MODULE-2

Data Protection: RAID

- In 1987, Patterson, Gibson, and Katz at the University of California, Berkeley, published a paper titled "A Case for **Redundant Arrays of Inexpensive Disks (RAID)**."
- RAID is the use of small-capacity, inexpensive disk drives as an alternative to large-capacity drives common on mainframe computers.
- Later RAID has been redefined to refer to ***independent*** disks to reflect advances in the storage technology.

- RAID stands for Redundant Array of Independent Disk.
- RAID is the way of combining several independent small disks into a single large-size storage.
- It appears to the OS as a single large-size disk.
- It is used to increase performance and availability of data-storage.
- There are two types of RAID implementation 1) hardware and 2) software.
- RAID-controller is a specialized hardware which

→ performs all RAID-calculations and

→ presents disk-volumes to host.

- Key functions of RAID-controllers:

1) Management and control of disk-aggregations.
2) Translation of I/O-requests between logical-disks and physical-disks.
3) Data-regeneration in case of disk-failures.

1.1 RAID Implementation Methods

- The two methods of RAID implementation are:
    1. Hardware RAID.
    2. Software RAID.

1.10.1 Hardware RAID

- In hardware RAID implementations, a specialized hardware controller is implemented either on the *host* or on the *array*.
- **Controller card RAID** is a *host-based hardware RAID* implementation in which a specialized RAID controller is installed in the host, and disk drives are connected to it.
- Manufacturers also integrate RAID controllers on motherboards.
- A host-based RAID controller is not an efficient solution in a data center environment with a large number of hosts.
- The external RAID controller is an *array-based hardware RAID*.
- It acts as an interface between the host and disks.
- It presents storage volumes to the host, and the host manages these volumes as physical drives.
- The key functions of the RAID controllers are as follows:
    - ✓ Management and control of disk aggregations
    - ✓ Translation of I/O requests between logical disks and physical disks
    - ✓ Data regeneration in the event of disk failures

1.10.2 Software RAID

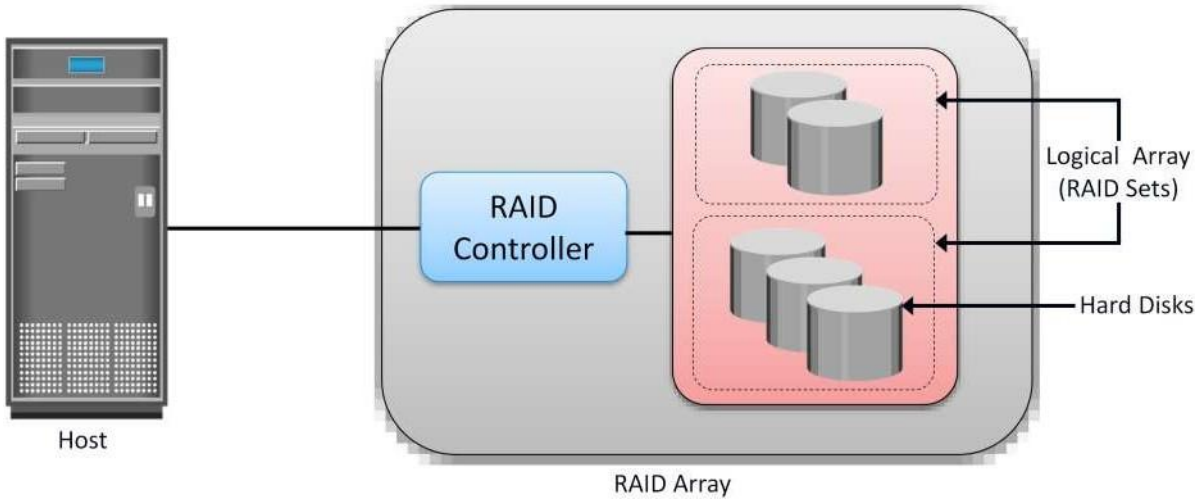
- **Software RAID** uses host-based software to provide RAID functions.
- It is implemented at the operating-system level and does not use a dedicated hardware controller to manage the RAID array.
- Advantages when compared to Hardware RAID:
    - ✓ cost
    - ✓ simplicity benefits

- Limitations:
  - ✓ **Performance**: Software RAID affects overall system performance. This is due to additional CPU cycles required to perform RAID calculations.
  - ✓ **Supported features**: Software RAID does not support all RAID levels.
  - ✓ **Operating system compatibility**: Software RAID is tied to the host operating system; hence, upgrades to software RAID or to the operating system should be validated for compatibility. This leads to inflexibility in the data-processing environment.

1.11 RAID Array Components

- A RAID array is an enclosure that contains a number of disk drives and supporting Hardware to implement RAID.
- A subset of disks within a RAID array can be grouped to form logical associations called logical arrays, also known as a RAID set or a RAID group

Fig: Components of RAID array



1.2 RAID Techniques

- There are three RAID techniques
    1. striping
    2. mirroring
    3. parity

1.11.1 Striping

- **Striping** is a technique to spread data across multiple drives (more than one) to use the drives in parallel.
- All the read-write heads work simultaneously, allowing more data to be processed in a shorter time and increasing performance, compared to reading and writing from a single disk.
- Within each disk in a RAID set, a **predefined number of contiguously addressable** disk blocks are defined as a **strip**.
- The set of aligned strips that spans across all the disks within the RAID set is called a **stripe**.
- Fig 1.11 shows physical and logical representations of a striped RAID set.

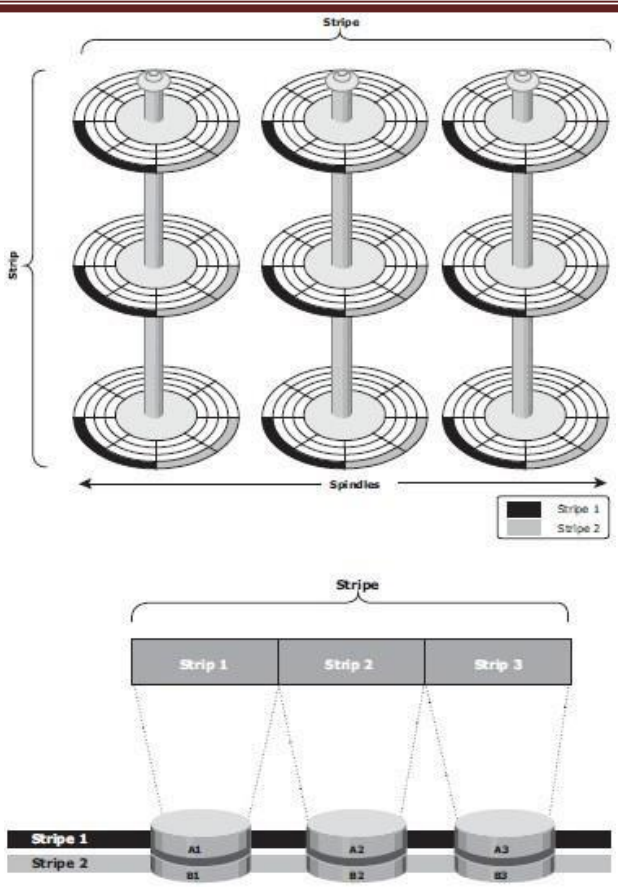


Fig 1.11: Striped RAID set

- **Strip size** (also called **stripe depth**) describes the number of blocks in a strip and is the maximum amount of data that can be written to or read from a single disk in the set.
- All strips in a stripe have the same number of blocks.
  - ✓ Having a smaller strip size means that data is broken into smaller pieces while spread across the disks.
- **Stripe size** is a multiple of strip size by the number of **data** disks in the RAID set.
  - ✓ Eg: In a 5 disk striped RAID set with a strip size of 64 KB, the stripe size is 320KB (64KB x 5).
- **Stripe width** refers to the number of *data* strips in a stripe.
- Striped RAID does not provide any data protection unless parity or mirroring is used.

1.11.2 Mirroring

- **Mirroring** is a technique whereby the same data is stored on two different disk drives, yielding two copies of the data.
- If one disk drive failure occurs, the data is intact on the surviving disk drive (see Fig 1.12) and the controller continues to service the host's data requests from the surviving disk of a mirrored pair.
- When the failed disk is replaced with a new disk, the controller copies the data from the surviving disk of the mirrored pair.
- This activity is transparent to the host.

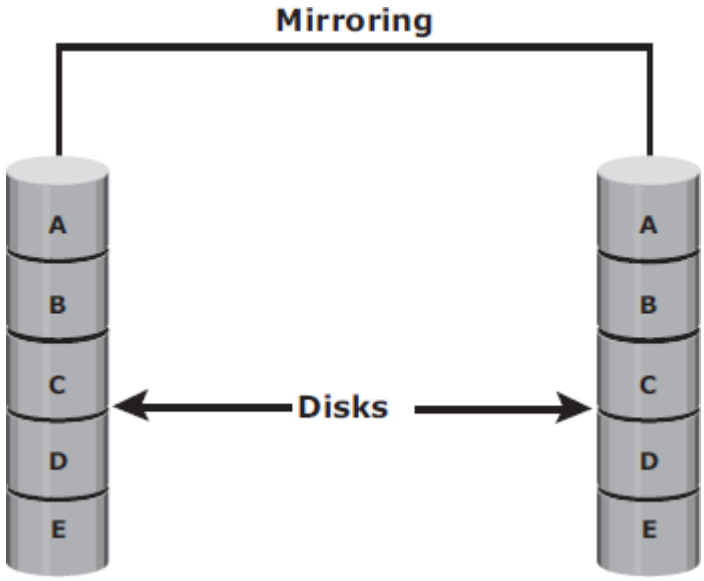


Fig 1.12: Mirrored disks in an array

- Advantages:
  - ✓ complete data redundancy,
  - ✓ mirroring enables fast recovery from disk failure.
  - ✓ data protection
- Mirroring is not a substitute for data backup. Mirroring constantly captures changes in the data, whereas a backup captures point-in-time images of the data.
- Disadvantages:
  - ✓ Mirroring involves duplication of data — the amount of storage capacity needed is

twice the amount of data being stored.

- ✓ Expensive

1.11.3 Parity

- **Parity** is a method to protect striped data from disk drive failure without the cost of mirroring.
- *An additional disk drive is added to hold parity*, a mathematical construct that allows re-creation of the missing data.
- Parity is a **redundancy technique** that ensures protection of data without maintaining a full set of duplicate data.
- Calculation of parity is a function of the RAID controller.
- Parity information can be stored on separate, dedicated disk drives or distributed across all the drives in a RAID set.
- Fig 1.13 shows a parity RAID set.

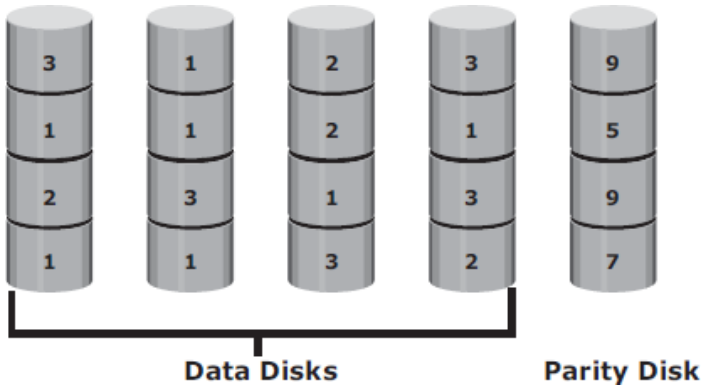


Fig 1.13: Parity RAID

- The first four disks, labeled "*Data Disks*," contain the data. The fifth disk, labeled "*Parity Disk*," stores the parity information, which, in this case, is the sum of the elements in each row.
- Now, if one of the data disks fails, the missing value can be calculated by subtracting the sum of the rest of the elements from the parity value.
- Here, computation of parity is represented as an arithmetic sum of the data. However, parity calculation is a bitwise XOR operation.

XOR Operation:

- A bit-by-bit Exclusive -OR (XOR) operation takes two bit patterns of equal length and performs the logical XOR operation on each pair of corresponding bits.
- The result in each position is 1 if the two bits are different, and 0 if they are the same.

- The truth table of the XOR operation is shown below (A and B denote inputs and C, the output the XOR operation).

Table 1.1: Truth table for XOR Operation

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

- If any of the data from A, B, or C is lost, it can be reproduced by performing an XOR operation on the remaining available data.
- Eg: if a disk containing all the data from A fails, the data can be regenerated by performing an XOR between B and C.
- Advantages:
  - ✓ Compared to mirroring, parity implementation considerably reduces the **cost** associated with data protection.
- Disadvantages:
  - ✓ Parity information is generated from data on the data disk. Therefore, parity is recalculated every time there is a change in data.
  - ✓ This recalculation is time-consuming and affects the performance of the RAID array.
- For parity RAID, the stripe size calculation does not include the parity strip.
- Eg: in a five (4 + 1) disk parity RAID set with a strip size of 64 KB, the stripe size will be 256 KB (64 KB x 4).

1.3 RAID Levels

- RAID Level selection is determined by below factors:
  - ✓ Application performance
  - ✓ data availability requirements
  - ✓ cost
- RAID Levels are defined on the basis of:
  - ✓ Striping
  - ✓ Mirroring
  - ✓ Parity techniques
- Some RAID levels use a single technique whereas others use a combination of techniques.
- Table 1.2 shows the commonly used RAID levels

Table 1.2: RAID Levels

LEVELS	BRIEF DESCRIPTION
RAID 0	Striped set with no fault tolerance
RAID 1	Disk mirroring
Nested	Combinations of RAID levels. Example: RAID 1 + RAID 0
RAID 3	Striped set with parallel access and a dedicated parity disk
RAID 4	Striped set with independent disk access and a dedicated parity disk
RAID 5	Striped set with independent disk access and distributed parity
RAID 6	Striped set with independent disk access and dual distributed parity

1.12.1 RAID 0

- **RAID 0** configuration uses *data striping techniques*, where data is striped across all the disks within a RAID set. Therefore it utilizes the full storage capacity of a RAID set.
- To read data, all the strips are put back together by the controller.
- Fig 1.14 shows RAID 0 in an array in which data is striped across five disks.

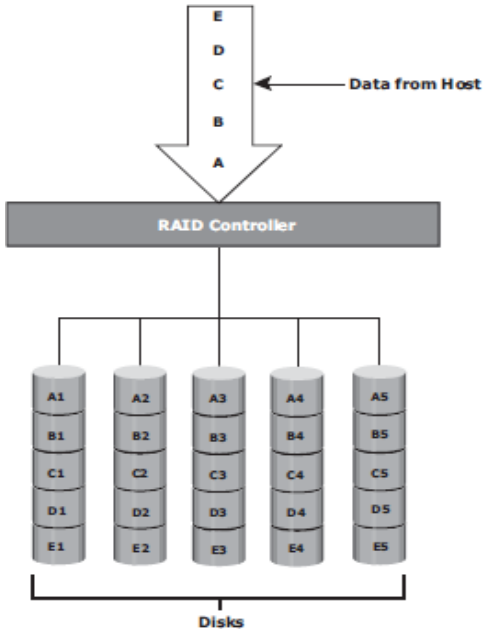


Fig 1.14: RAID 0

- When the number of drives in the RAID set increases, performance improves because more data can be read or written simultaneously.
- RAID 0 is a good option for applications that need high I/O throughput.
- However, if these applications require high availability during drive failures, RAID 0 does not provide data protection and availability.

1.12.2 RAID 1

- **RAID 1** is based on the *mirroring* technique.
- In this RAID configuration, data is mirrored to provide *fault tolerance* (see Fig 1.15). A
- RAID 1 set consists of two disk drives and every write is written to both disks.
- The mirroring is transparent to the host.
- During disk failure, the impact on data recovery in RAID 1 is the least among all RAID implementations. This is because the RAID controller uses the mirror drive for data recovery.
- RAID 1 is suitable for applications that require high availability and cost is no constraint.

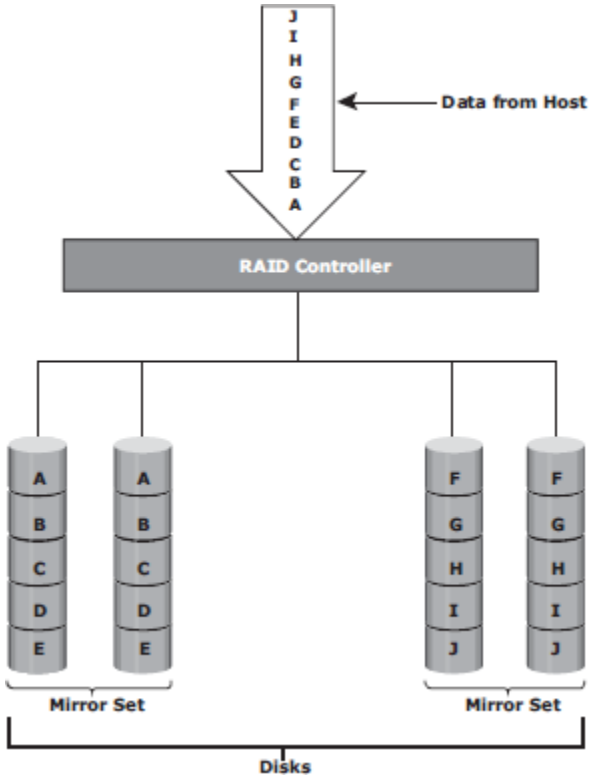


Fig 1.15: RAID 1

1.12.3 Nested RAID

- Most data centers require data redundancy and performance from their RAID arrays.
- RAID 1+0 and RAID 0+1 combine the performance benefits of RAID 0 with the redundancy benefits of RAID 1.
- They use striping and mirroring techniques and combine their benefits.
- These types of RAID require an even number of disks, the minimum being four (see Fig 1.16).

