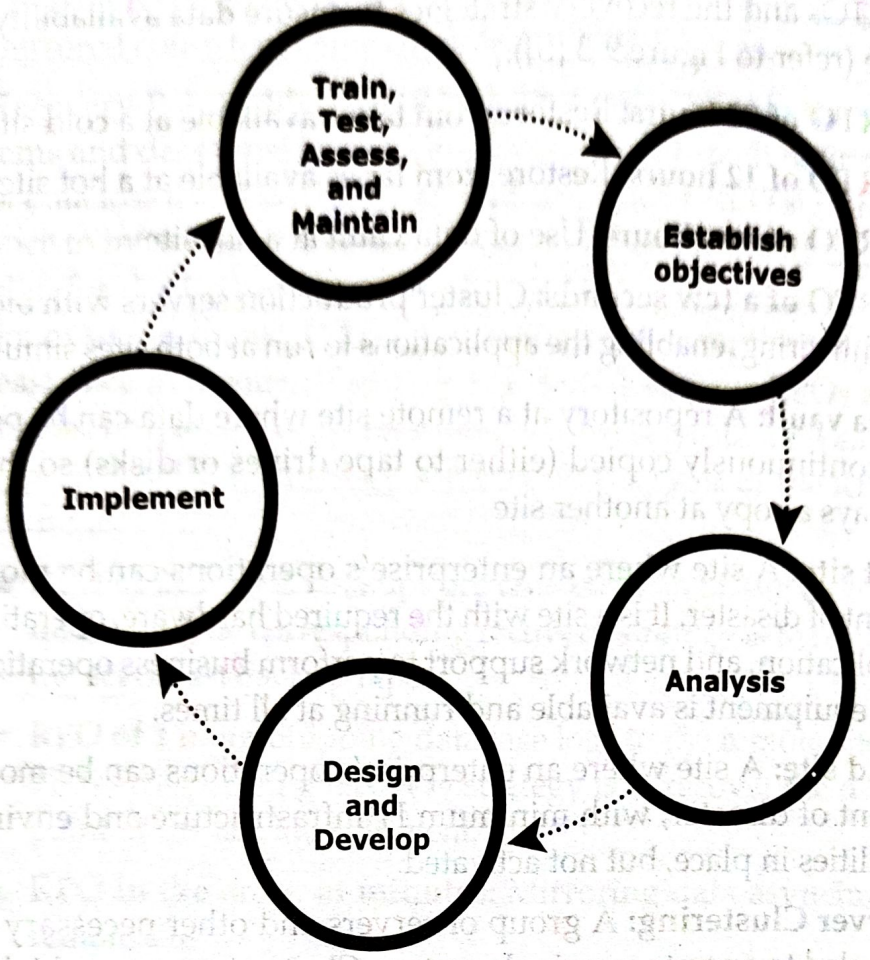


Figure 9-4: BC planning life cycle

Several activities are performed at each stage of the BC planning life cycle, including the following key activities:

1. Establish objectives:
   - Determine BC requirements.
   - Estimate the scope and budget to achieve requirements.
   - Select a BC team that includes subject matter experts from all areas of the business, whether internal or external.
   - Create BC policies.
2. Analysis:
   - Collect information on data profiles, business processes, infrastructure support, dependencies, and frequency of using business infrastructure.
   - Conduct a Business Impact Analysis (BIA).
   - Identify critical business processes and assign recovery priorities.
   - Perform risk analysis for critical functions and create mitigation strategies.

- Perform cost benefit analysis for available solutions based on the mitigation strategy.
- Evaluate options.

3. Design and develop:
   - Define the team structure and assign individual roles and responsibilities. For example, different teams are formed for activities, such as emergency response, damage assessment, and infrastructure and application recovery.
   - Design data protection strategies and develop infrastructure.
   - Develop contingency solutions.
   - Develop emergency response procedures.
   - Detail recovery and restart procedures.
4. Implement:
   - Implement risk management and mitigation procedures that include backup, replication, and management of resources.
   - Prepare the disaster recovery sites that can be utilized if a disaster affects the primary data center.
   - Implement redundancy for every resource in a data center to avoid single points of failure.
5. Train, test, assess, and maintain:
   - Train the employees who are responsible for backup and replication of business-critical data on a regular basis or whenever there is a modification in the BC plan.
   - Train employees on emergency response procedures when disasters are declared.
   - Train the recovery team on recovery procedures based on contingency scenarios.
   - Perform damage-assessment processes and review recovery plans.
   - Test the BC plan regularly to evaluate its performance and identify its limitations.
   - Assess the performance reports and identify limitations.
   - Update the BC plans and recovery/restart procedures to reflect regular changes within the data center.

7. Explain failure analysis & the resolving measures taken for the failure?

9.4 Failure Analysis

Failure analysis involves analyzing both the physical and virtual infrastructure components to identify systems that are susceptible to a single point of failure and implementing fault-tolerance mechanisms.

9.4.1 Single Point of Failure

A *single point of failure* refers to the failure of a component that can terminate the availability of the entire system or IT service. Figure 9-5 depicts a system setup in which an application, running on a VM, provides an interface to the client and performs I/O operations. The client is connected to the server through an IP network, and the server is connected to the storage array through an FC connection.

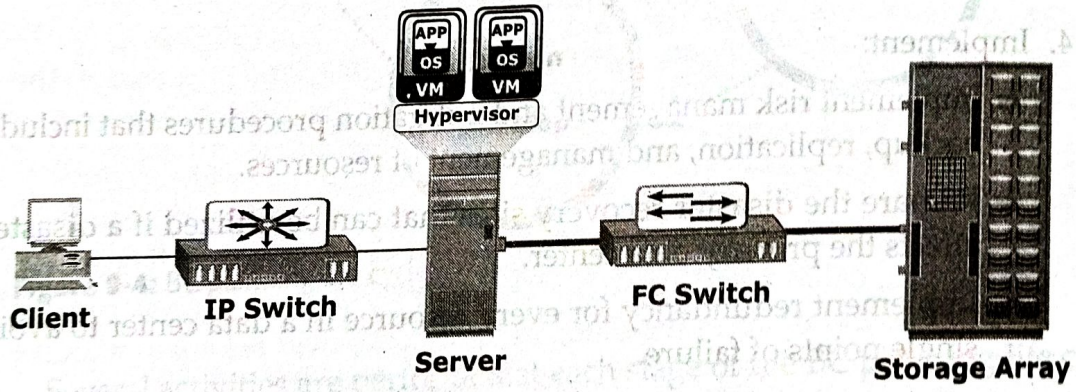


Figure 9-5: Single point of failure

In a setup in which each component must function as required to ensure data availability, the failure of a single physical or virtual component causes the unavailability of an application. This failure results in disruption of business operations. For example, failure of a hypervisor can affect all the running VMs and the virtual network, which are hosted on it. In the setup shown in Figure 9-5, several single points of failure can be identified. A VM, a hypervisor, an HBA/NIC on the server, the physical server, the IP network, the FC switch, the storage array ports, or even the storage array could be a potential single point of failure.

9.4.2 Resolving Single Points of Failure

To mitigate single points of failure, systems are designed with redundancy, such that the system fails only if all the components in the redundancy group fail. This ensures that the failure of a single component does not affect data availability. Data centers follow stringent guidelines to implement fault tolerance for uninterrupted information availability. Careful analysis is performed to eliminate every single point of failure. The example shown in Figure 9-6 represents all enhancements in the infrastructure to mitigate single points of failure:

- Configuration of redundant HBAs at a server to mitigate single HBA failure
- Configuration of NIC teaming at a server allows protection against single physical NIC failure. It allows grouping of two or more physical NICs and treating them as a single logical device. With NIC teaming, if one of the underlying physical NICs fails or its cable is unplugged, the traffic is redirected to another physical NIC in the team. Thus, NIC teaming eliminates the single point of failure associated with a single physical NIC.
- Configuration of redundant switches to account for a switch failure
- Configuration of multiple storage array ports to mitigate a port failure
- RAID and hot spare configuration to ensure continuous operation in the event of disk failure
- Implementation of a redundant storage array at a remote site to mitigate local site failure
- Implementing server (or compute) clustering, a fault-tolerance mechanism whereby two or more servers in a cluster access the same set of data volumes. Clustered servers exchange a *heartbeat* to inform each other about their health. If one of the servers or hypervisors fails, the other server or hypervisor can take up the workload.
- Implementing a VM Fault Tolerance mechanism ensures BC in the event of a server failure. This technique creates duplicate copies of each VM on another server so that when a VM failure is detected, the duplicate VM can be used for failover. The two VMs are kept in synchronization with each other in order to perform successful failover.

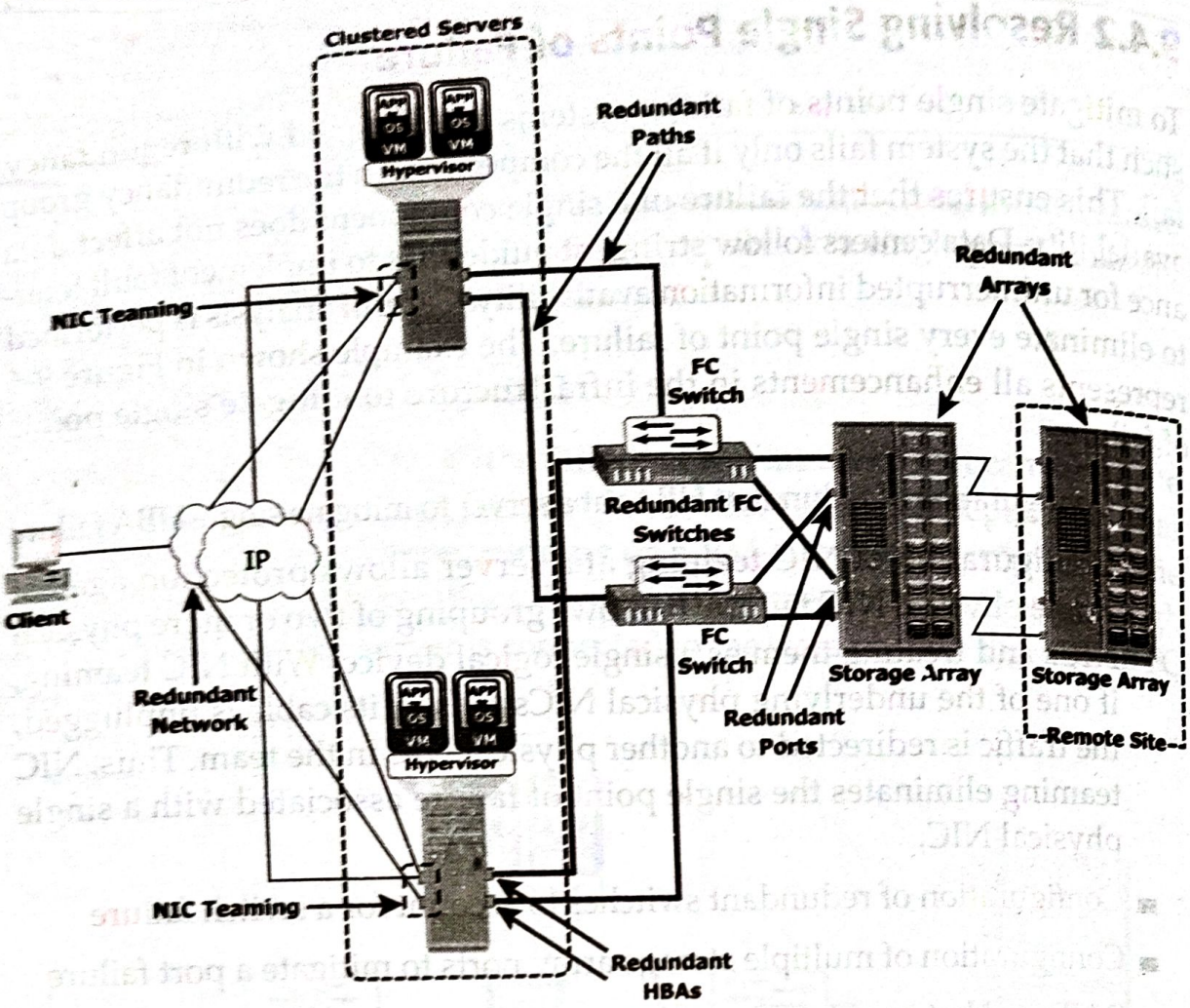


Figure 9-6: Resolving single points of failure

9.4.3 Multipathing Software

Configuration of multiple paths increases the data availability through path failover. If servers are configured with one I/O path to the data, there will be no access to the data if that path fails. Redundant paths to the data eliminate the possibility of the path becoming a single point of failure. Multiple paths to data also improve I/O performance through load balancing among the paths and maximize server, storage, and data path utilization.

In practice, merely configuring multiple paths does not serve the purpose. Even with multiple paths, if one path fails, I/O does not reroute unless the system recognizes that it has an alternative path. Multipathing software provides the functionality to recognize and utilize alternative I/O paths to data. Multipathing software also manages the load balancing by distributing I/Os to all available, active paths.

Multipathing software intelligently manages the paths to a device by sending I/O down the optimal path based on the load balancing and failover policy setting for the device. It also takes into account path usage and availability before deciding the path through which to send the I/O. If a path to the device fails, it automatically reroutes the I/O to an alternative path.

In a virtual environment, multipathing is enabled either by using the hypervisor's built-in capability or by running a third-party software module, added to the hypervisor.

8. Business impact analysis

9.5 Business Impact Analysis

A *business impact analysis* (BIA) identifies which business units, operations, and processes are essential to the survival of the business. It evaluates the financial, operational, and service impacts of a disruption to essential business processes. Selected functional areas are evaluated to determine resilience of the infrastructure to support information availability. The BIA process leads to a report detailing the incidents and their impact over business functions. The impact may be specified in terms of money or in terms of time. Based on the potential impacts associated with downtime, businesses can prioritize and implement countermeasures to mitigate the likelihood of such disruptions. These are detailed in the BC plan. A BIA includes the following set of tasks:

- Determine the business areas.
- For each business area, identify the key business processes critical to its operation.
- Determine the attributes of the business process in terms of applications, databases, and hardware and software requirements.
- Estimate the costs of failure for each business process.
- Calculate the maximum tolerable outage and define RTO and RPO for each business process.
- Establish the minimum resources required for the operation of business processes.
- Determine recovery strategies and the cost for implementing them.
- Optimize the backup and business recovery strategy based on business priorities.
- Analyze the current state of BC readiness and optimize future BC planning.

9. BC technology solutions.

9.6 BC Technology Solutions

After analyzing the business impact of an outage, designing the appropriate solutions to recover from a failure is the next important activity. One or more copies of the data are maintained using any of the following strategies so that

data can be recovered or business operations can be restarted using an alternative copy:

- **Backup:** Data backup is a predominant method of ensuring data availability. The frequency of backup is determined based on RPO, RTO, and the frequency of data changes.
- **Local replication:** Data can be replicated to a separate location within the same storage array. The replica is used independently for other business operations. Replicas can also be used for restoring operations if data corruption occurs.
- **Remote replication:** Data in a storage array can be replicated to another storage array located at a remote site. If the storage array is lost due to a disaster, business operations can be started from the remote storage array.