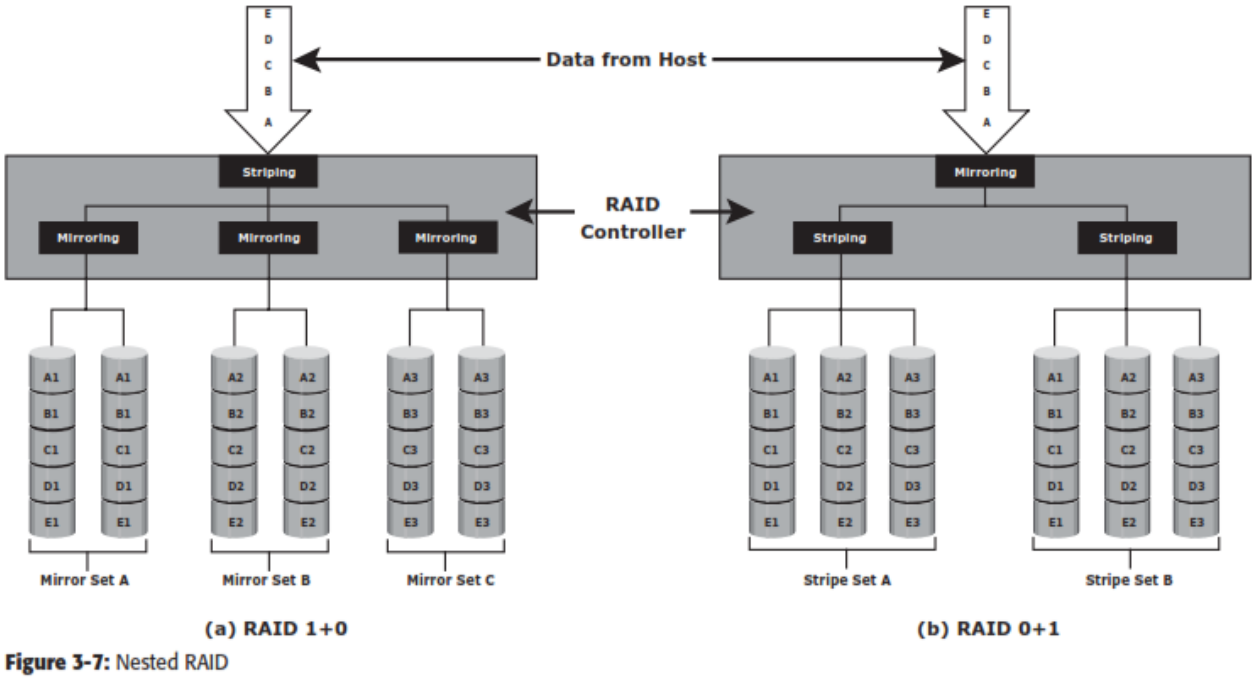


Figure 3-7: Nested RAID

Fig 1.16: Nested RAID

RAID 1+0:

- RAID 1+0 is also known as RAID 10 (Ten) or RAID 1/0.
- RAID 1+0 performs well for workloads with small, random, write-intensive I/Os.
- Some applications that benefit from RAID 1+0 include the following:
  - ✓ High transaction rate Online Transaction Processing (OLTP)
  - ✓ Large messaging installations
  - ✓ Database applications with write intensive random access workloads
- **RAID 1+0** is also called striped mirror.
- The basic element of RAID 1+0 is a mirrored pair, which means that data is first mirrored and then both copies of the data are striped across multiple disk drive pairs in a RAID set.
- When replacing a failed drive, only the mirror is rebuilt. The disk array controller uses the surviving drive in the mirrored pair for data recovery and continuous operation.

Working of RAID 1+0:

- Eg: consider an example of six disks forming a RAID 1+0 (RAID 1 first and then RAID 0) set.
- These six disks are paired into three sets of two disks, where each set acts as a RAID 1 set (mirrored pair of disks). Data is then striped across all the three mirrored sets to form RAID 0.

- Following are the steps performed in RAID 1+0 (see Fig 1.16 [a]):
  - ✓ Drives 1+2 = RAID 1 (Mirror Set A)
  - ✓ Drives 3+4 = RAID 1 (Mirror Set B)
  - ✓ Drives 5+6 = RAID 1 (Mirror Set C)
- Now, RAID 0 striping is performed across sets A through C.
- In this configuration, if drive 5 fails, then the mirror set C alone is affected. It still has drive 6 and continues to function and the entire RAID 1+0 array also keeps functioning.
- Now, suppose drive 3 fails while drive 5 was being replaced. In this case the array still continues to function because drive 3 is in a different mirror set.
- So, in this configuration, up to three drives can fail without affecting the array, as long as they are all in different mirror sets.
- **RAID 0+1** is also called a mirrored stripe.
- The basic element of RAID 0+1 is a stripe. This means that the process of striping data across disk drives is performed initially, and then the entire stripe is mirrored.
- In this configuration if one drive fails, then the entire stripe is faulted.

Working of RAID 0+1:

- Eg: Consider the same example of six disks forming a RAID 0+1 (that is, RAID 0 first and then RAID 1).
- Here, six disks are paired into two sets of three disks each.
- Each of these sets, in turn, act as a RAID 0 set that contains three disks and then these two sets are mirrored to form RAID 1.
- Following are the steps performed in RAID 0+1 (see Fig 1.16 [b]):
  - ✓ Drives 1 + 2 + 3 = RAID 0 (Stripe Set A)
  - ✓ Drives 4 + 5 + 6 = RAID 0 (Stripe Set B)
- These two stripe sets are mirrored.
- If one of the drives, say drive 3, fails, the entire stripe set A fails.
- A rebuild operation copies the entire stripe, copying the data from each disk in the healthy stripe to an equivalent disk in the failed stripe.
- This causes increased and unnecessary I/O load on the surviving disks and makes the RAID set more vulnerable to a second disk failure.

1.12.4 RAID 3

- RAID 3 stripes data for high performance and uses parity for improved fault tolerance.
- Parity information is stored on a dedicated drive so that data can be reconstructed if a drive fails. For example, of five disks, four are used for data and one is used for parity.
- RAID 3 always reads and writes complete stripes of data across all disks, as the drives operate in parallel. There are no partial writes that update one out of many strips in a stripe.
- RAID 3 provides good bandwidth for the transfer of large volumes of data. RAID 3 is used in applications that involve large sequential data access, such as video streaming.
- Fig 1.17 shows the RAID 3 implementation

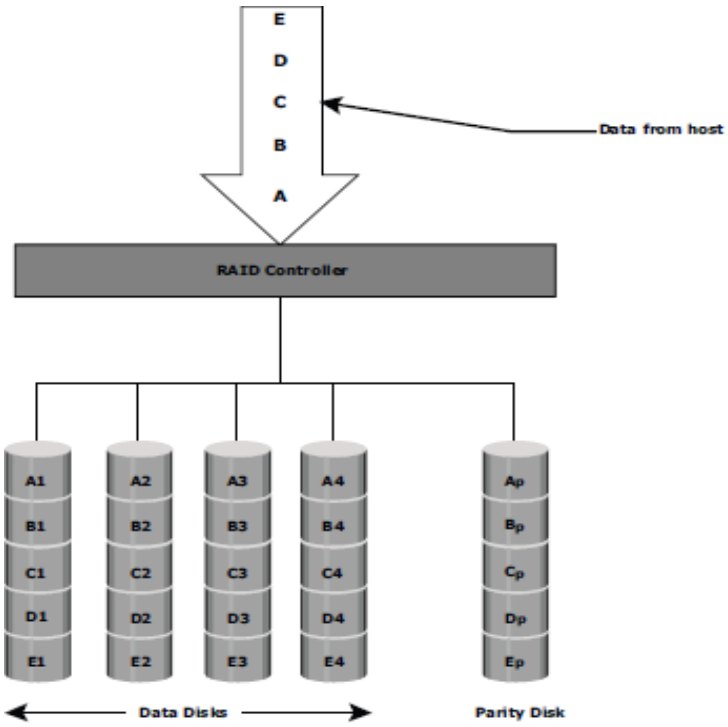


Fig 1.17: RAID 3

1.12.5 RAID 4

- RAID 4 stripes data for high performance and uses parity for improved fault tolerance. Data is striped across all disks except the parity disk in the array.
- Parity information is stored on a dedicated disk so that the data can be rebuilt if a drive fails. Striping is done at the block level.

- Unlike RAID 3, data disks in RAID 4 can be accessed independently so that specific data elements can be read or written on single disk without read or write of an entire stripe. RAID 4 provides good read throughput and reasonable write throughput.

1.12.6 RAID 5

- RAID 5 is a versatile RAID implementation.
- It is similar to RAID 4 because it uses striping. The drives (strips) are also independently accessible.
- The difference between RAID 4 and RAID 5 is the parity location. In RAID 4, parity is written to a dedicated drive, creating a write bottleneck for the parity disk
- In RAID 5, parity is distributed across all disks. The distribution of parity in RAID 5 overcomes the Write bottleneck. Below Figure illustrates the RAID 5 implementation.
- Fig 1.18 illustrates the RAID 5 implementation.
- RAID 5 is good for random, read-intensive I/O applications and preferred for messaging, data mining, medium-performance media serving, and relational database management system (RDBMS) implementations, in which database administrators (DBAs) optimize data access.

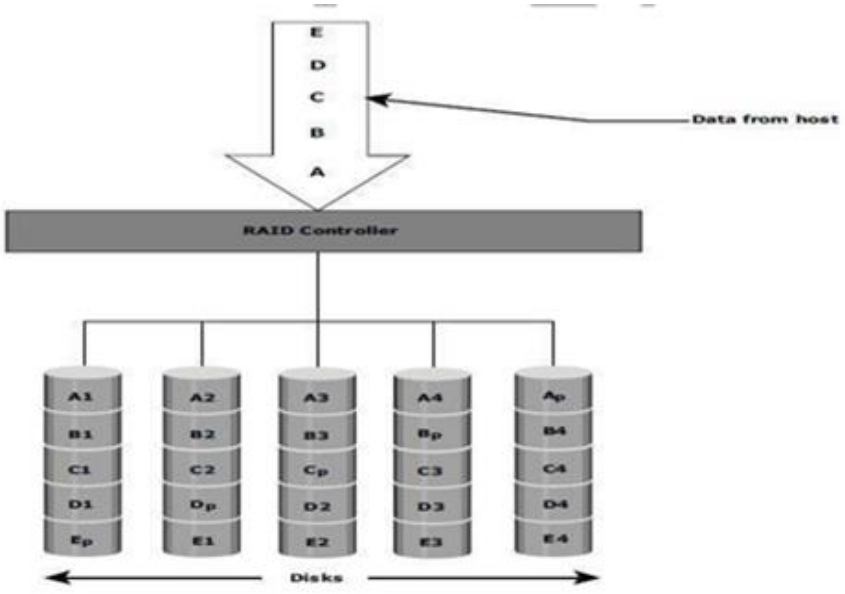


Fig 1.18: RAID 5

1.12.7 RAID 6

- RAID 6 includes a second parity element to enable survival in the event of the failure of two disks in a RAID group. Therefore, a RAID 6 implementation requires at least four disks.
- RAID 6 distributes the parity across all the disks. The write penalty in RAID 6 is more than that in RAID 5; therefore, RAID 5 writes perform better than RAID 6. The rebuild operation in RAID 6 may take longer than that in RAID 5 due to the presence of two parity sets.
- Fig 1.19 illustrates the RAID 6 implementation

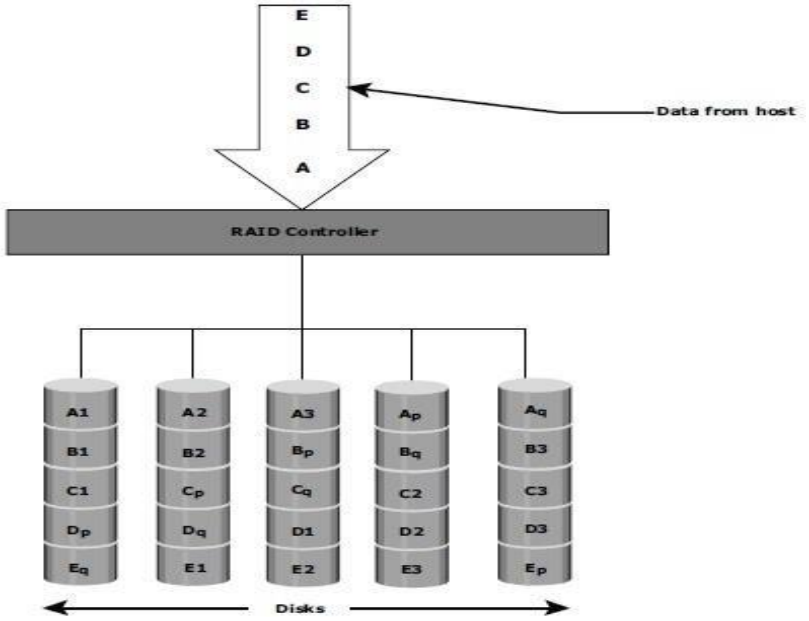


Fig 1.19: RAID 6

1.4 RAID Impact on Disk Performance

- When choosing a RAID type, it is imperative to consider its impact on disk performance and application IOPS.
- In both mirrored (RAID 1) and parity RAID (RAID 5) configurations, every write operation translates into more I/O overhead for the disks which is referred to as **write penalty**.
- In a RAID 1 implementation, every write operation must be performed on two disks configured as a mirrored pair. **The write penalty is 2.**
- In a RAID 5 implementation, a write operation may manifest as four I/O operations. When performing small I/Os to a disk configured with RAID 5, the controller has to read, calculate, and write a parity segment for every data write operation.
- Fig 1.20 illustrates a single write operation on RAID 5 that contains a group of five disks.

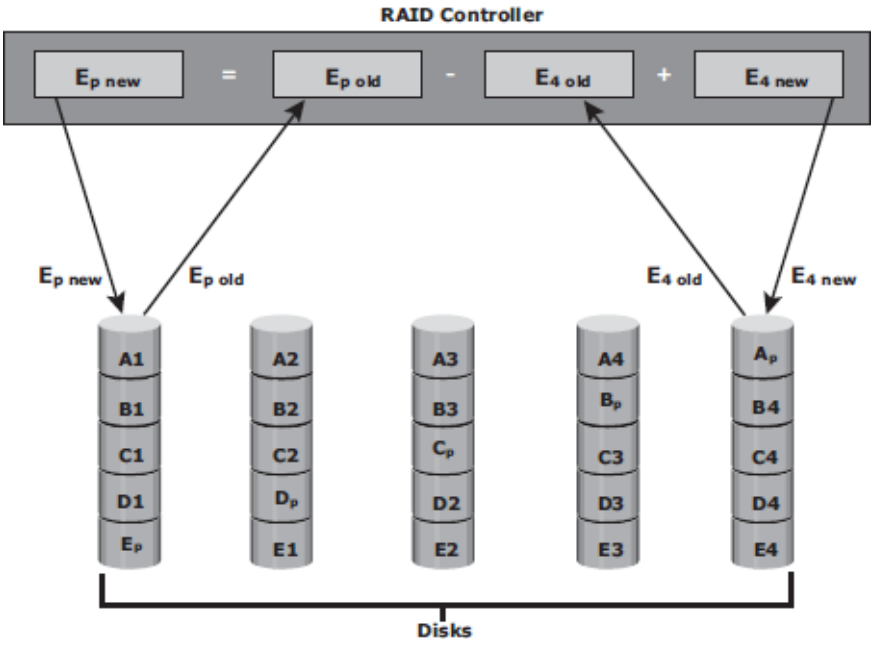


Fig 1.20: Write Penalty in RAID 5

- Four of these disks are used for data and one is used for parity.
- The **parity (E_p)** at the controller is calculated as follows:

$$E_p = E_1 + E_2 + E_3 + E_4 \text{ (XOR operations)}$$

- Whenever the controller performs a write I/O, parity must be computed by reading the old parity (E_p old) and the old data (E_4 old) from the disk, which means two read I/Os.
- The new parity (E_p new) is computed as follows:

$$E_p \text{ new} = E_p \text{ old} - E_4 \text{ old} + E_4 \text{ new (XOR operations)}$$

- After computing the new parity, the controller completes the write I/O by doing two write I/Os for the new data and the new parity onto the disks..
- Therefore, the controller performs two disk reads and two disk writes for every write operation, and **the write penalty is 4**.
- In RAID 6, which maintains dual parity, a disk write requires **three read operations**: two parity and one data.
- After calculating both new parities, the controller performs **three write operations**: two parity and an I/O.
- Therefore, in a RAID 6 implementation, the controller performs six I/O operations for each write I/O, and the **write penalty is 6**.

3.1.1 Application IOPS and RAID Configurations

When deciding the number of disks required for an application, it is important to consider the impact of RAID based on IOPS generated by the application. The total disk load should be computed by considering the type of RAID configuration and the ratio of read compared to write from the host. The following example illustrates the method of computing the disk load in different types of RAID.

i. Consider an application that generates 5,200 IOPS, with 60 percent of them being reads. The disk load in RAID 5 is calculated as follows:

RAID 5 disk load = 0.6 × 5,200 + **4** × (0.4 × 5,200) [because the write penalty for RAID 5 is 4]
= 3,120 + 4 × 2,080 [1.0-0.6=0.4]
= 3,120 + 8,320
= 11,440 IOPS

The disk load in RAID 1 is calculated as follows:

RAID 1 disk load = 0.6 × 5,200 + **2** × (0.4 × 5,200)
[because every write manifests as two writes to the disks]
= 3,120 + 2 × 2,080
= 3,120 + 4,160
= 7,280 IOPS

ii. Computed disk load determines the number of disks required for the application. If in this example an HDD with a specification of a maximum 180 IOPS for the application needs to be used, the number of disks required to meet the workload for the RAID configuration as follows:

RAID 5: 11,440 / 180 = 64 disks
RAID 1: 7,280 / 180 = 42 disks (approximated to the nearest even number)

Calculating IOPS from disks available:

Consider a server/storage with 8 450GB 15,000 RPM drives. We will consider two scenarios of Workload 80% Write 20%Read and another scenario with 20% Write 80% Read. Also we will calculate IOPS that cab be achieved in RAID5 and RAID 10 Scenario.

Total Raw IOPS = Disk Speed IOPS * Number of disks
Functional IOPS =(((Total Raw IOPS × Write %))/(RAID Penalty))+(Total Raw IOPS×Read %)

In our example ,
Total Raw IOPS = 175*8 = 1400 IOPS (Since 15K RPM disk can give 175 IOPS)

RAID-5:

Scenario 1(80% Write 20%Read) Functional IOPS = (((1400*0.8))/(4))+(1400*0.2) = 560 IOPS **Scenario 2**(20% Write 80%Read) Functional IOPS = (((1400*0.2))/(4))+(1400*0.8) = 1190 IOPS

RAID-1:

Scenario 1(80%Write20%Read) Functional IOPS = (((1400*0.8))/(2))+(1400*0.2) = 840 IOPS **Scenario 2**(20%Write80%Read) Functional IOPS = (((1400*0.2))/(2))+(1400*0.8) = 1260 IOPS

Calculating number of Disks required to achieve certain IOPS:

Consider a scenario where you will have to decide on RAID and number of disks required

to achieve 2000 IOPS with a workload characterization of 80% Write 20%Read and another scenario with 20% Write 80% Read.

Total number of Disks required = ((Total Read IOPS + (Total Write IOPS*RAID Penalty))/Disk Speed IOPS)

Total IOPS = 2000

Note : 80% 0f 2000 IOPS = 1600 IOPS & 20% of 2000 IOPS = 400 IOPS

RAID-5:

Scenario 1(80% Write 20% Read) – Total Number of disks required = ((400+(1600*4))/175) = 39 Disks approximately

Scenario 2(20% Write 80%Read) – Total Number of disks required = ((1600+(400*4))/175) = 18 Disks approximately

RAID-1:

Scenario 1(80% Write 20% Read) – Total Number of disks required = ((400+(1600*2))/175) = 21 Disks approximately

Scenario 2(20% Write 80% Read) – Total Number of disks required = ((1600+(400*2))/175) = 14 Disks approximately

RAID Comparison

Table 3-2: Comparison of Common RAID Types

RAID	MIN. DISKS	STORAGE EFFICIENCY %	COST	READ PERFORMANCE	WRITE PERFORMANCE	WRITE PENALTY	PROTECTION
0	2	100	Low	Good for both random and sequential reads	Good	No	No protection
1	2	50	High	Better than single disk	Slower than single disk because every write must be committed to all disks	Moderate	Mirror protection
3	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Fair for random reads and good for sequential reads	Poor to fair for small random writes and fair for large, sequential writes	High	Parity protection for single disk failure
4	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
5	3	[(n-1)/n] × 100 where n= number of disks	Moderate	Good for random and sequential reads	Fair for random and sequential writes	High	Parity protection for single disk failure
6	4	[(n-2)/n] × 100 where n= number of disks	Moderate but more than RAID 5.	Good for random and sequential reads	Poor to fair for random writes and fair for sequential writes	Very High	Parity protection for two disk failures
1+0 and 0+1	4	50	High	Good	Good	Moderate	Mirror protection

1.5 Components of an Intelligent Storage System

- Intelligent Storage Systems are **feature-rich RAID arrays** that provide highly optimized I/O processing capabilities.
- These storage systems are configured with a large amount of memory (called *cache*) and multiple I/O paths and use sophisticated algorithms to meet the requirements of performance-sensitive applications.
- An intelligent storage system consists of **four key components** (Refer Fig 1.21):
  - ✓ Front End
  - ✓ Cache
  - ✓ Back end
  - ✓ Physical disks.
- An I/O request received from the host at the front-end port is processed through cache and the back end, to enable storage and retrieval of data from the physical disk.
- A read request can be serviced directly from cache if the requested data is found in cache.
- In modern intelligent storage systems, front end, cache, and back end are typically integrated on a single board (referred to as a storage processor or storage controller).