10. Describe the components of FCOE with FCOE frame mapping?

6.3 FCoE

Data centers typically have multiple networks to handle various types of I/O traffic — for example, an Ethernet network for TCP/IP communication and an FC network for FC communication. TCP/IP is typically used for client-server communication, data backup, infrastructure management communication, and so on. FC is typically used for moving block-level data between storage and servers. To support multiple networks, servers in a data center are equipped with multiple redundant physical network interfaces — for example, multiple Ethernet and FC cards/adapters. In addition, to enable the communication, different types of networking switches and physical cabling infrastructure are implemented in data centers. The need for two different kinds of physical network infrastructure increases the overall cost and complexity of data center operation.

Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffic over a single physical interface infrastructure. FCoE helps organizations address the challenges of having multiple discrete network infrastructures. FCoE uses the Converged Enhanced Ethernet (CEE) link (10 Gigabit Ethernet) to send FC frames over Ethernet.

6.3.2 Components of an FCoE Network

This section describes the key physical components required to implement FCoE in a data center. The key FCoE components are:

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

Converged Network Adapter

A CNA provides the functionality of both a standard NIC and an FC HBA in a single adapter and consolidates both types of traffic. CNA eliminates the need to deploy separate adapters and cables for FC and Ethernet communications, thereby reducing the required number of server slots and switch ports. CNA offloads the FCoE protocol processing task from the server, thereby freeing the server CPU resources for application processing. As shown in Figure 6-14, a CNA contains separate modules for 10 Gigabit Ethernet, Fibre Channel, and FCoE Application Specific Integrated Circuits (ASICs). The FCoE ASIC encapsulates FC frames into Ethernet frames. One end of this ASIC is connected to 10GbE and FC ASICs for server connectivity, while the other end provides a 10GbE interface to connect to an FCoE switch.

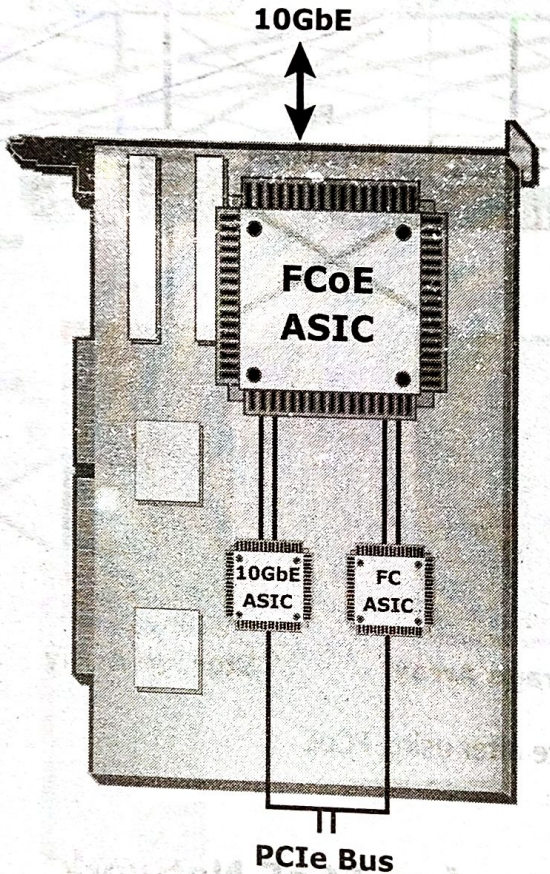
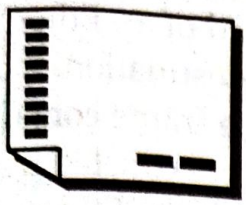


Figure 6-14: Converged Network Adapter

Cables

Currently two options are available for FCoE cabling: Copper based Twinax and standard fiber optical cables. A Twinax cable is composed of two pairs of copper cables covered with a shielded casing. The Twinax cable can transmit data at the speed of 10 Gbps over shorter distances up to 10 meters. Twinax cables require less power and are less expensive than fiber optic cables. 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.



A typical strategy for FCoE deployment is the *top of rack* implementation. Here, a pair of redundant FCoE switches is installed at the top of each rack of servers. Both FC and IP connectivity to each server is accomplished using inexpensive Twinax cabling from the server to the top of rack FCoE switches. This short distance is well supported with Twinax. Connectivity from the top of rack switches to existing backbone LAN and SAN infrastructures, that is connections across racks, is typically done with optical links, which can support the longer cable runs that may be required.

FCoE Switches

An FCoE switch has both Ethernet switch and Fibre Channel switch functionalities. The FCoE switch has a Fibre Channel Forwarder (FCF), Ethernet Bridge, and set of Ethernet ports and optional FC ports, as shown in Figure 6-15. The function of the FCF is to encapsulate the FC frames, received from the FC port, into the FCoE frames and also to de-encapsulate the FCoE frames, received from the Ethernet Bridge, to the FC frames.

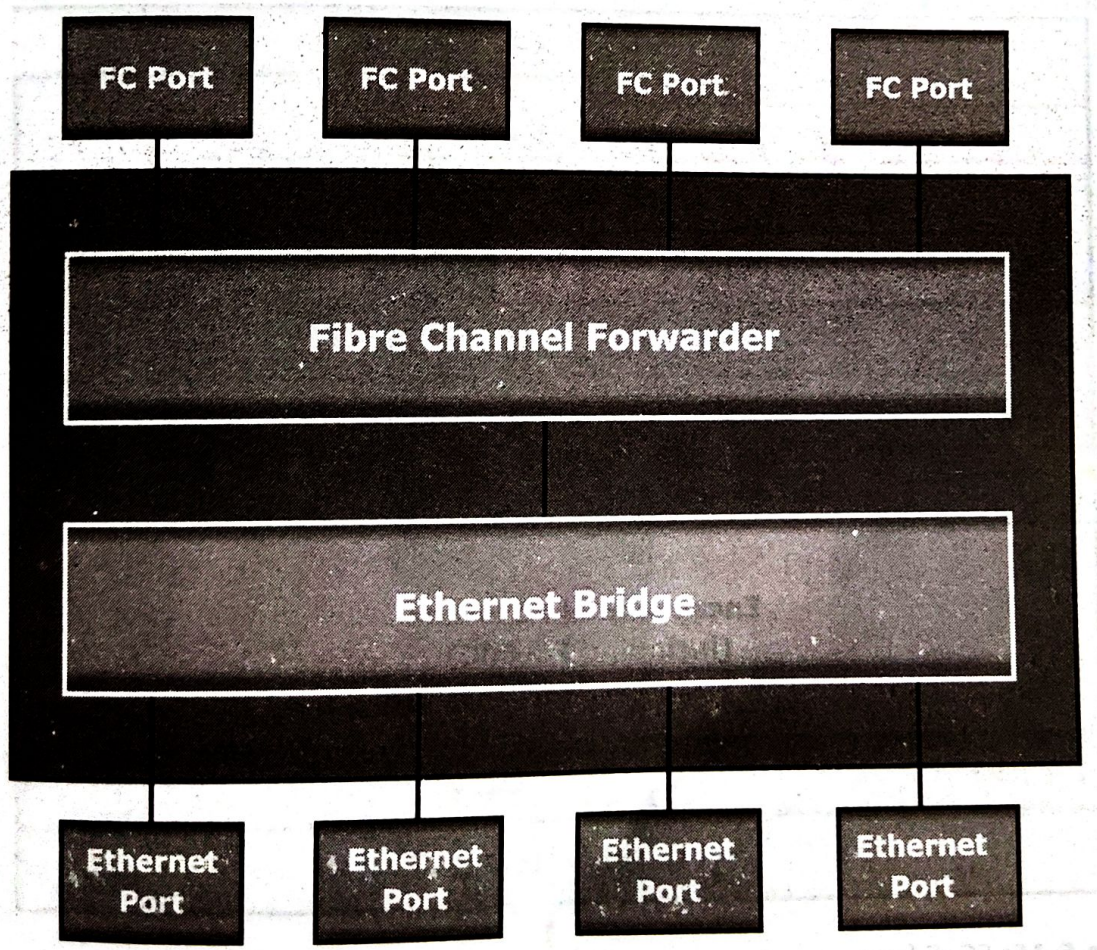


Figure 6-15: FCoE switch generic architecture

Upon receiving the incoming traffic, the FCoE switch inspects the Ethertype (used to indicate which protocol is encapsulated in the payload of an Ethernet frame) of the incoming frames and uses that to determine the destination. If the Ethertype of the frame is FCoE, the switch recognizes that the frame contains an FC payload and forwards it to the FCF. From there, the FC is extracted from the FCoE frame and transmitted to FC SAN over the FC ports. If the Ethertype is not FCoE, the switch handles the traffic as usual Ethernet traffic and forwards it over the Ethernet ports.

6.3.3 FCoE Frame Structure

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

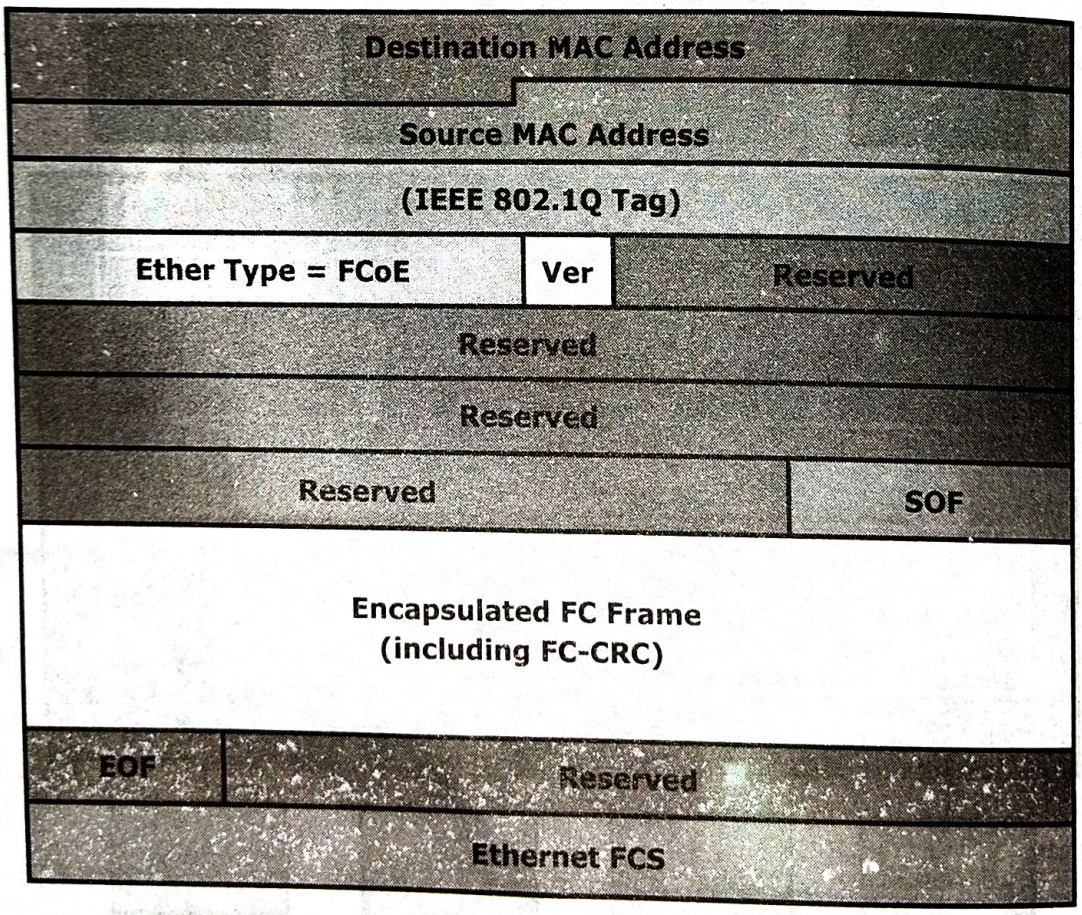


Figure 6-16: FCoE frame structure

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. The next chapter discusses jumbo frames in detail. FCoE requires Converged Enhanced Ethernet, which provides lossless Ethernet and jumbo frame support.

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.