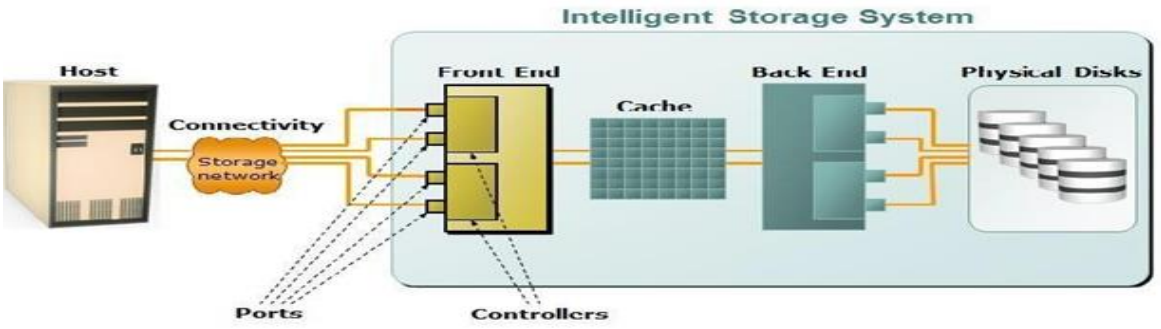


Fig 1.21 Components of an Intelligent Storage System

1.14.1 Front End

- The front end provides the interface between the storage system and the host.
- It consists of two components:
  i. Front-End Ports
  ii. Front-End Controllers.

- A front end has redundant controllers for high availability, and each controller contains multiple **front-end ports** that enable large numbers of hosts to connect to the intelligent storage system.
- Each front-end controller has processing logic that executes the appropriate transport protocol, such as Fibre Channel, iSCSI, FICON, or FCoE for storage connections.
- **Front-end controllers** route data to and from cache via the internal data bus.
- When the cache receives the write data, the controller sends an acknowledgment message back to the host.

1.14.2 Cache

- **Cache** is semiconductor memory where data is placed temporarily to reduce the time required to service I/O requests from the host.
- Cache improves storage system **performance** by isolating hosts from the mechanical delays associated with rotating disks or hard disk drives (HDD).
- Rotating disks are the slowest component of an intelligent storage system. Data access on rotating disks usually takes several millisecond because of seek time and rotational latency.
- Accessing data from cache is fast and typically takes less than a millisecond.
- On intelligent arrays, write data is first placed in cache and then written to disk.

Structure Of Cache

- Cache is organized into pages, which is the smallest unit of cache allocation. The size of a cache page is configured according to the application I/O size.
- Cache consists of the **data store** and **tag RAM**.
- The data store holds the data whereas the tag RAM tracks the location of the data in the data store (see Fig 1.22) and in the disk.
- Entries in tag RAM indicate where data is found in cache and where the data belongs on the disk.
- Tag RAM includes a dirty bit flag, which indicates whether the data in cache has been committed to the disk.
- It also contains time-based information, such as the time of last access, which is used to identify cached information that has not been accessed for a long period and may be freed up.

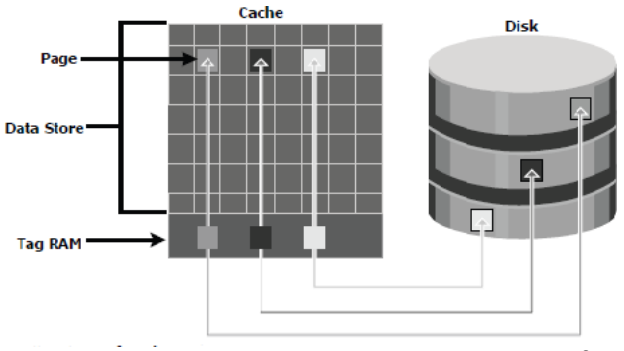


Fig 1.22: Structure of cache

Read Operation with Cache

- When a host issues a read request, the storage controller reads the tag RAM to determine whether the required data is available in cache.
- If the requested data is found in the cache, it is called a **read cache hit** or **read hit** and data is sent directly to the host, without any disk operation (see Fig 1.23[a]).This provides a fast response time to the host (about a millisecond).
- If the requested data is not found in cache, it is called a **cache miss** and the data must be read from the disk. The back-end controller accesses the appropriate disk and retrieves the requested data. Data is then placed in cache and is finally sent to the host through the front- end controller.
- Cache misses increase I/O response time.
- A **Pre-fetch**, or **Read-ahead,** algorithm is used when read requests are sequential. In a sequential read request, a contiguous set of associated blocks is retrieved. Several other blocks that have not yet been requested by the host can be read from the disk and placed into cache in advance. When the host subsequently requests these blocks, the read operations will be read hits.
- This process significantly improves the response time experienced by the host.
- The intelligent storage system offers *fixed* and *variable prefetch sizes*.
- In **fixed pre-fetch**, the intelligent storage system pre-fetches a fixed amount of data. It is most suitable when I/O sizes are uniform.
- In **variable pre-fetch**, the storage system pre-fetches an amount of data in multiples of the size of the host request.

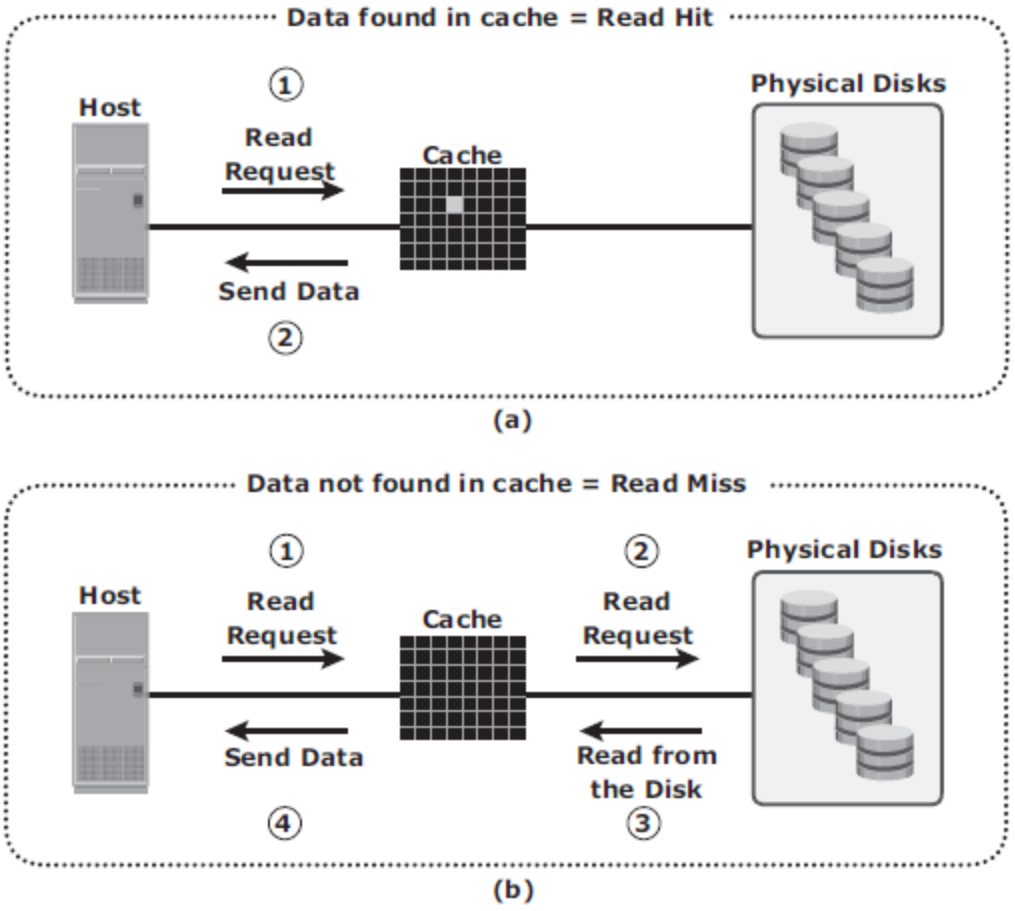


Fig 1.23 : Read hit and read miss

Write Operation with Cache

- Write operations with cache provide performance advantages over writing directly to disks.
- When an I/O is written to cache and acknowledged, it is completed in far less time (from the host's perspective) than it would take to write directly to disk.
- *Sequential writes* also offer opportunities for optimization because many smaller writes can be coalesced for larger transfers to disk drives with the use of cache.
- **A write operation** with cache is implemented in the following ways:
- **Write-back cache:** Data is placed in cache and an acknowledgment is sent to the host immediately. Later, data from several writes are committed to the disk. Write response times are much faster, as the write operations are isolated from the mechanical delays of the disk. However, uncommitted data is at risk of loss in the event of cache failures.
- **Write-through cache**: Data is placed in the cache and immediately written to the disk, and an acknowledgment is sent to the host. Because data is committed to disk as it arrives,

the risks of data loss are low but write response time is longer because of the disk operations.

- Cache can be bypassed under certain conditions, such as large size write I/O.
- In this implementation, if the size of an I/O request exceeds the predefined size, called **write aside size**, writes are sent to the disk directly to reduce the impact of large writes consuming a large cache space.
- This is useful in an environment where cache resources are constrained and cache is required for small random I/Os.

Cache Implementation

- Cache can be implemented as either **dedicated cache** or **global cache**.
- With **dedicated cache**, separate sets of memory locations are reserved for reads and writes.
- In **global cache**, both reads and writes can use any of the available memory addresses.
- Cache management is more efficient in a global cache implementation because only one global set of addresses has to be managed.
- Global cache allows users to specify the percentages of cache available for reads and writes for cache management.

Cache Management

- Cache is a finite and expensive resource that needs proper management.
- Even though modern intelligent storage systems come with a large amount of cache, when all cache pages are filled, some pages have to be freed up to accommodate new data and avoid performance degradation.
- Various cache management algorithms are implemented in intelligent storage systems to proactively maintain a set of free pages and a list of pages that can be potentially freed up whenever required.
- The most commonly used algorithms are listed below:
  - ✓ **Least Recently Used (LRU):** An algorithm that continuously monitors data access in cache and identifies the cache pages that have not been accessed for a long time. LRU either frees up these pages or marks them for reuse. This algorithm is based on the assumption that data which hasn't been accessed for a while will not be requested by the host.

- ✓ **Most Recently Used (MRU):** In MRU, the pages that have been accessed most recently are freed up or marked for reuse. This algorithm is based on the assumption that recently accessed data may not be required for a while

- As cache fills, the storage system must take action to **flush dirty pages** (data written into the cache but not yet written to the disk) to manage space availability.
- **Flushing** is the process that commits data from cache to the disk.
- On the basis of the I/O access rate and pattern, high and low levels called **watermarks** are set in cache to manage the flushing process.
- **High watermark (HWM)** is the cache utilization level at which the storage system starts high-speed flushing of cache data.
- **Low watermark (LWM)** is the point at which the storage system stops flushing data to the disks.
- The *cache utilization level*, as shown in Fig 1.24, drives the mode of flushing to be used:
  - ✓ **Idle flushing:** Occurs continuously, at a modest rate, when the cache utilization level is between the high and low watermark.
  - ✓ **High watermark flushing:** Activated when cache utilization hits the high watermark. The storage system dedicates some additional resources for flushing. This type of flushing has some impact on I/O processing.
  - ✓ **Forced flushing:** Occurs in the event of a large I/O burst when cache reaches 100 percent of its capacity, which significantly affects the I/O response time. In forced flushing, system flushes the cache on priority by allocating more resources.

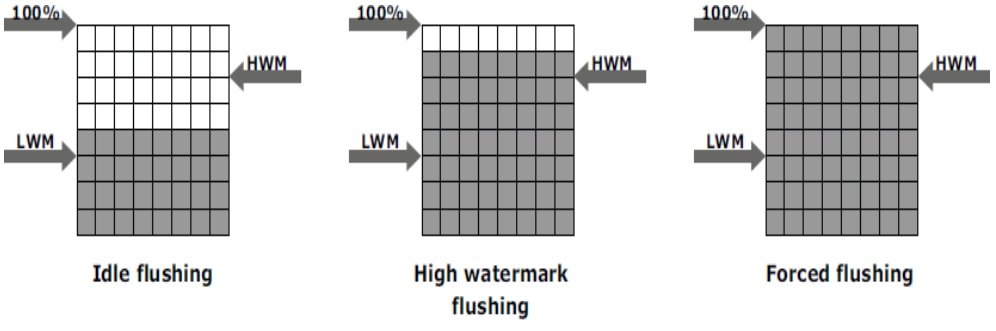


Fig 1.24 : Types of flushing

Cache Data Protection

- Cache is volatile memory, so a power failure or any kind of cache failure will cause loss of the data that is not yet committed to the disk.

- This risk of losing uncommitted data held in cache can be mitigated using
  i. cache mirroring
  ii. cache vaulting

- Cache mirroring
  - ✓ Each write to cache is held in two different memory locations on two independent memory cards. In the event of a cache failure, the write data will still be safe in the mirrored location and can be committed to the disk.
  - ✓ Reads are staged from the disk to the cache, therefore, in the event of a cache failure, the data can still be accessed from the disk.
  - ✓ In cache mirroring approaches, the problem of maintaining *cache coherency* is introduced.
  - ✓ Cache coherency means that data in two different cache locations must be identical at all times. It is the responsibility of the array operating environment to ensure coherency.

- Cache vaulting
  - ✓ The risk of data loss due to power failure can be addressed in various ways:
    - powering the memory with a battery until the AC power is restored
    - using battery power to write the cache content to the disk.
  - ✓ If an extended power failure occurs, using batteries is not a viable option.
  - ✓ This is because in intelligent storage systems, large amounts of data might need to be committed to numerous disks, and batteries might not provide power for sufficient time to write each piece of data to its intended disk.
  - ✓ Storage vendors use a set of physical disks to dump the contents of cache during power failure. This is called *cache vaulting* and the disks are called vault drives.
  - ✓ When power is restored, data from these disks is written back to write cache and then written to the intended disks.

1.14.3 Back End

- The **back end** provides an interface between cache and the physical disks.
- It consists of two components:
  i. Back-end ports
  ii. Back-end controllers.
- The back end controls data transfers between cache and the physical disks.
- From cache, data is sent to the back end and then routed to the destination disk.

- Physical disks are connected to *ports* on the back end.
- The *back end controller* communicates with the disks when performing reads and writes and also provides additional, but limited, temporary data storage.
- The algorithms implemented on back-end controllers provide error detection and correction, and also RAID functionality.
- For high data protection and high availability, storage systems are configured with dual controllers with multiple ports.

1.14.4 Physical Disk

- A physical disk stores data persistently.
- Physical disks are connected to the back-end storage controller and provide persistent data storage.
- Modern intelligent storage systems provide support to a variety of disk drives with different speeds and types, such as FC, SATA, SAS, and flash drives.
- They also support the use of a mix of flash, FC, or SATA within the same array.

1.6 Types/ Implementation of Intelligent Storage Systems

- An intelligent storage system is divided into following two categories:
    1. High-end storage systems
    2. Midrange storage systems
- High-end storage systems have been implemented with active-active configuration, whereas midrange storage systems have been implemented with active-passive configuration.
- The distinctions between these two implementations are becoming increasingly insignificant.

1.15.1 High-end Storage Systems

- High-end storage systems, referred to as **active-active arrays,** are generally aimed at large enterprises for centralizing corporate data. These arrays are designed with a large number of controllers and cache memory.
- An active-active array implies that the host can perform I/Os to its LUNs across any of the available paths (see Fig 1.25).

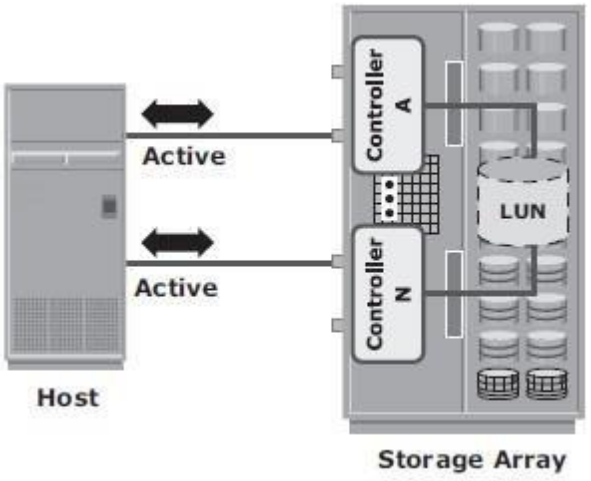


Fig 1.25 : Active-active configuration

Advantages of High-end storage:

- Large storage capacity
- Large amounts of cache to service host I/Os optimally
- Fault tolerance architecture to improve data availability
- Connectivity to mainframe computers and open systems hosts Availability of multiple front-end ports and interface protocols to serve a large number of hosts
- Availability of multiple back-end Fibre Channel or SCSI RAID controllers to manage disk processing
- Scalability to support increased connectivity, performance, and storage
- capacity requirements
- Ability to handle large amounts of concurrent I/Os from a number of servers and applications
- Support for array-based local and remote replication

1.15.2 Midrange Storage System

- Midrange storage systems are also referred to as **Active-Passive Arrays** and they are best suited for small- and medium-sized enterprises.
- They also provide optimal storage solutions at a *lower cost*.
- In an *active-passive* array, a host can perform I/Os to a LUN only through the paths to the **owning controller** of that LUN. These paths are called *Active Paths*. The other paths are *passive* with respect to this LUN.