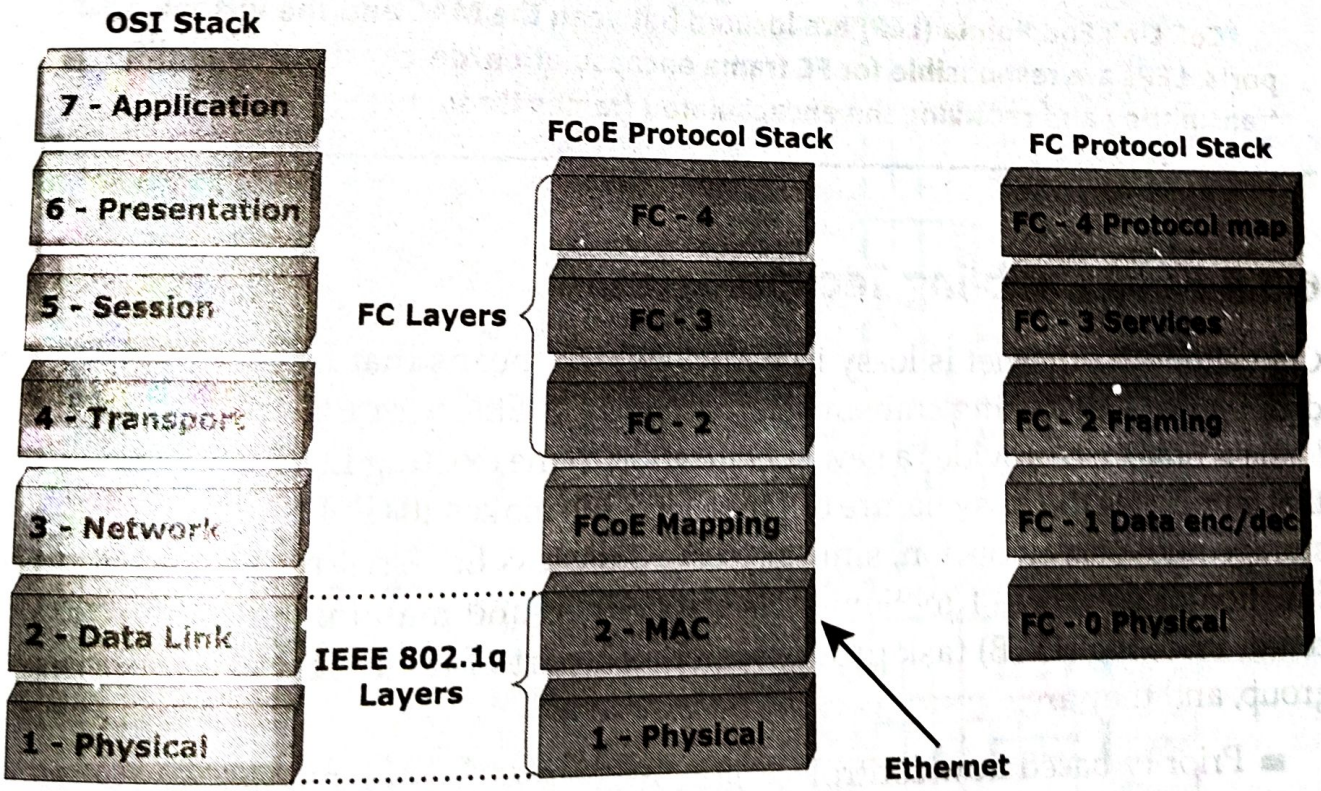


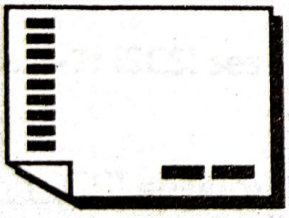
Figure 6-17: FCoE frame mapping

11. describe fcip protocol stack & topology. iscsi

6.2 FCIP

FC SAN provides a high-performance infrastructure for localized data movement. Organizations are now looking for ways to transport data over a long distance between their disparate SANs at multiple geographic locations. One of the best ways to achieve this goal is to interconnect geographically dispersed SANs through reliable, high-speed links. This approach involves transporting the FC block data over the IP infrastructure. FCIP is a tunneling protocol that enables distributed FC SAN islands to be interconnected over the existing IP-based networks.

The FCIP standard has rapidly gained acceptance as a manageable, cost-effective way to blend the best of the two worlds: FC SAN and the proven, widely deployed IP infrastructure. As a result, organizations now have a better way to store, protect and move their data by leveraging investments in their existing IP infrastructure. FCIP is extensively used in disaster recovery implementations in which data is duplicated to the storage located at a remote site.



FCIP might require high network bandwidth when replicating or backing up data. FCIP does not handle data traffic throttling or flow control; these are controlled by the communicating FC switches and devices within the fabric.

6.2.1 FCIP Protocol Stack

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

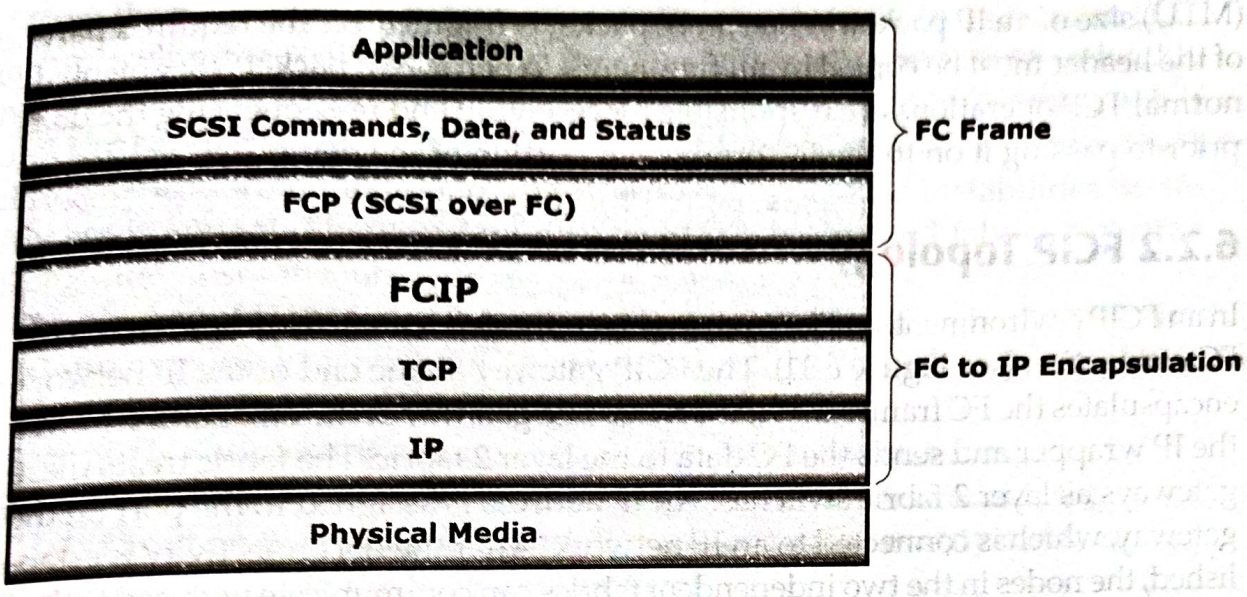


Figure 6-9: FCIP protocol stack

The upper layer protocol SCSI includes the SCSI driver program that executes the read-and-write commands. Below the SCSI layer is the Fibre Channel Protocol (FCP) layer, which is simply a Fibre Channel frame whose payload is SCSI. The FCP layer rides on top of the Fibre Channel transport layer. This enables the FC frames to run natively within a SAN fabric environment. In addition, the FC frames can be encapsulated into the IP packet and sent to a remote SAN over the IP. The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Figure 6-10). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IP traffic.

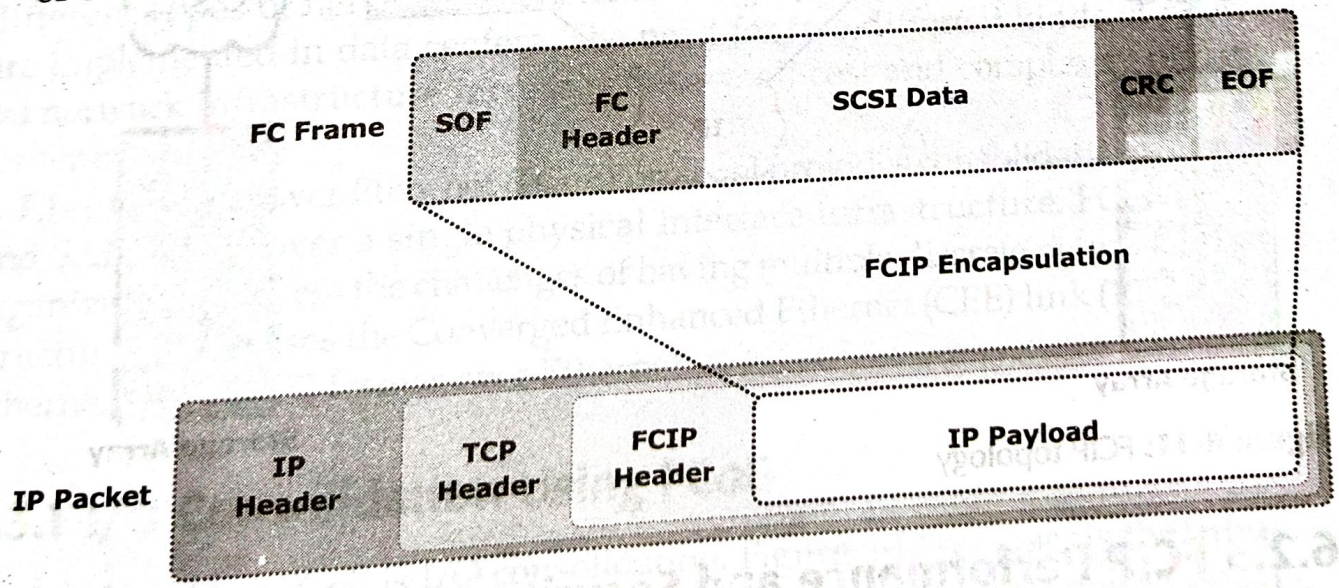


Figure 6-10: FCIP encapsulation

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.

6.2.2 FCIP Topology

In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (see Figure 6-11). The FCIP gateway at one end of the IP network encapsulates the FC frames into IP packets. The gateway at the other end removes the IP wrapper and sends the FC data to the layer 2 fabric. 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.

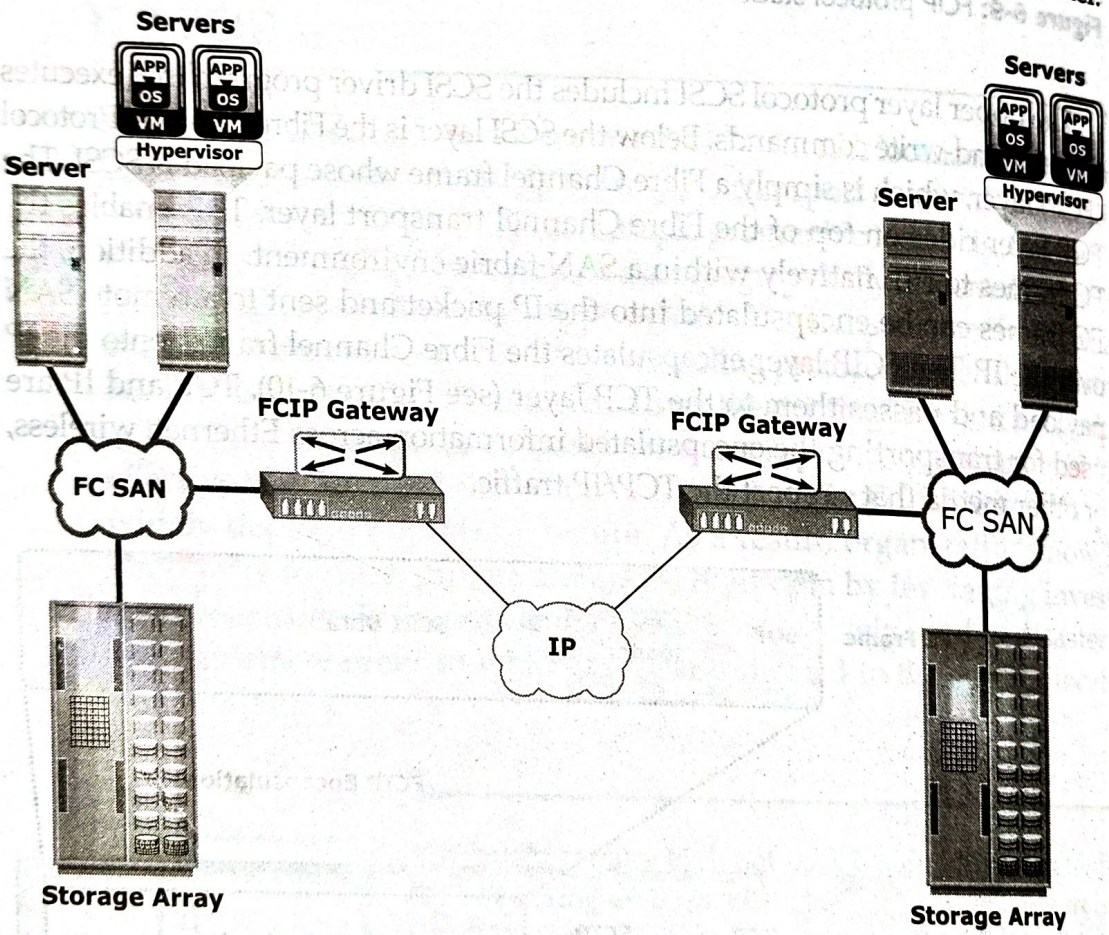


Figure 6-11: FCIP topology

6.2.3 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.

12. Explain the establishing of iscsi session, iscsi command sequencing.

6.1.8 iSCSI Session

An iSCSI session is established between an initiator and a target, as shown in Figure 6-7. A session is identified by a session ID (SSID), which includes part of an initiator ID and a target ID. The session can be intended for one of the following:

- The discovery of the available targets by the initiators and the location of a specific target on a network
- The normal operation of iSCSI (transferring data between initiators and targets)

There might be one or more TCP connections within each session. Each TCP connection within the session has a unique connection ID (CID).

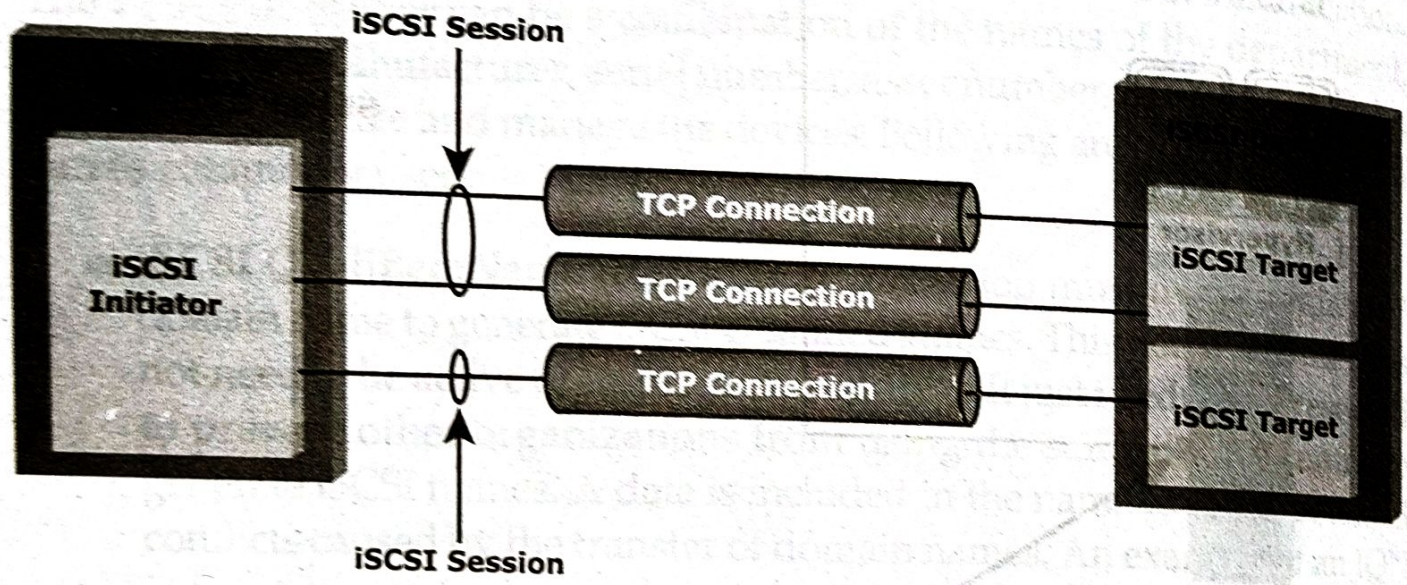


Figure 6-7: iSCSI session

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.

6.1.9 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.