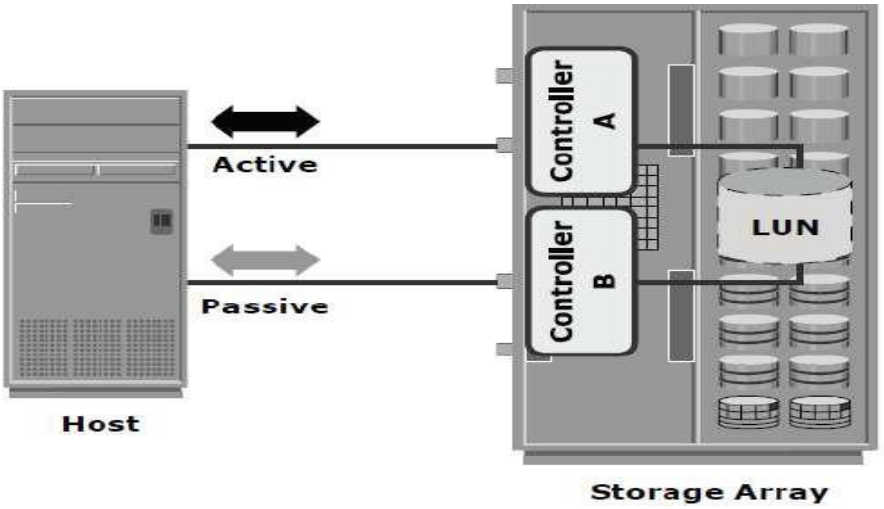


Fig 1.26 : Active-passive configuration

- As shown in Fig 1.26, the host can perform reads or writes to the LUN only through the path to controller A, as controller A is the owner of that LUN.
- The path to controller B remains **Passive** and no I/O activity is performed through this path.
- Midrange storage systems are typically designed with two controllers, each of which contains host interfaces, cache, RAID controllers, and disk drive interfaces.
- Midrange arrays are designed to meet the requirements of small and medium enterprise applications; therefore, they host less storage capacity and cache than high-end storage arrays.
- There are also fewer front-end ports for connection to hosts.
- But they ensure high redundancy and high performance for applications with predictable workloads.
- They also support array-based local and remote replication.

1.7 Virtual Storage Provisioning

- **Virtual provisioning** enables creating and presenting a LUN with more capacity than is physically allocated to it on the storage array.
- The LUN created using virtual provisioning is called a ***thin LUN*** to distinguish it from the traditional LUN.
- Thin LUNs do not require physical storage to be completely allocated to them at the time they are created and presented to a host.
- Physical storage is allocated to the host "*on-demand*" from a *shared pool* of physical

capacity.

- A *shared pool* consists of physical disks.
- A shared pool in virtual provisioning is analogous to a *RAID group*, which is a collection of drives on which LUNs are created.
- Similar to a RAID group, a shared pool supports a single RAID protection level. However, unlike a RAID group, a shared pool might contain large numbers of drives.
- Shared pools can be homogeneous (containing a single drive type) or heterogeneous (containing mixed drive types, such as flash, FC, SAS, and SATA drives).
- Virtual provisioning enables more efficient allocation of storage to hosts.
- Virtual provisioning also enables oversubscription, where more capacity is presented to the hosts than is actually available on the storage array.
- Both shared pool and thin LUN can be expanded non-disruptively as the storage requirements of the hosts grow.
- Multiple shared pools can be created within a storage array, and a shared pool may be shared by multiple thin LUNs.
- Fig 1.27 illustrates the provisioning of thin LUNs.

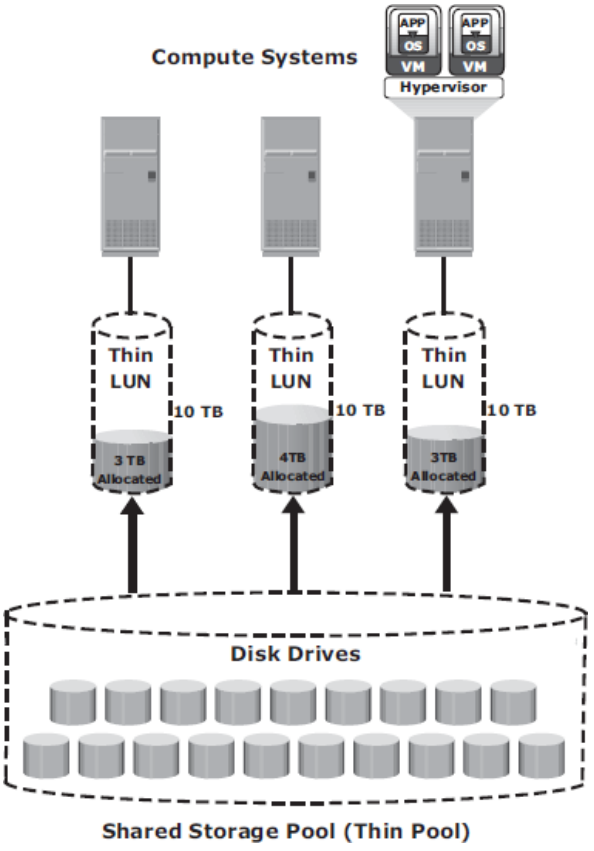


Fig 1.27: Virtual Provisioning

Comparison between Virtual and Traditional Storage Provisioning

- Virtual provisioning improves storage capacity utilization and simplifies storage management.
- Figure 1.28 shows an example, comparing virtual provisioning with traditional storage provisioning.

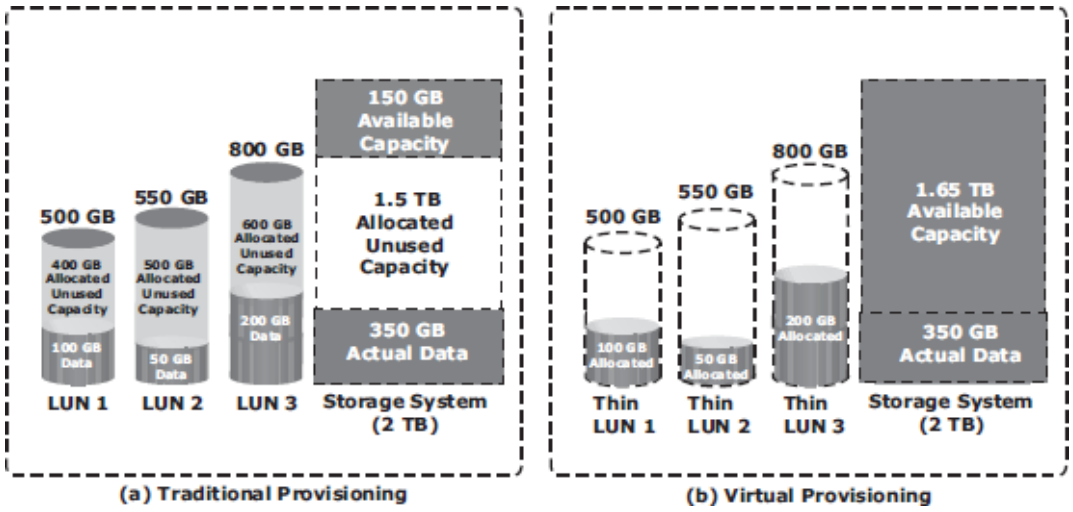


Fig 1.28: Traditional versus Virtual Provisioning

- With **traditional provisioning,** three LUNs are created and presented to one or more hosts (see Fig 1.28 [a]). The total storage capacity of the storage system is 2 TB.
- The allocated capacity of LUN 1 is 500 GB, of which only 100 GB is consumed, and the remaining 400 GB is unused. The size of LUN 2 is 550 GB, of which 50 GB is consumed, and 500 GB is unused. The size of LUN 3 is 800 GB, of which 200 GB is consumed, and 600 GB is unused.
- In total, the storage system has 350 GB of data, 1.5 TB of allocated but unused capacity, and only 150 GB of remaining capacity available for other applications.

  Now consider the same 2 TB storage system with **virtual provisioning** (see Fig 1.28 [b]).
- Here, three *thin LUNs* of the same sizes are created. However, there is no allocated unused capacity.In total, the storage system with virtual provisioning has the same 350 GB of data, but 1.65 TB of capacity is available for other applications, whereas only 150 GB is available in traditional storage provisioning.

STORAGE NETWORKING TECHNOLOGIES AND VIRTUALIZATION

Business Needs and TechnologyChallenges

- Companies are experiencing an explosive growth ininformation.
- This information needs to be stored, protected, optimized, and managedefficiently.
- Challenging task for data-centermanagers:
  Providing low-cost, high-performance information-management-solution (ISM).
  - ISM must provide the following functions:

1) Just-in-time information tousers

- Information must be available to users when they needit.
- Following key challenges must beaddressed:
  → explosive growth in online-storage
  → creation of new servers and applications
  → spread of mission-critical data throughout the company and
  → demand for 24×7 data-availability

2) Integration of information infrastructure withbusiness-processes

- Storage-infrastructure must be integrated with business-processes w/o compromising onsecurity

3) Flexible and resilient storagearchitecture

- Storage-infrastructure must provide flexibility that aligns with changingbusiness-requirements.
- Storage should scale without compromising performance requirements of theapplications.
- At the same time, the total cost of managing information must below.

- Direct-attached storage (DAS) is often referred to as a stovepiped storageenvironment.
- Problem withDAS:
  1) Hosts "own" thestorage.
  Hence, it is difficult to manage and share resources on these separated storage-devices.
- Solution:
  1) Efforts to organize this dispersed data led to the emergence of the storage area network (SAN).

  SAN is a high-speed dedicated network of servers and shared storage. Commom SAN deployments are:
  - ✓ FCSAN
  - ✓ IPSAN

2.1 Fibre Channel:Overview

- The FC architecture forms the fundamental construct of the SANinfrastructure.
- **Fibre Channel** is a high-speed network technology that runs on high-speed optical fiber cables (preferred for front-end SAN connectivity) and serial copper cables (preferred for back-end diskconnectivity).
- The FC technology was created to meet the demand for increased speeds of data transfer among computers, servers, and mass storagesubsystems.

2.2 The SAN and ItsEvolution

A storage area network (SAN) carries data between servers or hosts and storage devices through fibre channel switches as shown in Figure 6-1.

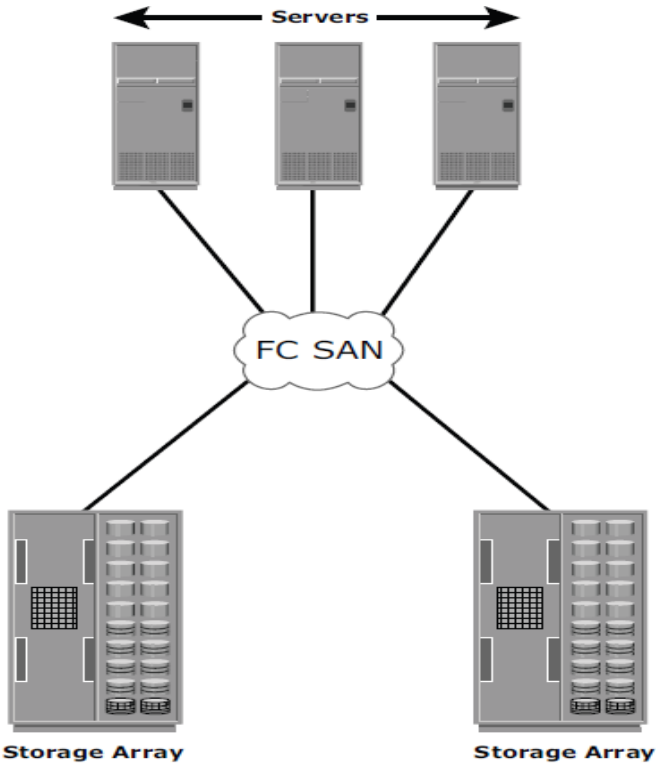


Figure 6-1: SAN implementation

A SAN enables storage consolidation and allows storage to be shared across multiple servers.

A SAN provides the physical communication infrastructure and enables secure and robust communication between host and storage devices.

The SAN management interface organizes connections and manages storage elements and hosts. In its earliest implementation, the SAN was a simple grouping of hosts and the associated storage that was connected to a network using a hub as a connectivity device.

This configuration of a SAN is known as a Fibre**Channel Arbitrated Loop (FC-AL).** Use of hubs resulted in isolated FC-AL SAN islands because hubs provide ***limited connectivity and bandwidth***.

The inherent limitations associated with hubs gave way to high-performance FC *switches*.

The switched fabric topologies improved connectivity and performance, which enabled SANs to be ***highly scalable.***

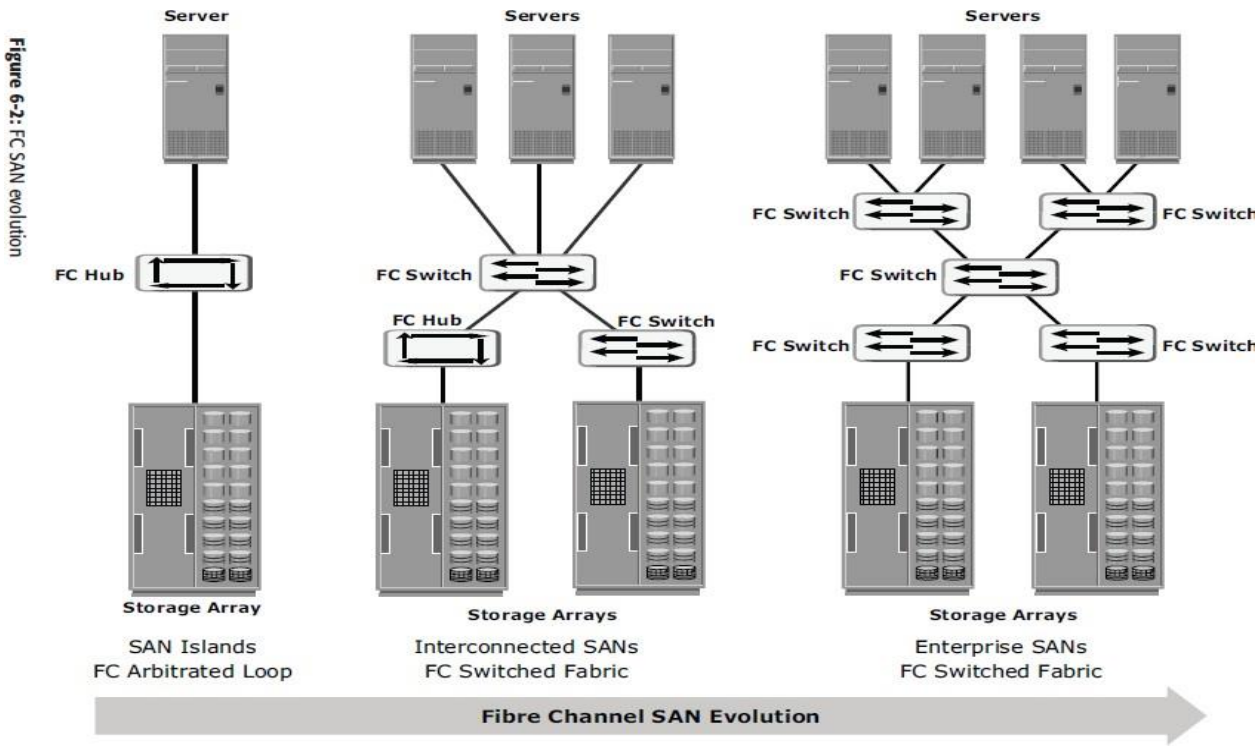


Figure 6-2: FC SAN evolution

Fig. 6.2. FC SAN Evolution

This enhanced data accessibility to applications across the enterprise. FC-AL has been abandoned for SANs due to its limitations, but still survives as a disk-drive interface. Figure 6-2 illustrates the FC SAN evolution from FC-AL to enterprise SANs.

2.3 Components ofSAN

- Components of FC SAN infrastructureare:

1) **NodePorts,**
2) **Cabling,**
3) **Connectors,**
4) **Interconnecting Devices (Such As Fc Switches OrHubs),**
5) **San ManagementSoftware**.

Node Ports

- In fibre channel, devices such as hosts, storage and tape libraries are all referred toas **Nodes.**
- Each node is a **source or destination** of information for one or morenodes.
- Each node requires one or more ports to provide a physical interface for communicating with othernodes.
- A port operates in full-duplex data transmission mode with a **transmit (Tx) link anda receive (Rx) link** (see Fig 2.1).

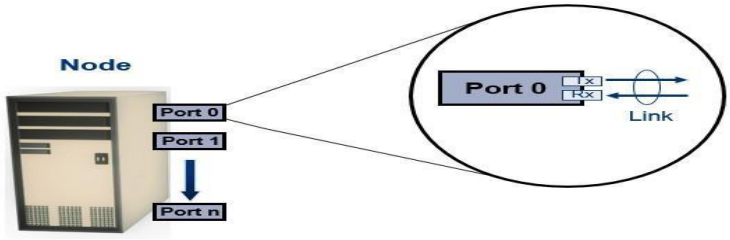


Fig 2.1: Nodes, Ports, links

Cabling

- SAN implementations use optical fibercabling.
- Copper can be used for shorter distances for back-endconnectivity
- Optical fiber cables carry data in the form oflight.
- There are two types of optical cables :**Multi-Mode AndSingle-Mode.**

1) **Multi-mode fiber (MMF)** cable carries multiple beams of light projected at differentangles simultaneously onto the core of the cable (see Fig 2.2(a)).
    - 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 *modaldispersion*.
    - MMFs are generally used within data centers for shorter distanceruns
2) **Single-mode fiber (SMF)** carries a single ray of light projected at the center of the core(see Fig 2.2(b)).

- In an SMF transmission, a single light beam travels in a straight line through the core of thefiber.
- 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 10km).
- Asingle-modecableisusedforlong-distancecableruns,limitedonlybythepower of the laser at the transmitter and sensitivity of the receiver.
- SMFs are used for longerdistances.

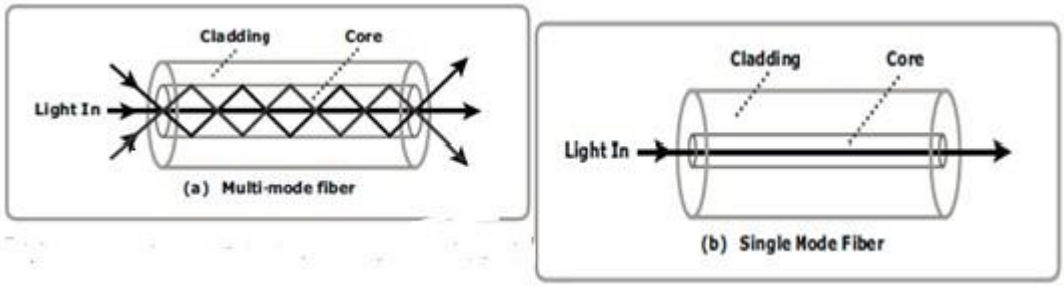


Fig 2.2: Multimode fiber and single-mode fiber

Connectors

- They are attached at the end of the cable to enable swift connection and disconnection of the cable to and from aport.
- A **Standard connector (SC)** (see Fig 2.3 (a)) and a **Lucent connector (LC)** (see Fig 2.3 (b)) are two commonly used connectors for fiber opticcables.
- 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 2.3 depicts a Lucent connector and a Standardconnector.
- 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 Fig 2.3(c)).