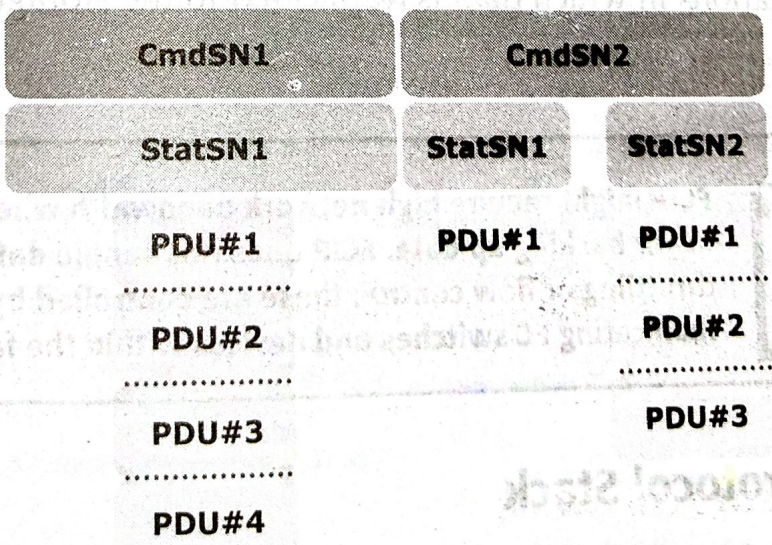


Figure 6-8: Command and status sequence number

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.

13. write a short notes of iscsi topology, iscsi protocol stack, iscsi pdu.

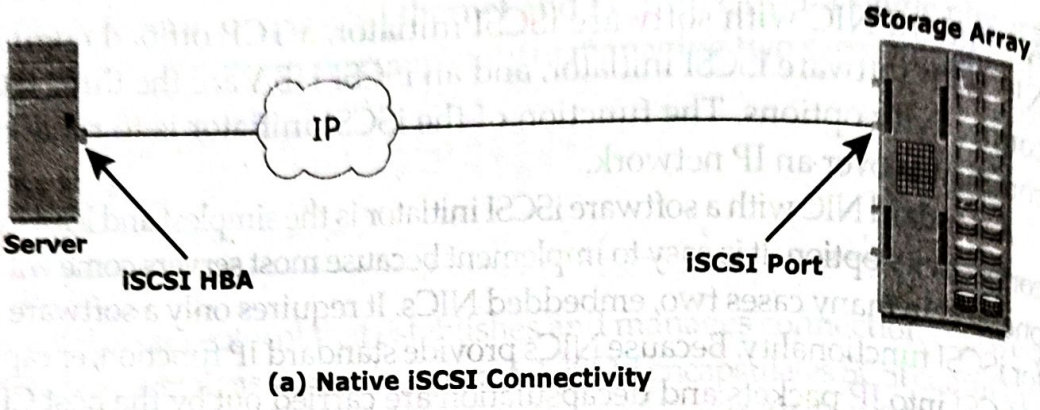
6.1.3 iSCSI Topologies

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

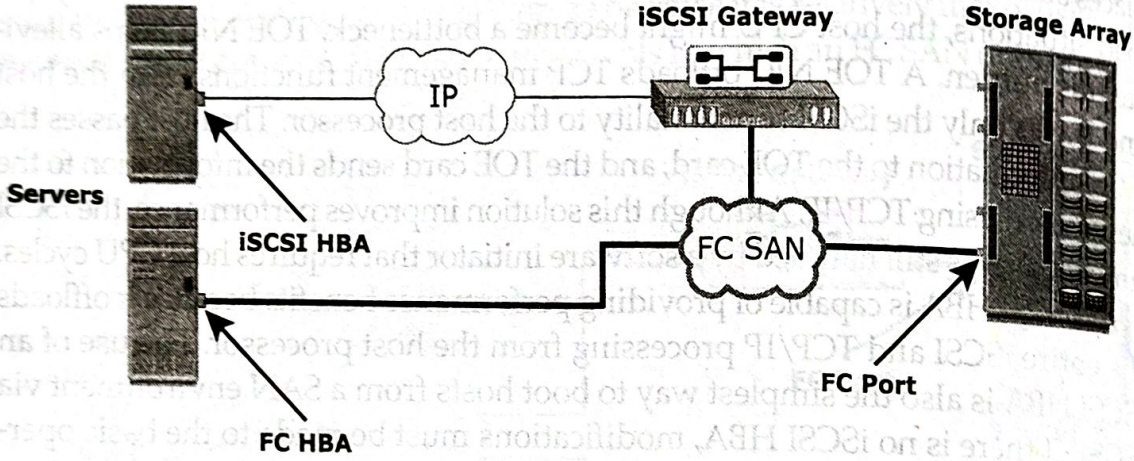
Native iSCSI Connectivity

FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed. In Figure 6-2 (a), the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernet switch.

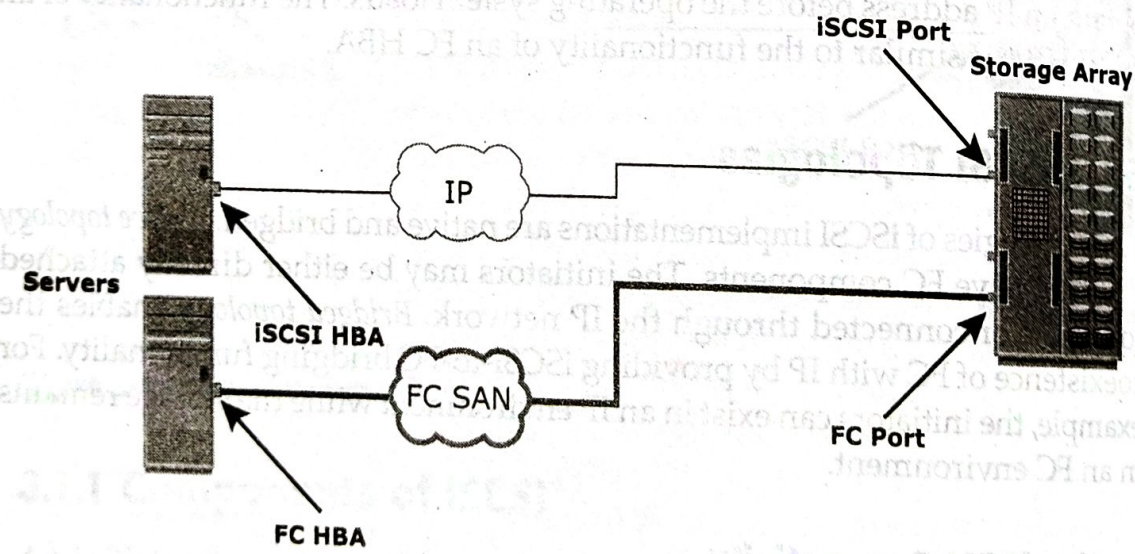
After an initiator is logged on to the network, it can access the available LUNs on the storage array. A single array port can service multiple hosts or initiators as long as the array port can handle the amount of storage traffic that the hosts generate.



(a) Native iSCSI Connectivity



(b) Bridged iSCSI Connectivity



(c) Combining FC and Native iSCSI Connectivity

Figure 6-2: iSCSI Topologies

Bridged iSCSI Connectivity

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

In this case, the array 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 Figure 6-2 (c).

6.1.4 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. 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.