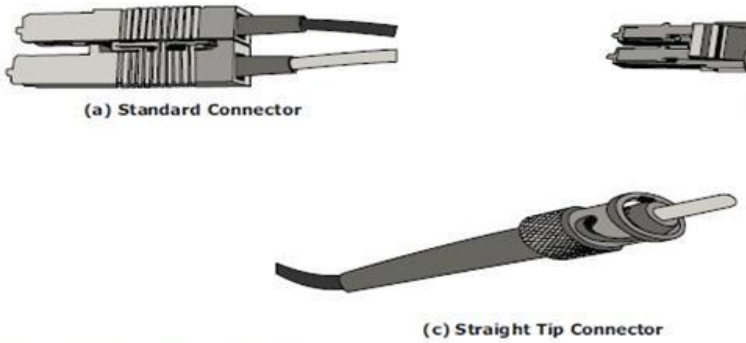


Fig 2.3: SC,LC, and ST connectors

Interconnect Devices

The commonly used interconnecting devices in SAN are

1) **Hubs**
2) **Switches**
3) **Directors**

- **Hubs** are used as communication devices in FC-AL implementations. Hubs physically connect nodes in a logical loop or a physical startopology.
- 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 inSANs.
- **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 bandwidthaggregation.
- Switches are availablewith:
  - ✓ Fixed portcount
  - ✓ Modular design : port count is increased by installing additional port cards to open slots.
- **Directors are larger than switches** and are deployed for data centerimplementations.
- The function of directors is similar to that of FC switches, but directors havehigher port count and fault tolerancecapabilities.
- Port card or blade has multiple ports for connecting nodes and other FCswitches

SAN Management Software

- SAN management software manages the interfaces between hosts, interconnect devices, and storagearrays.
- The software provides a view of the SAN environment and enables management of various resources from one centralconsole.
- 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*

2.4 FC Connectivity

The FC architecture supports three basic interconnectivity options:

1) **Point-To-point,**
2) **Arbitrated Loop(Fc-AL),**
3) **FC Switched**

FabricPoint-to-Point

- **Point-to-point** is the simplest FC configuration — two devices are connected directly to each other, as shown in Fig2.4.
- This configuration provides a dedicated connection for data transmission betweennodes.
- The point-to-point configuration offers limited connectivity, as only two devices can communicate with each other at a giventime.
- It cannot be scaled to accommodate a large number of network devices. Standard DAS uses point to- pointconnectivity.

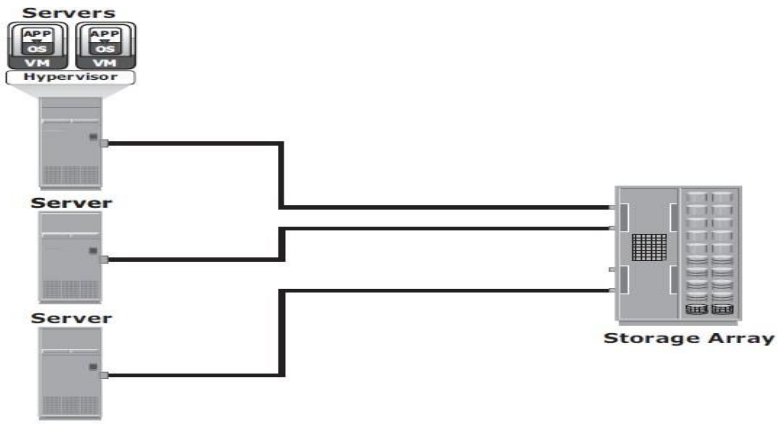


Fig 2.4: Point-to-point connectivity

Fibre Channel Arbitrated Loop

- In the FC-AL configuration, devices are attached to a shared loop, as shown in Fig2.5.
- FC-AL has the characteristics of a **token ring topology and a physical startopology**.
- In FC-AL, each device contends with other devices to perform I/O operations. Devices on the loop must "arbitrate" to gain control of theloop.
- At any given time, only one device can perform I/O operations on theloop.
- FC-AL implementations may also use hubs whereby the arbitrated loop is physically connected in a startopology.

The FC-AL configuration has the following limitations in terms of scalability:

- FC-AL shares the bandwidth in theloop.
- 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-ALtopology.
- FC-AL uses 8-bit addressing. It can support up to 127 devices on aloop.
- Adding or removing a device results in loop re-initialization, which can cause a momentary pause in looptraffic.

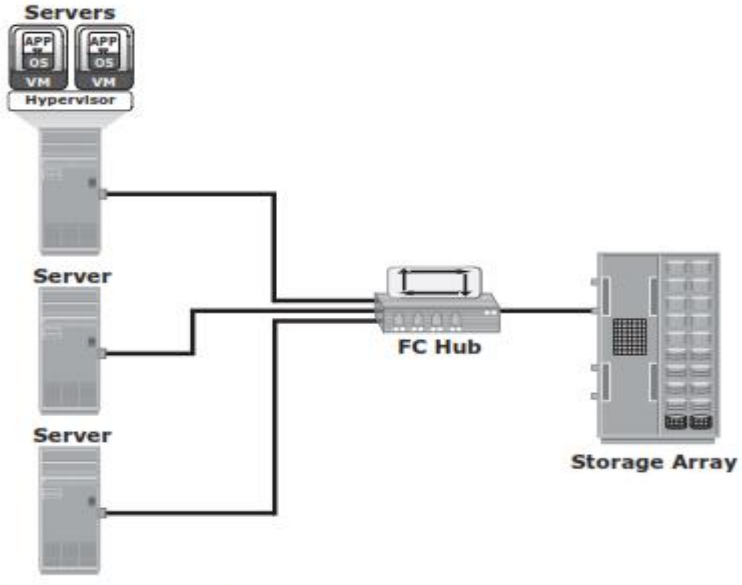


Figure 5-7: Fibre Channel Arbitrated Loop

Fig 2.5: Fibre Channel Arbitrated Loop

Fibre Channel Switched Fabric(FC-SW)

- FC-SW provides dedicated data path andscalability.
- The addition and removal of a device doesnot affect the on-going traffic between other devices.
- FC-SW is referred to as **Fabricconnect**.
- 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 ofswitches.
- Each switch in a fabric contains a a unique domain identifi er, which is part of the fabric's addressingscheme.
- 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, largerfabric.
- 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 ofnodes.

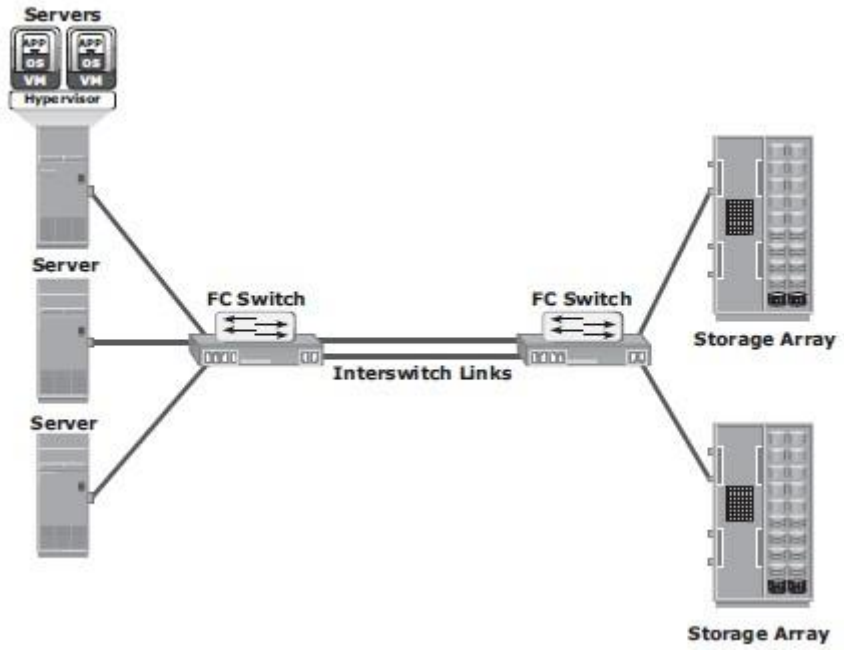


Fig 2.6: Fibre Channel switched Fabric

- A Fabric may containtiers.
- The number of tiers in a fabric is based on the number of switches between two points that are farthest from eachother

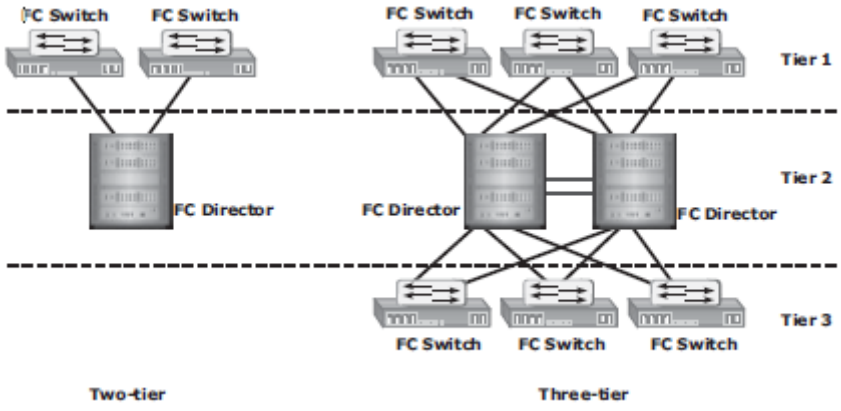


Fig 2.7: Tiered structure of Fibre Channel switched Fabric

FC-SW Transmission

- FC-SW uses switches that can switch data traffi c between nodes directly through switch ports.
- Frames are routed between source and destination by thefabric.

Node A want to communicate with Node B

1. High priority initiator, Node A inserts the ARB frame in the loop.
2. ARB frame is passed to the next node (Node D) in the loop.
3. Node D receives high priority ARB, therefore remains idle.
4. ARB is forwarded to next node (Node C) in the loop.
5. Node C receives high priority ARB, therefore remains idle.
6. ARB is forwarded to next node (Node B) in the loop.
7. Node B receives high priority ARB, therefore remains idle and
8. ARB is forwarded to next node (Node A) in the loop.
9. Node A receives ARB back; now it gains control of the loop and can start communicating with target Node B.

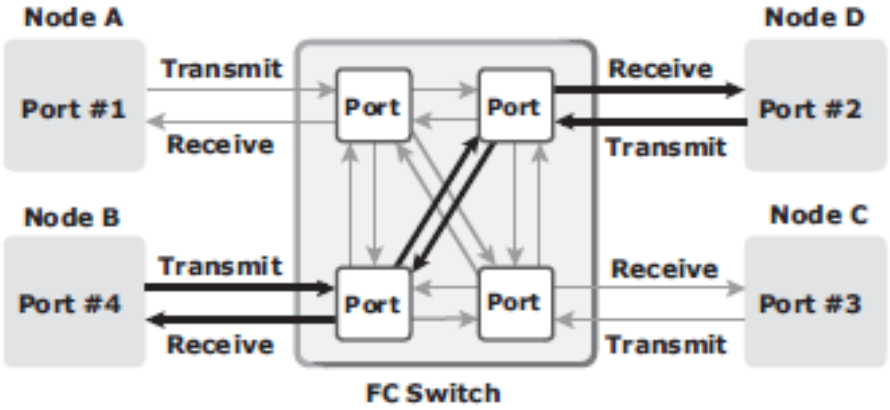


Fig 2.8: Data transmission in fibre channel switched fabric

2.5 Fibre ChannelArchitecture

- Connections in a SAN are accomplished usingFC.
- ***Fibre Channel Protocol (FCP)* is the implementation of serial SCSI-3 over an FC network.** In the FCP architecture, all external and remote storage devices attached to the SAN appear as local devices to the host operatingsystem.
- The key advantages of FCP are asfollows:
  - Sustained transmission bandwidth over longdistances.
  - Support for a larger number of addressable devices over anetwork.
  - Theoretically, FC can support over 15 million device addresses on anetwork.
  - Exhibits the characteristics of channel transport and provides speeds up to 8.5 Gb/s (8GFC).

Fibre Channel Protocol Stack

- It is easier to understand a communication protocol by viewing it as a structure of independentlayers.
- FCP defines the communication protocol in five layers:FC-0 through FC-4 (except FC-3 layer, which is notimplemented).
- In a layered communication model, the peer layers on each node talk to each other through definedprotocols.
- Fig 2.9 illustrates the fibre channel protocolstack.

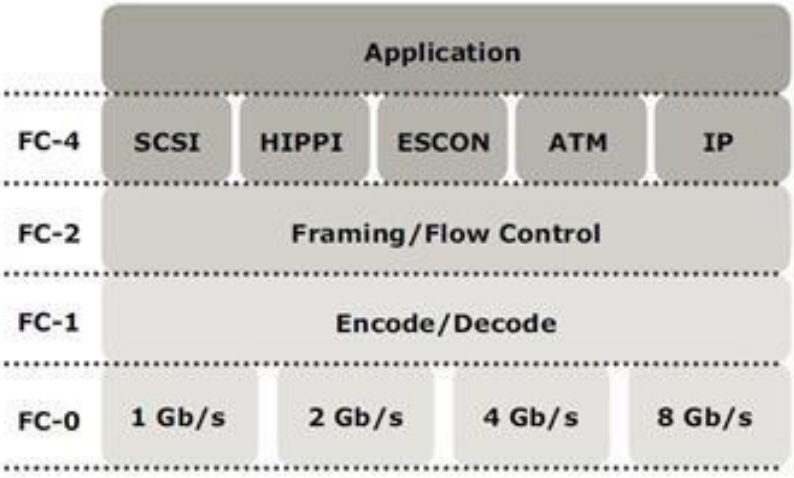


Fig 2.9: Fibre Channel Protocol Stack

- **FC-4 Upper LayerProtocol**
  - ✓ FC-4 is the uppermost layer in the FCPstack.
  - ✓ This layer defines the application interfaces and the way **Upper Layer Protocols (ULPs) are mapped to the lower FClayers.**
  - ✓ The FC standard defines several protocols that can operate on the FC-4 layer (see Fig 2.9). Some of the protocols include SCSI, HIPPI Framing Protocol, Enterprise Storage Connectivity (ESCON), ATM, andIP.
- **FC-2 TransportLayer**
  - ✓ The FC-2 is the transport layer that contains the payload, addresses of the source and destination ports, and link controlinformation.
  - ✓ 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, androuting.**
- **FC-1 TransmissionProtocol**
  - ✓ This layer defines the transmission protocol that includes **serial encoding and decoding rules, special characters used, and errorcontrol.**
  - ✓ At the transmitter node, an 8-bit character is encoded into a 10-bit transmissions character.

- ✓ This character is then transmitted to the receivernode.
- ✓ 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-bitcharacter.

- **FC-0 PhysicalInterface**
  - ✓ FC-0 is the lowest layer in the FCPstack.
  - ✓ This layer defines the physical interface, media, and transmission of rawbits.
  - ✓ The FC-0 specification includes cables, connectors, and optical and electrical parameters for a variety of datarates.
  - ✓ The FC transmission can use both electrical and opticalmedia.

2.6 Fibre ChannelAddressing

- An FC address is **dynamically assigned** when a port logs on to thefabric.
- The FC address has a distinct format, as shown in Fig 2.10. The addressing mechanism provided here corresponds to the fabric with the switch as an interconnectingdevice.
- The first field of the FC address contains the domain ID of the switch (see Fig2.10).
- A *domain ID* is a unique number provided to each switch in thefabric.
- This is an 8-bit field, there are only 239 available addresses for domain ID because some addresses are deemed special and reserved for fabric managementservices.
- For example, FFFFFC is reserved for the name server, and FFFFFE is reserved for the fabric loginservice.
- The *area ID* is used to identify a group of switch ports used for connecting nodes. An example of a group of ports with a common area ID is a port card on theswitch.
- The last field, the *port ID*, identifies the port within thegroup.
- The maximum possible number of node ports in a switched fabric is calculated as: 239 domains ¥ 256 areas ¥ 256 ports =15,663,104

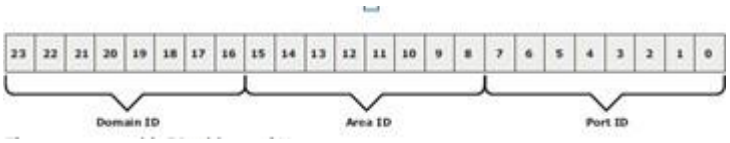


Fig 2.10 24-bit FC address of N_port

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 theHBA.

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

FCFrame

An FC frame shown in 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.

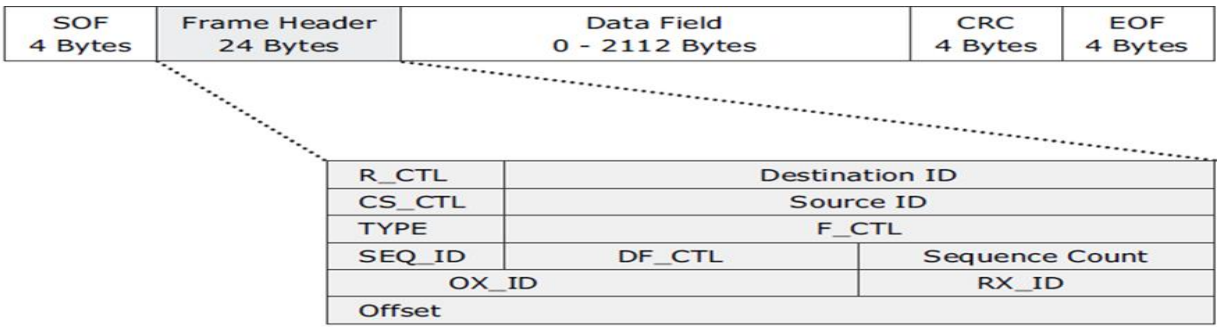


Figure 6-17: FC frame

The frame header also defines the following fields:

a) **Routing Control (R_CTL)**: This field denotes whether the frame is a link control frame
oradataframe.Linkcontrolframesarenondataframesthatdonotcarryanypayload.

These frames are used for setup and messaging. In contrast, data frames carry the payload and are used for data transmission.

b) **Class Specific Control (CS_CTL)**: This field specifies link speeds for class 1 and class 4 datatransmission.

c) **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 anFC.

d) **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 thepayload.

e) **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 theexchange.

Structure and Organization of FCData

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

i. **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 thetarget.

ii. **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 theULP.

iii. **Frame:** A frame is the fundamental unit of data transfer at Layer 2. Each frame can contain up to 2,112 bytes ofpayload.

FlowControl

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).

i. ***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. TheBB_Creditmechanism provides frame acknowledgment through the *Receiver Ready (R_RDY)*primitive.

ii. ***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 trafficonly.

Classes ofService

The FC standards define different classes of service to meet the requirements of a wide range of applications. The table below shows three classes of services and their features (Table 6-1).

Table 6-1: FC Class of Services

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

Another class of services is *class F,* which is intended for use by the switches communicating through ISLs. Class F is similar to Class 2, and it provides notification of non delivery of frames. Other defined Classes 4, 5, and 6 are used for specific applications.

2.7 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 Fig2.11).
- 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 thechange.
- 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-managementtraffic.
- 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 hasoccurred.