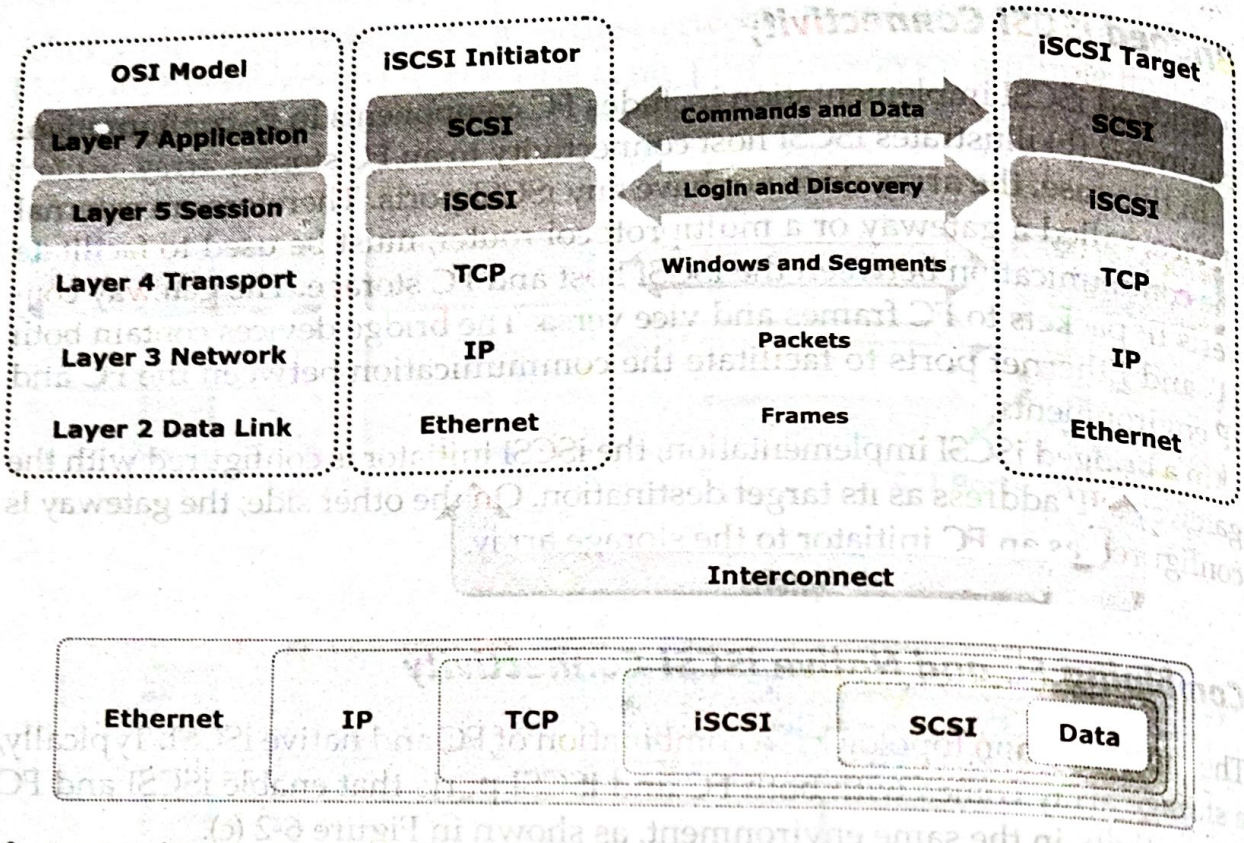


Figure 6-3: iSCSI protocol stack

6.1.5 iSCSI PDU

A *protocol data unit* (PDU) is the basic "information unit" in the iSCSI environment. The iSCSI initiators and targets communicate with each other using iSCSI PDUs. This communication includes establishing iSCSI connections and iSCSI sessions, performing iSCSI discovery, sending SCSI commands and data, and receiving SCSI status. All iSCSI PDUs contain one or more header segments followed by zero or more data segments. The PDU is then encapsulated into an IP packet to facilitate the transport.

A PDU includes the components shown in Figure 6-4. The IP header provides packet-routing information to move the packet across a network. The TCP header contains the information required to guarantee the packet delivery to the target. The iSCSI header (basic header segment) describes how to extract SCSI commands and data for the target. iSCSI adds an optional CRC, known as the *digest*, to ensure datagram integrity. This is in addition to TCP checksum and Ethernet CRC. The header and the data digests are optionally used in the PDU to validate integrity and data placement.

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

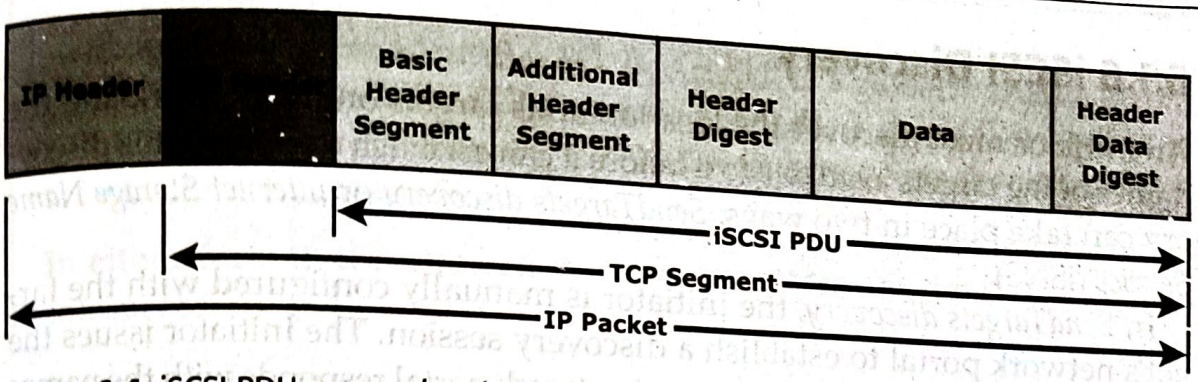


Figure 6-4: iSCSI PDU encapsulated in an IP packet

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

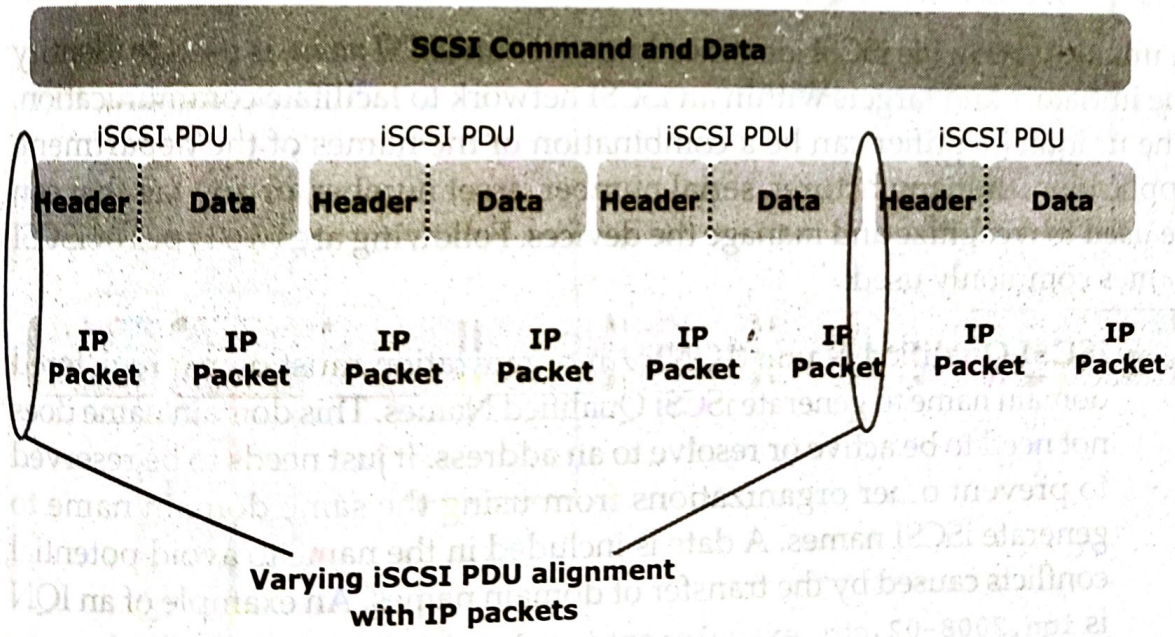


Figure 6-5: Alignment of iSCSI PDUs with IP packets

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.