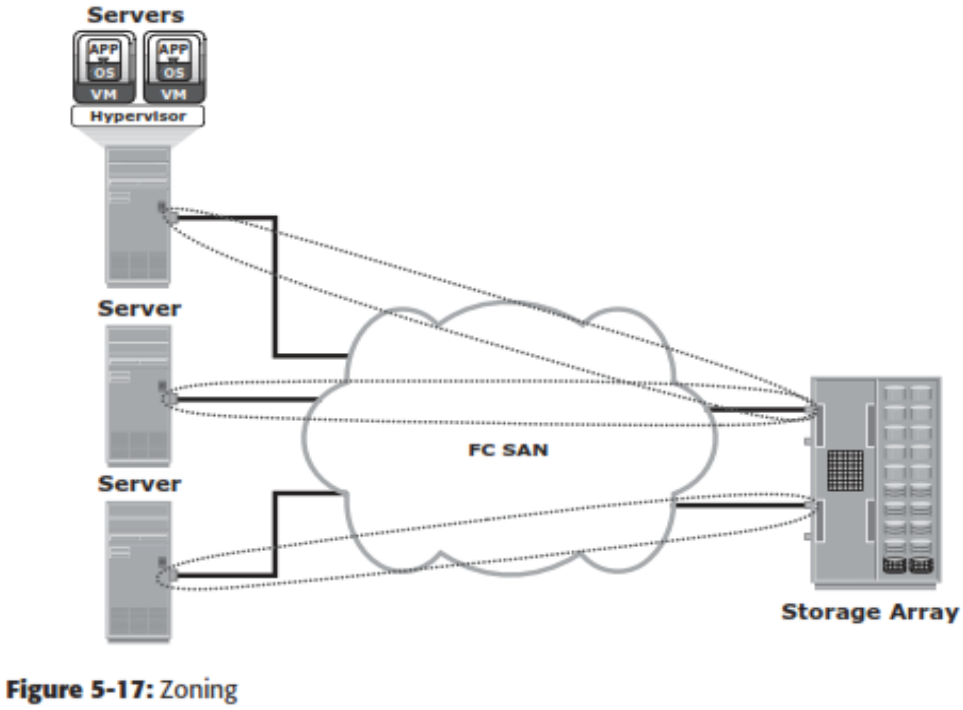


Figure 5-17: Zoning

Fig 2.11 Zoning

- Multiple zone sets may be defined in a fabric, but only one zone set can be active at a time.
- A **zone set is a set of zones** and a **zone is a set ofmembers**.
- A member may be in multiple zones. Members, zones, and zone sets form the hierarchy defined in the zoning process (see Fig2.12).

- **Members** are nodes within the SAN that can be included in azone.
- **Zones** comprise a set of members that have access to one another. A port or a node can be a member of multiplezones.
- **Zone sets** comprise a group of zones that can be activated or deactivated as a single entity in a fabric. Only one zone set per fabric can be active at atime.
- Zone sets are also referred to as *zoneconfigurations*.

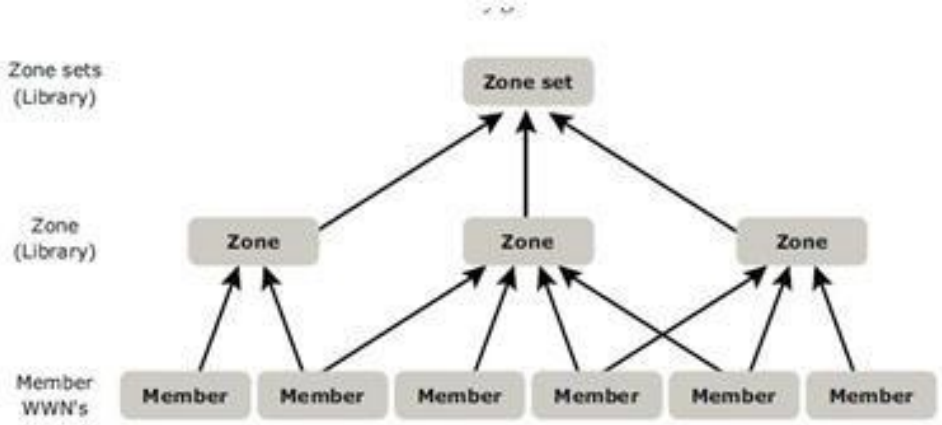


Fig 2.12: Members,Zones, and Zone sets

Types of Zoning

Zoning can be categorized into three types:

1) **Portzoning**
2) **WWNzoning**
3) **Mixed zoning**

Portzoning:

- It uses the **FC addresses** of the physical ports to definezones.
- In port zoning, access to data is determined by the physical switch port to which a node is connected.
- The **FC address is dynamically** assigned when the port logs on to the fabric. Therefore, any change in the fabric configuration affectszoning.
- Port zoning is also called **hardzoning.**
- Although this method is secure, it requires updating of zoning configuration information in the event of fabricreconfiguration.

WWN zoning:

- It uses World Wide Names to definezones.
- WWN zoning is also referred to as **softzoning**.
- A major advantage of WWN zoning is itsflexibility.
- It allows the SAN to be recabled without reconfiguring the zone information. This is possible because the WWN is static to the nodeport.

Mixed zoning:

- It combines the qualities of both WWN zoning and portzoning.
- Using mixed zoning enables a specific port to be tied to the WWN of anode.

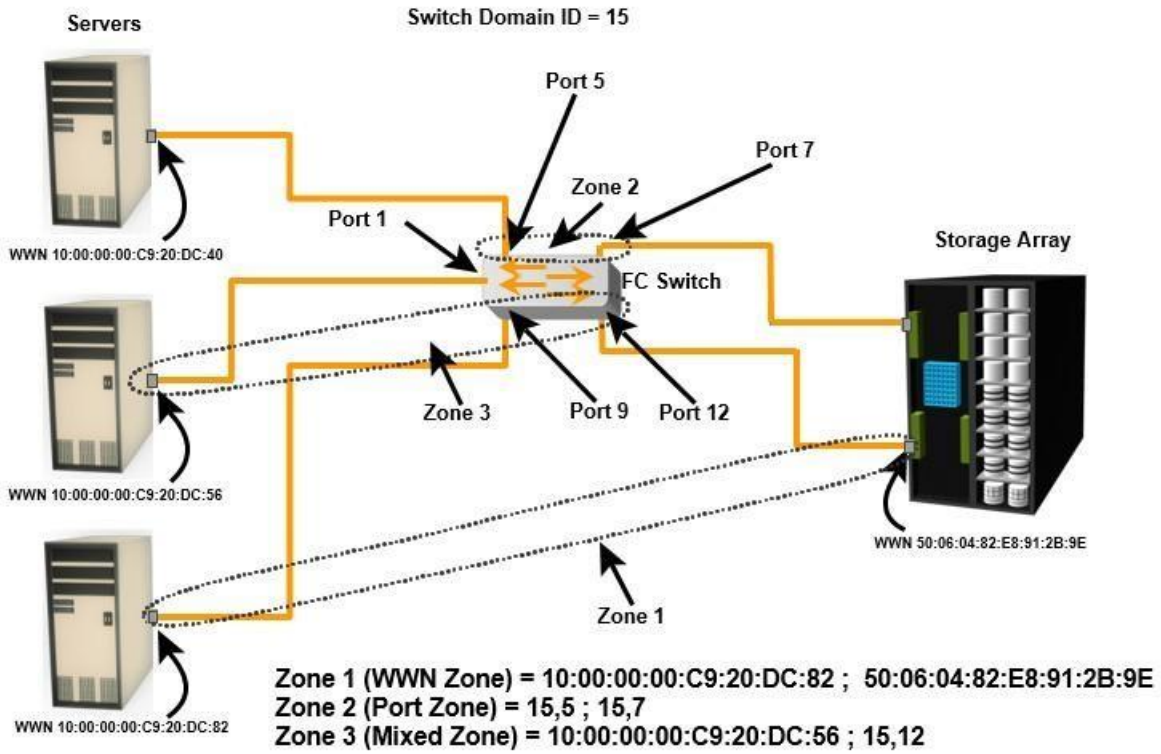


Fig 2.14: Types of Zoning

- Zoning is used in conjunction with LUN masking for controlling server access to storage. However, these are two different activities. Zoning takes place at the fabric level and LUN masking is done at the arraylevel.

2.8 FCTopologies

- Fabric design follows standard topologies to connect devices. There are two typesof topologies.
  - **MeshTopology**
  - **Core-EdgeFabric**

Mesh Topology

- In a mesh topology, **each switch is directly connected to other switches by usingISLs**.
- This topology promotes enhanced connectivity within theSAN.
- When the number of ports on a network increases, the number of nodes that can participate and communicate alsoincreases.
- A mesh topology may be one of the two types: **full mesh or partialmesh.**
- In a **full mesh, every switch is connected to every other switch** in thetopology.
- 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-storagetraffic.
- In a **partial mesh** topology, several hops or ISLs may be required for the traffic to reach its destination. Hosts and storage can be located anywhere in the fabric, and storage can be localized to a director or a switch in both mesh topologies. A full mesh topology with a symmetric design results in an even number of switches, whereas a partial mesh has an asymmetric design and may result in an odd number of switches. Fig 2.15 depicts both a full mesh and a partial meshtopology.

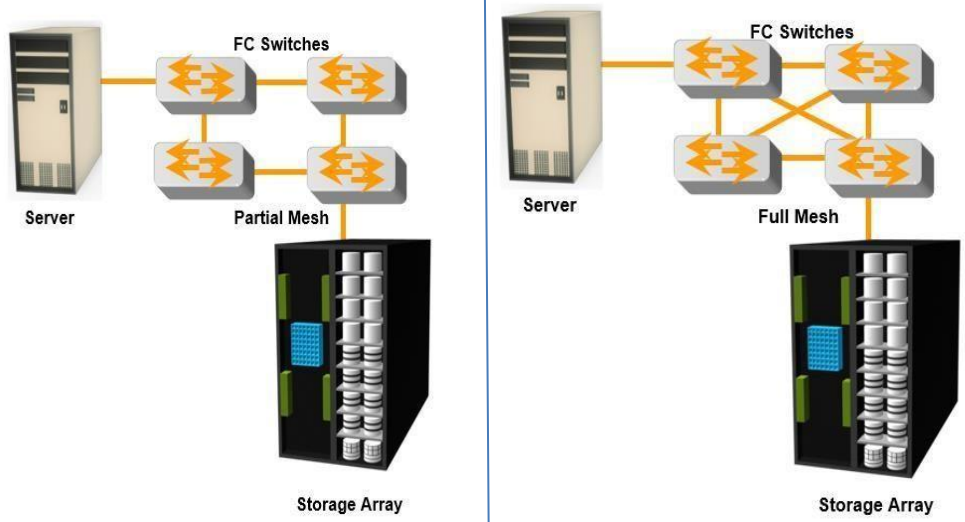


Fig a :Partial **Fig b:Full**

Fig 2.15: Partial and Full mesh Topologies

Core-Edge Fabric

- In the **core-edge fabric** topology, there are two types of switch tiers in thisfabric.
- The **edge tier** usually comprises switches and offers an inexpensive approach to adding more hosts in a fabric. The tier at the edge fans out from the tier at the core. The nodes on the edge can communicate with eachother.
- The **core tier** usually comprises **enterprise directors** that ensure high fabricavailability. Additionally all traffic has to either traverse through or terminate at thistier.
- In a two-tier configuration, all storage devices are connected to the core tier, facilitating fan-out.
- The host-to-storage traffic has to traverse one and two ISLs in a two-tier and three-tier configuration,respectively.
- The core-edge fabric topology increases connectivity within the SAN while conserving overall port utilization. If expansion is required, an additional edge switch can be connected to the core. This topology can have differentvariations.
- In a **single-core topology**, all hosts are connected to the edge tier and all storageis

connected to the core tier. Fig 2.16 depicts the core and edge switches in a single- core topology.

- A **dual-core topology** can be expanded to include more core switches. However, to maintain the topology, it is essential that new ISLs are created to connect each edge switch to the new core switch that is added. Fig 2.17 illustrates the core and edge switches in a dual-coretopology.

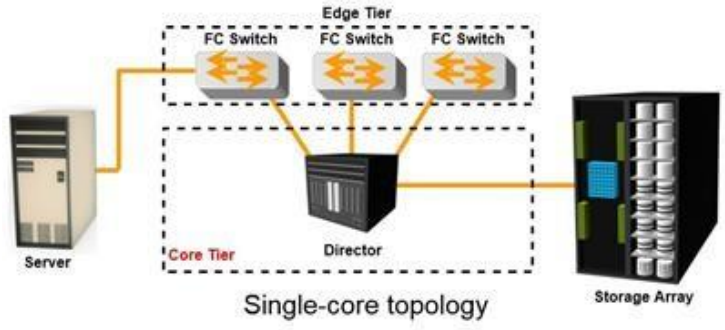


Fig 2.16: Single-core topology

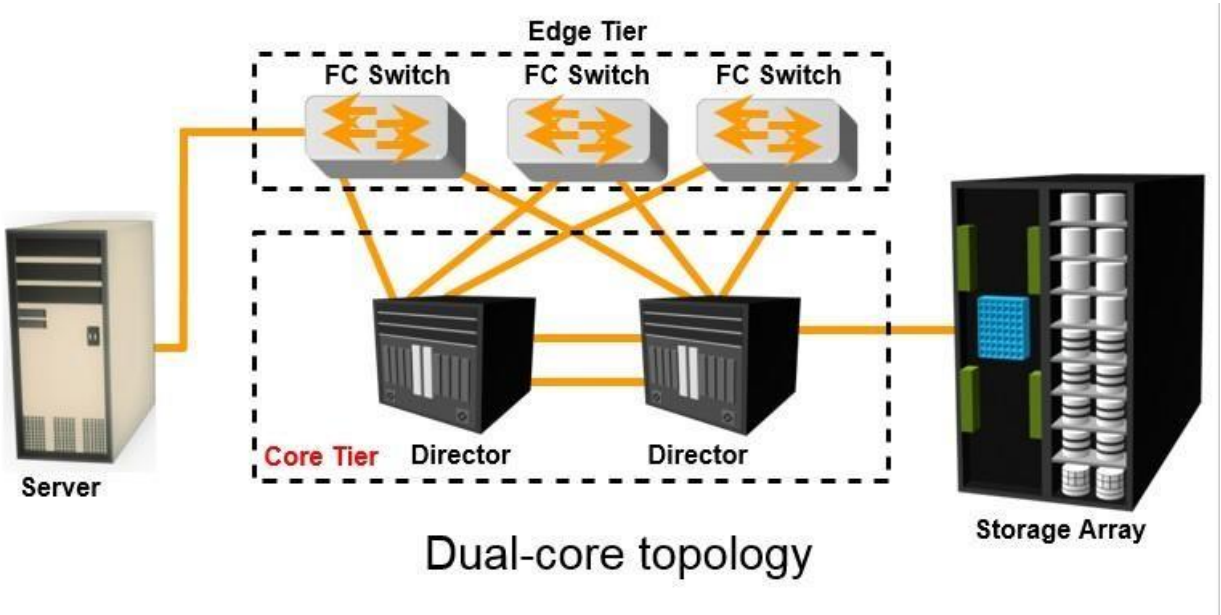


Fig 2.17: multi-core topology

Benefits and Limitations of Core-Edge Fabric

- The core-edge fabric provides one-hop storage access to all storage in the system. Because traffic travels in a deterministic pattern (from the edge to the core), a core-edge provides easier calculation of ISL loading and trafficpatterns.
- Because each tier's switch is used for either storage or hosts, one can easily identify which resources are approaching their capacity, making it easier to develop a set of rules for scaling andapportioning.
- Core-edge fabrics can be scaled to larger environments by linking core switches, adding more core switches, or adding more edgeswitches.
- However, the core-edge fabric may lead to some performance-related problems because scaling a core-edge topology involves increasing the number of ISLs in thefabric.
- As more edge switches are added, the domain count in the fabricincreases.

As the number of cores increases, it is prohibitive to continue to maintain ISLs from each core to each edge switch. When this happens, the fabric design is changed to a **compound or complex core-edge design.**

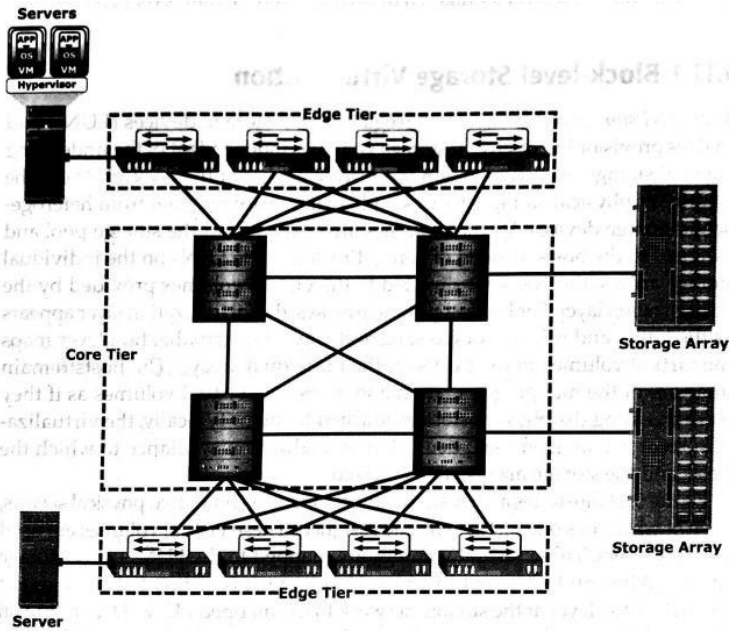


Fig 2.18: Compound core-edge topology

2.9 SAN based virtualization and VSANtechnology

There are two network-based virtualization techniques in a SAN environment:

- block-level storagevirtualization
- virtual SAN(VSAN).

Block-level Storage Virtualization

- *Block-level storage virtualization* aggregates block storage devices (LUNs) and enables provisioning of virtual storage volumes, independent of the underlying physicalstorage.
- A virtualization layer, which exists at the SAN, abstracts the identity of physical storage devices and creates a storage pool from heterogeneous storagedevices.
- Virtual volumes are created from the storage pool and assigned to thehosts.
- Instead of being directed to the LUNs on the individual storage arrays, the hosts are directed to the virtual volumes provided by the virtualizationlayer.
- 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 individualarrays.
- The hosts remain unaware of the mapping operation and access the virtual volumes as if they were accessing the physical storage attached tothem.
- Typically, the virtualization layer is managed via a dedicated virtualization appliance to which the hosts and the storage arrays are connected.
- Fig 2.19 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 storagearrays.
- 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 virtualvolumes.

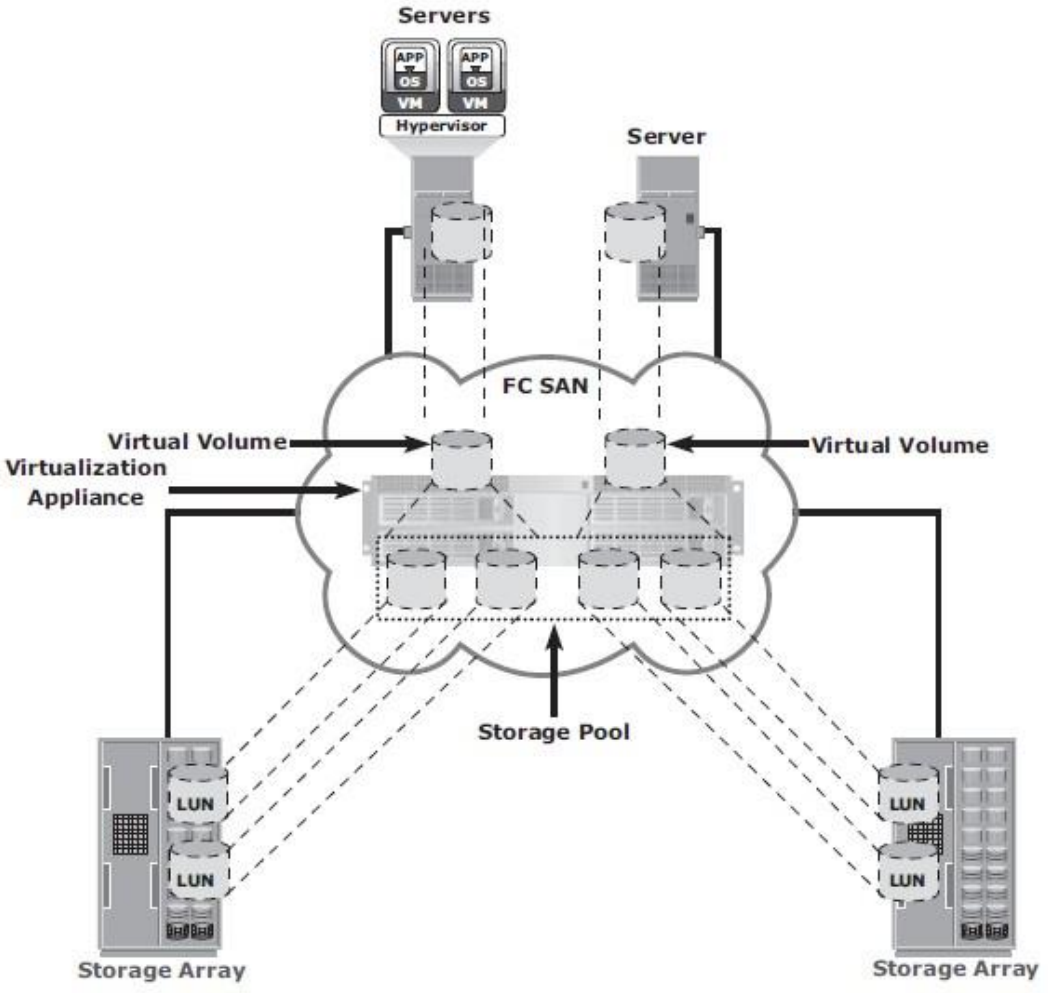


Fig 2.19 Block-level storage virtualization

- Block-level storage virtualization also provides the advantage of nondisruptive datamigration.
- 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 arrayconfiguration.
- In other instances, host CPU cycles were required to migrate data from one array to the other, especially in a multivendorenvironment.
- Withablock-levelvirtualizationasasolution,thevirtualizationlayerhandlestheback-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 virtualizationlayer.
- 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 datacenters.
- 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 (Fig 2.20). This enables the federation of block-storage resources both within and across data centers. The virtual volumes are created from the federated storage resources.

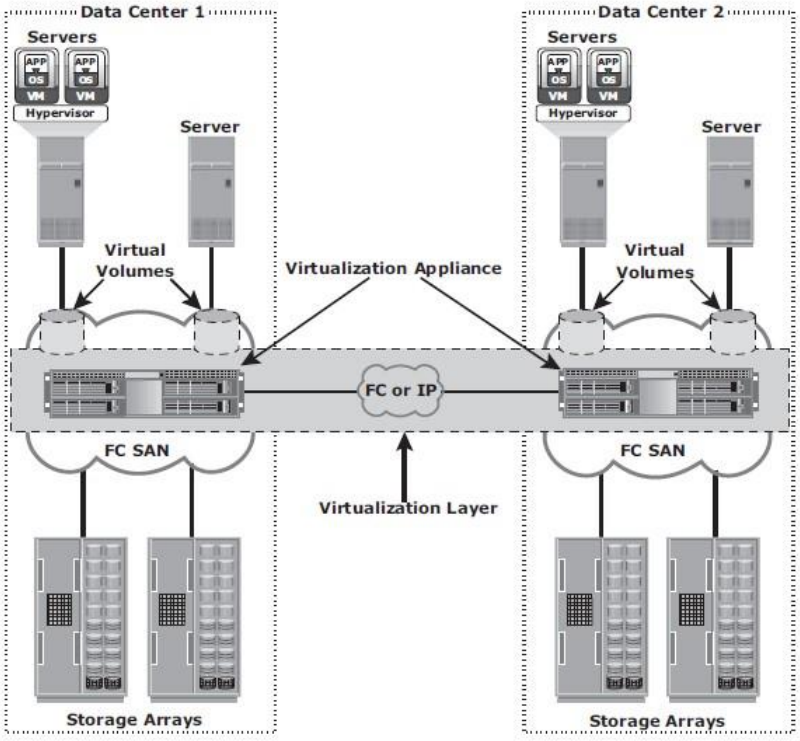


Fig 2.20 Federation of block storage across data centers

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 thefabric.
- 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 physicalSAN.
- Each VSAN acts as an independent fabric with its own set of fabric services, such as name server, andzoning.
- Fabric-related configurations in one VSAN do not affect the traffic inanother.
- VSANs improve SAN security, scalability, availability, andmanageability.
- VSANs facilitate an easy, flexible, and less expensive way to managenetworks.
- Configuring VSANs is easier and quicker compared to building separate physical FC SANs for various nodegroups.
- To regroup nodes, an administrator simply changes the VSAN configurations without moving nodes andrecabling.

IP SAN and FCoE

Traditional SAN environments ***allow block I/O over Fibre Channel***. IP offers easier management and better interoperability. When block I/O is run over IP, the existing network infrastructure can be leveraged, which is more economical than investing in new SAN hardware andsoftware.

Many long-distance, disaster recovery (DR) solutions are already leveraging IP-based networks. In addition, many robust and mature security options are now available for IP networks. With the advent of block storage technology that leverages IP networks (the result is often referred to as IP SAN), organizations can extend the geographical reach of their storage infrastructure. IP SAN technologies can be used in a variety of situations.

Primary protocols that leverage IP as the ***transport mechanism are iSCSI and Fibre Channel over IP (FCIP). iSCSI is the host-based encapsulation of SCSI I/O over IP using an Ethernet NIC card or an iSCSI HBA in the host.***

2.10 iSCSI

- Internet Small Computer Systems Interface (iSCSI) is an IP based protocol that establishes and manages connections between host and storage over IP, as shown in Fig2.21.
- iSCSI encapsulates SCSI commands and data into an IP packet and transports them using TCP/IP.
- iSCSI is widely adopted for connecting servers to storage because it is relatively inexpensive and easy to implement, especially in environments in which an FC SAN does not exist.

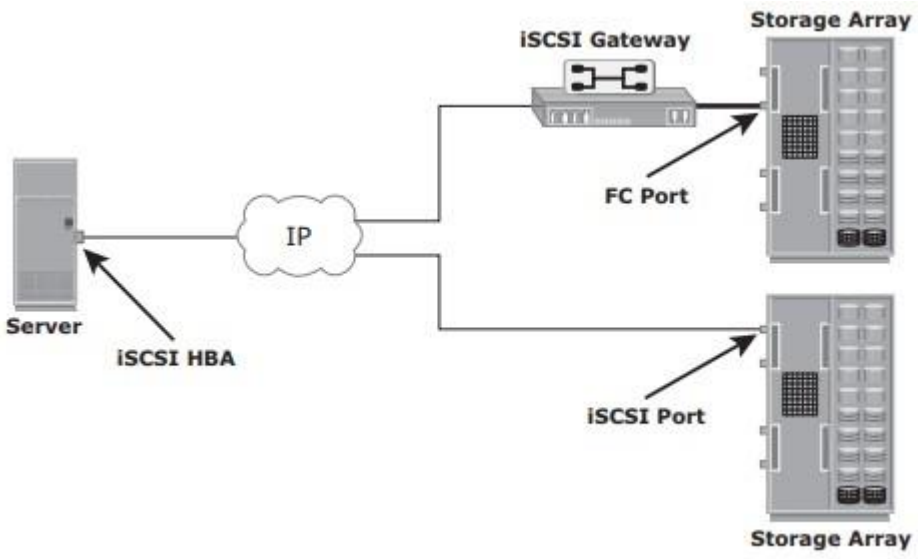


Fig 2.21: iSCSI implementation

2.9.1 Components ofiSCSI

- An initiator (host), target (storage or iSCSI gateway), and an IP-based network are the key iSCSIcomponents.
- If an iSCSI-capable storage array is deployed, then a host with the iSCSI initiator can directly communicate with the storage array over an IPnetwork.
- However, in an implementation that uses an existing FC array for iSCSI communication, an iSCSI gateway isused.

- These devices perform the translation of IP packets to FC frames and vice versa, thereby bridging the connectivity between the IP and FCenvironments.

2.9.2 iSCSI Host Connectivity

The three iSCSI host connectivity options are:

- A standard NIC with software iSCSIinitiator,
- a TCP offload engine (TOE) NIC with software iSCSIinitiator,
- an iSCSIHBA

- The function of the iSCSI initiator is to route the SCSI commands over an IPnetwork.
- A **standard NIC with a software iSCSI** initiator is the simplest and least expensive connectivity option. It is easy to implement because most servers come with at least one, and in many cases two, embedded NICs. It requires only a software initiator for iSCSI functionality. Because NICs provide standard IP function, encapsulation of SCSI into IP packets and decapsulation are carried out by the host CPU. This places additional overhead on the host CPU. If a standard NIC is used in heavy I/O load situations, the host CPU might become a bottleneck. TOE NIC helps reduce thisburden.
- 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 CPUcycles.
- 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 FCHBA.

2.9.3 iSCSI Topologies

- Two topologies of iSCSI implementations are **native andbridged**.
- Native topology does not have FCcomponents.
- The initiators may be either directly attached to targets or connected through the IPnetwork.
- 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 anFC environment.

Native iSCSI Connectivity

- FC components are not required for iSCSI connectivity if an iSCSI-enabled array is deployed.
- In Fig 2.22(a), the array has one or more iSCSI ports configured with an IP address and is connected to a standard Ethernetswitch.
- 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 hostsgenerate.

Bridged iSCSI Connectivity

- A bridged iSCSI implementation includes FC components in itsconfiguration.
- Fig 2.22(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 FCstorage.
- The gateway converts IP packets to FC frames and viceversa.
- The bridge devices contain both FC and Ethernet ports to facilitate the communication between the FC and IPenvironments.

- In a bridged iSCSI implementation, the iSCSI initiator is configured with the gateway's IP address as its targetdestination.
- On the other side, the gateway is configured as an FC initiator to the storagearray.
- **Combining FC and Native iSCSI Connectivity:** The most common topologyis a combination of FC and native iSCSI. Typically, a storage array comes with both FC and iSCSI ports that enable iSCSI and FC connectivity in the same environment, as shown in Fig 2.22(c).

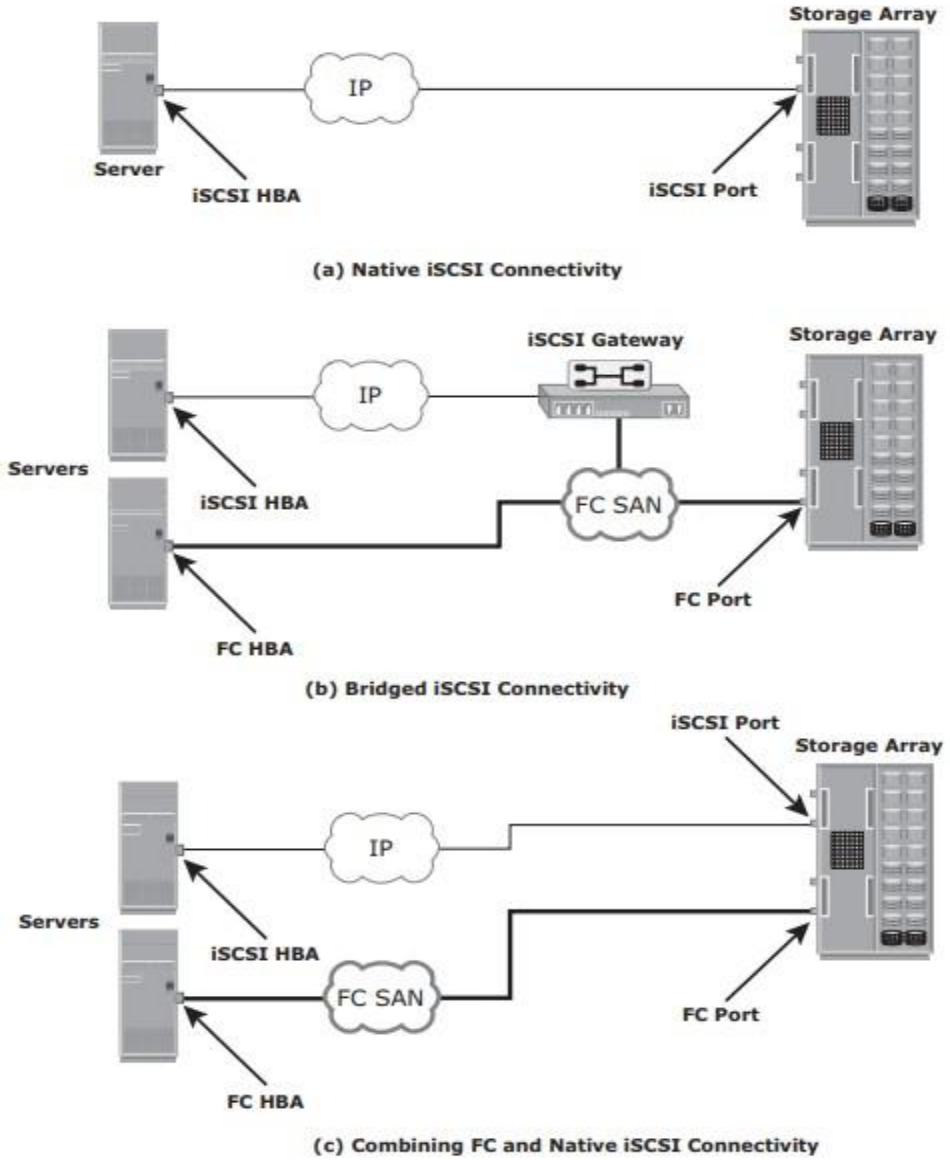


Fig 2.22 : iSCSI Topologies

2.9.4 iSCSI ProtocolStack

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

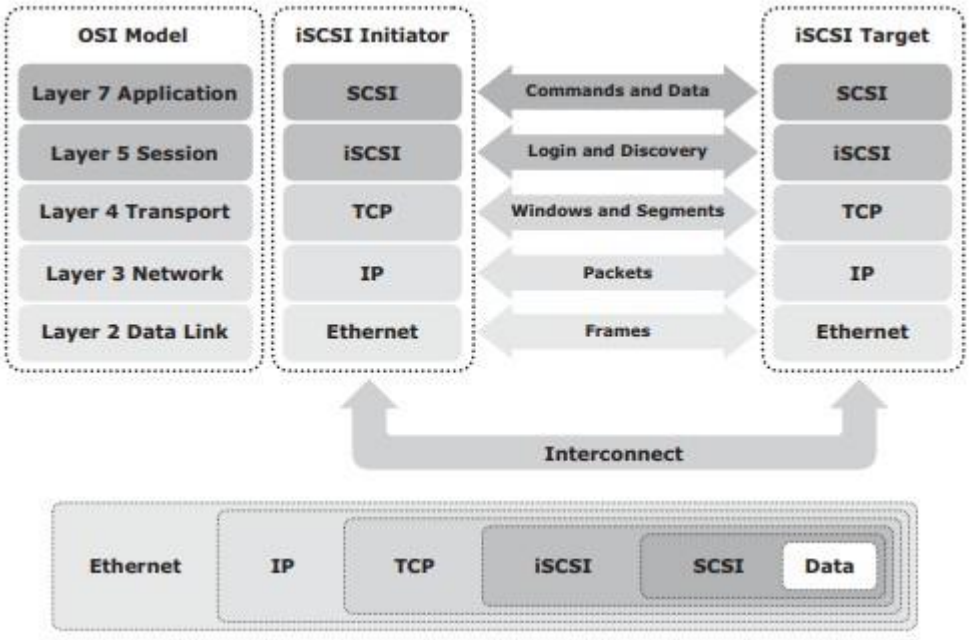


Fig 2.23: iSCSI protocol stack

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

- TCP controls message flow, windowing, error recovery, andretransmission.
- 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.

2.9.5 iSCSI PDU

- A *protocol data unit* (PDU) is the basic "information unit" in the iSCSIenvironment.
- 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 SCSIstatus.
- 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 thetransport.
- A PDU includes the components shown in Fig2.23.
- The IP header provides packet-routing information to move the packet across anetwork.
- 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 EthernetCRC.
- The header and the data digests are optionally used in the PDU to validate integrity and data placement.

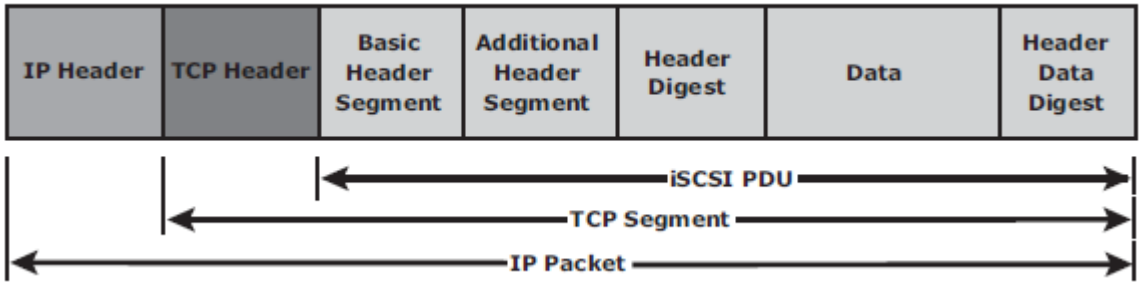


Fig 2.23 iSCSI PDU encapsulated in an IP packet

- The iSCSI header describes how to extract SCSI commands and data for the target. iSCSI adds an optional CRC, known as the *digest*, beyond the TCP checksum and Ethernet CRC to ensure datagram integrity.
- The header and the data digests are optionally used in the PDU to validate integrity, data placement, and correct operation As shown in Figure 8-7.

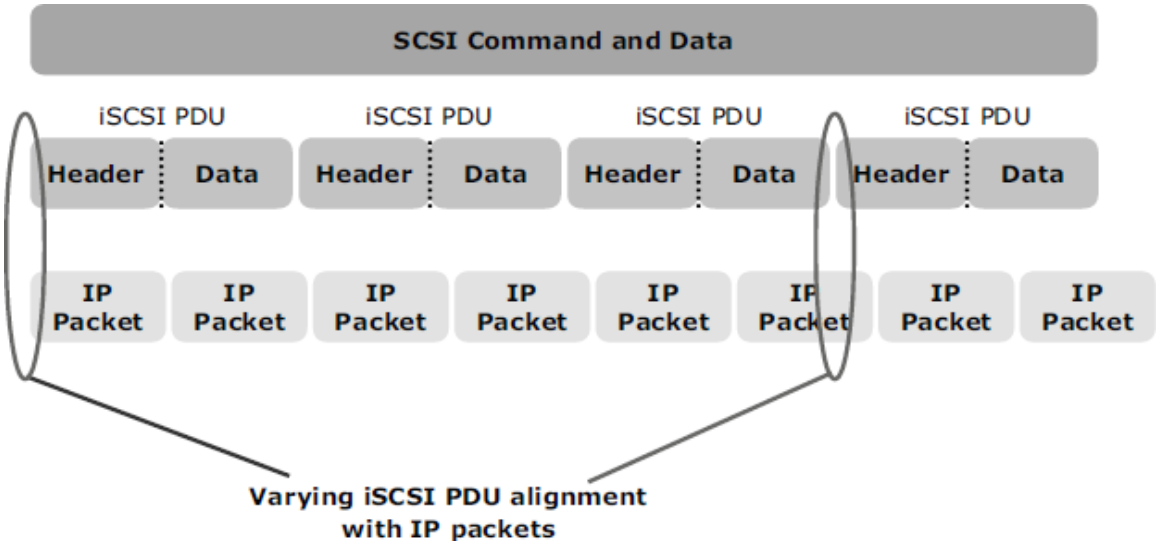


Figure 8-7: Alignment of iSCSI PDUs with IP packets

Each iSCSI PDU does not correspond in a 1:1 relationship with an IP packest.

Depending on its size, an iSCSI PDU can span an IP packet or even coexist with another PDU in the same packet.

Therefore, each IP packet and Ethernet frame can be used more efficiently because fewer packets and frames are required to transmit the SCSI information.

2.9.6 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 asession.
- This discovery can take place in twoways:
  - **SendTargetsdiscovery**
  - **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 thehost.
- iSNS (Fig 2.24) enables automatic discovery of iSCSI devices on an IP network. The initiators and targets can be configured to automatically register themselves with the iSNS server. Whenever an initiator wants to now the targets that it can access, it can query the iSNS server for a list of availabletargets.
- The discovery can also take place by using service location protocol (SLP). However, this is less commonly used than SendTargets discovery andiSNS.

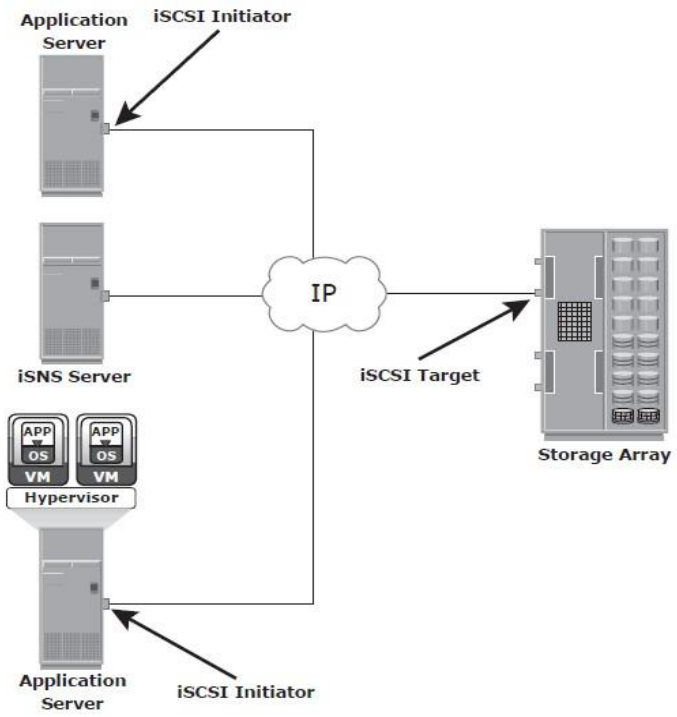


Fig 2.24 : Discovery using iSNS

2.9.7 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 facilitatecommunication.
- 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 thedevices.
- Following are two types of iSCSI names commonlyused:
    - **iSCSI Qualified Name(IQN):**
    - **Extended Unique Identifier(EUI)**
- **iSCSI Qualified Name (IQN):** An organization must own a registered domain name to generate iSCSI Qualified Names. This domain name does not need to be active or resolve to an address. It just needs to be reserved to prevent other organizations from using the same domain name to generate iSCSI names. A date is included in the name to avoid potential conflicts caused by the transfer of domainnames.

  An example of an IQN is iqn.2008-02.com.example:*optional_string.* The *optional_string* provides a serial number, an asset number, or any otherdevice identifiers.
- **Extended Unique 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 hexadecimalname, suchaseui.0300732A32598D26.
- In either format, the allowed special characters are dots, dashes, and blankspaces.

2.9.8 iSCSI Session

- An iSCSI session is established between an initiator and a target, as shown in Fig2.25.
- 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 thefollowing:
    - The discovery of the available targets by the initiators and the location of a specific target on anetwork
    - The normal operation of iSCSI (transferring data between initiators andtargets)

- 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 targetport.
- During the login phase, the initiator and the target authenticate each other and negotiate on variousparameters.
- 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 thetarget.
- The final phase of the iSCSI session is the connection termination phase, which is referred to as the logoutprocedure.
- The initiator is responsible for commencing the logout procedure; however, the target may also prompt termination by sending an iSCSI message, indicating the occurrence of an internal error condition.
- After the logout request is sent from the initiator and accepted by the target, no further request and response can be sent on that connection.

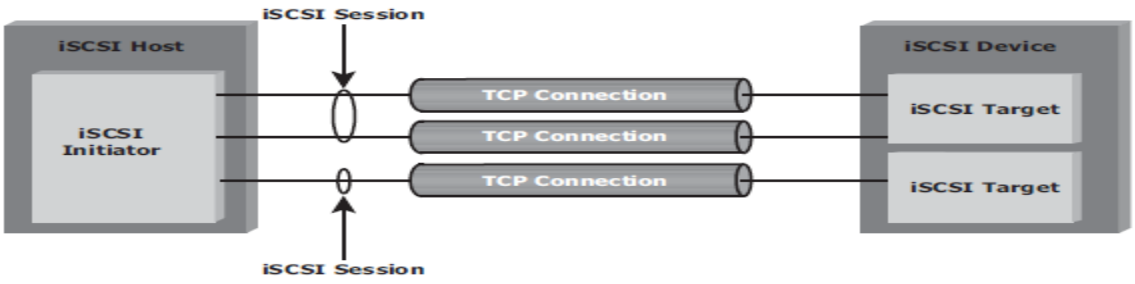


Fig 2.25 iSCSI session

iSCSICommandSequencing

- The iSCSI communication between the initiators and targets is based on the request-response commandsequences.A command sequence may generate multiplePDUs.
- A ***command sequence number*** **(CmdSN**) within an iSCSI session is used for numbering all initiator-to-target command PDUs belonging to thesession.
- 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 thesession.
- Command sequencing begins with the first login command, and the CmdSN is incremented by one for each subsequentcommand.
- The iSCSI target layer is responsible for delivering the commands to the SCSI layer in the order of theirCmdSN.
- Similar to command numbering, a ***status sequence number*** **(StatSN)** is used to sequentially number status responses, as shown in Fig2.26.
- These unique numbers are established at the level of the TCPconnection.
- A target sends ***request-to-transfer*** **(R2T)** PDUs to the initiator when it is ready to acceptdata.
- A *data sequence number* (DataSN) is used to ensure in-order delivery of data within the same command.
- The DataSN and R2TSN are used to sequence data PDUs and R2Ts,respectively.

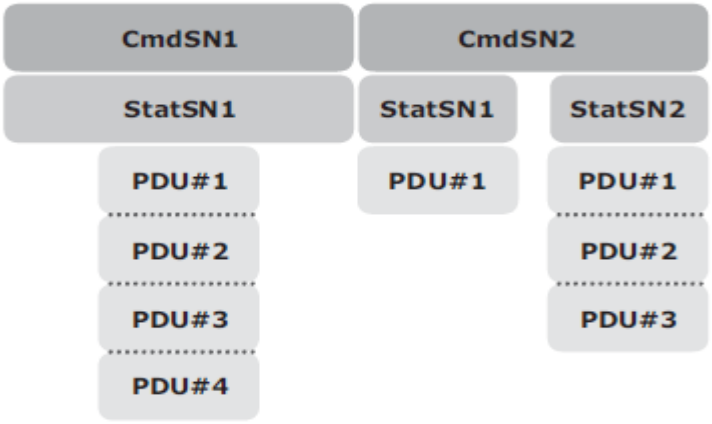


Fig 2.26 Command and status sequence number

2.11 FCIP (Fibre channel overIP)

- FCIP is a IP-based protocol that is used to connect distributed FC-SANislands.
- Creates virtual FC links over existing IP network that is used to transport FC data between different FCSANS.
- It encapsulates FC frames into IPpacket.
- It provides disaster recoverysolution.

2.10.1 FCIP ProtocolStack

- The FCIP protocol stack is shown in Fig 2.27. Applications generate SCSI commands and data, which are processed by various layers of the protocolstack.
- The upper layer protocol SCSI includes the SCSI driver program that executes the read-and-writecommands.
- Below the SCSI layer is the Fibre Channel Protocol (FCP) layer, which is simply a Fibre Channel frame whose payload isSCSI.
- 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 theIP.

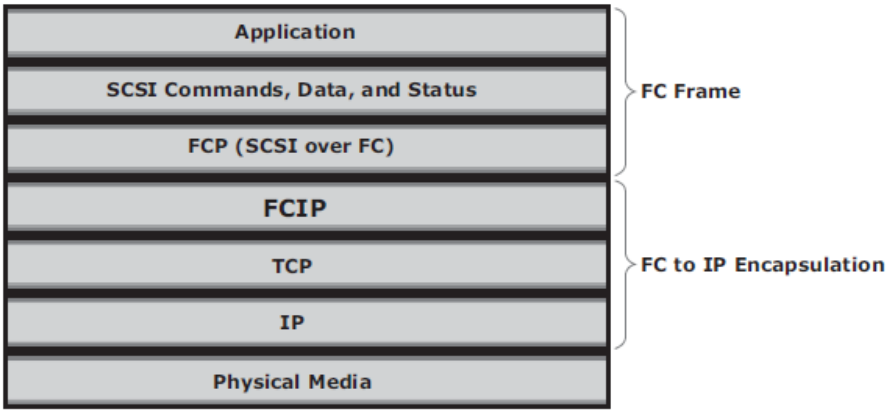


Fig 2.27 FCIP protocol stack

- The FCIP layer encapsulates the Fibre Channel frames onto the IP payload and passes them to the TCP layer (see Fig 2.28). TCP and IP are used for transporting the encapsulated information across Ethernet, wireless, or other media that support the TCP/IPtraffic.
- 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 IPpacket.
- 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 thedevice.

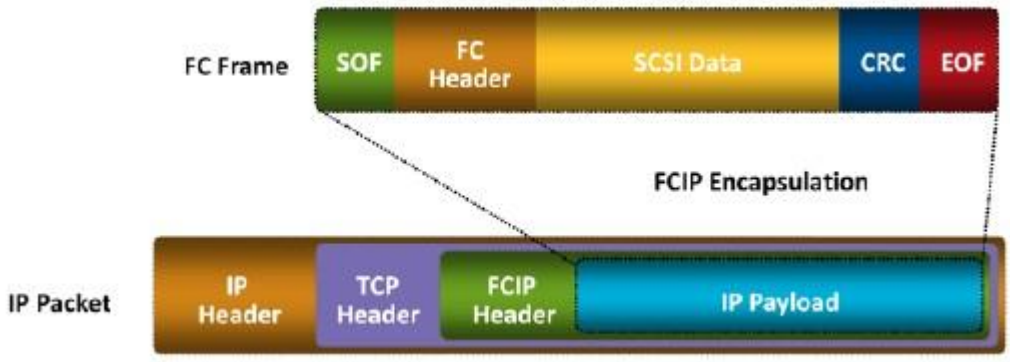


Fig 2.28 FCIP encapsulation

2.10.2 FCIPTopology

- In an FCIP environment, an FCIP gateway is connected to each fabric via a standard FC connection (Fig2.29).
- The FCIP gateway at one end of the IP network encapsulates the FC frames into IPpackets.
- The gateway at the other end removes the IP wrapper and sends the FC data to the layer 2fabric.
- The fabric treats these gateways as layer 2 fabricswitches.
- An IP address is assigned to the port on the gateway, which is connected to an IP network. After theIPconnectivityisestablished,thenodesinthetwoindependentfabricscancommunicate

with each other.

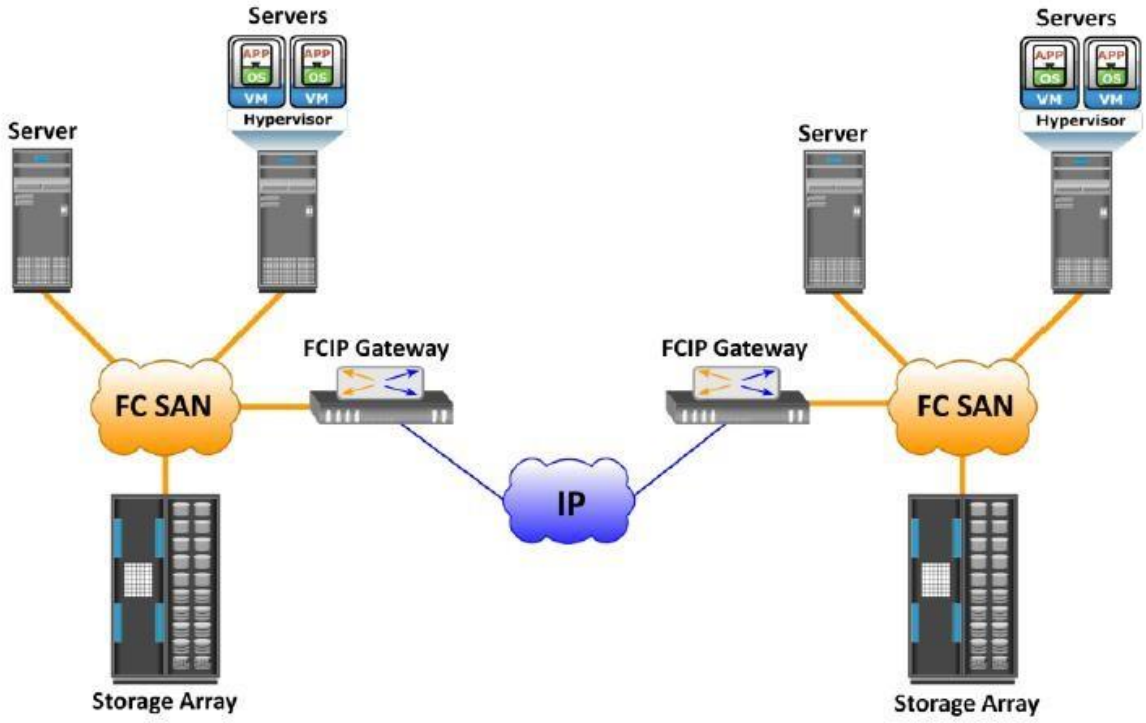


Fig 2.29 FCIP topology

FCIP Performance andSecurity

- Performance, reliability, and security should always be taken into consideration when implementing storage solutions.
- The implementation of FCIP is also subject to the same consideration. From the perspective of performance, multiple paths to multiple FCIP gateways from different switches in the layer 2 fabric eliminates single points of failure and provides increased bandwidth. In a scenario of extended distance, the IP network may be a bottleneck if sufficient bandwidth is notavailable.
- In addition, because FCIP creates a unified fabric, disruption in the underlying IP network can cause instabilities in the SAN environment. These include a segmented fabric, excessive RSCNs, and hosttimeouts.
- The vendors of FC switches have recognized some of the drawbacks related to FCIP and have implemented features to provide additional stability, such as the capability to segregate 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

2.12 FCoE (Fibre Channel over Ethernet)

- Data centers typically have multiple networks to handle various types of I/O traffic — for example, an Ethernet network for TCP/IP communication and an FC network for FC communication.
- TCP/IP is typically used for client-server communication, data backup, infrastructure management communication, and so on.
- FC is typically used for moving block-level data between storage andservers.
- 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 datacenters.
- The need for two different kinds of physical network infrastructure increases the overall cost and complexity of data centeroperation.
- Fibre Channel over Ethernet (FCoE) protocol provides consolidation of LAN and SAN traffic over a single physical interfaceinfrastructure.
- 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 overEthernet.

I/O Consolidation UsingFCoE

- The key benefit of FCoE is I/O consolidation.
- Figure 6-12 represents the infra- structure 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.

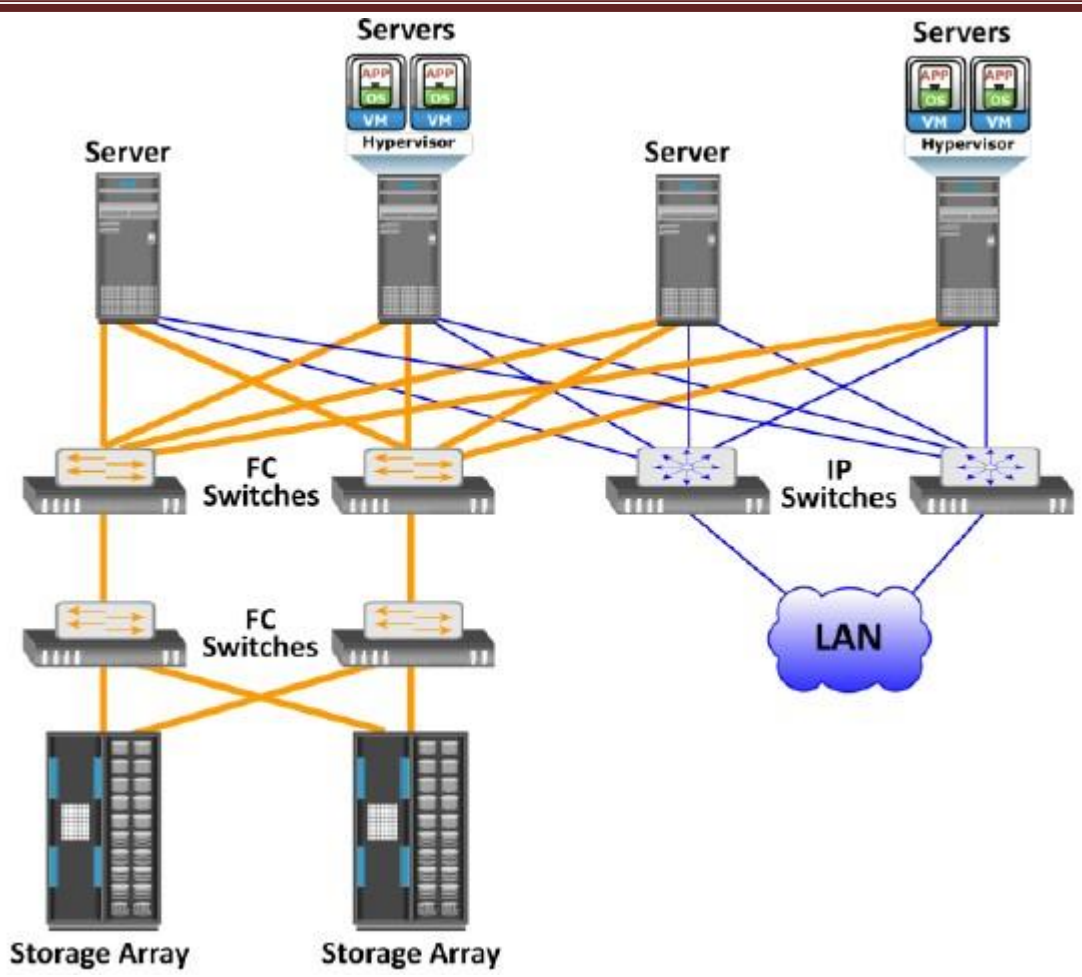


Fig 2.30 Infrastructure Before using FCoE

- Figure 6-13 shows the I/O consolidation with FCoE using FCoE switches and Converged Network Adapters (CNAs).
- A CNA (discussed in the section "ConvergedNetworkAdapter")replacesbothHBAsandNICsintheserverand consolidatesboththeIPandFCtraffic.
- Thisreducestherequirementofmultiple network adapters at the server to connect to different networks. Overall, this reduces the requirement of adapters, cables, and switches.
- This also consider- ably reduces the cost and managementoverhead.

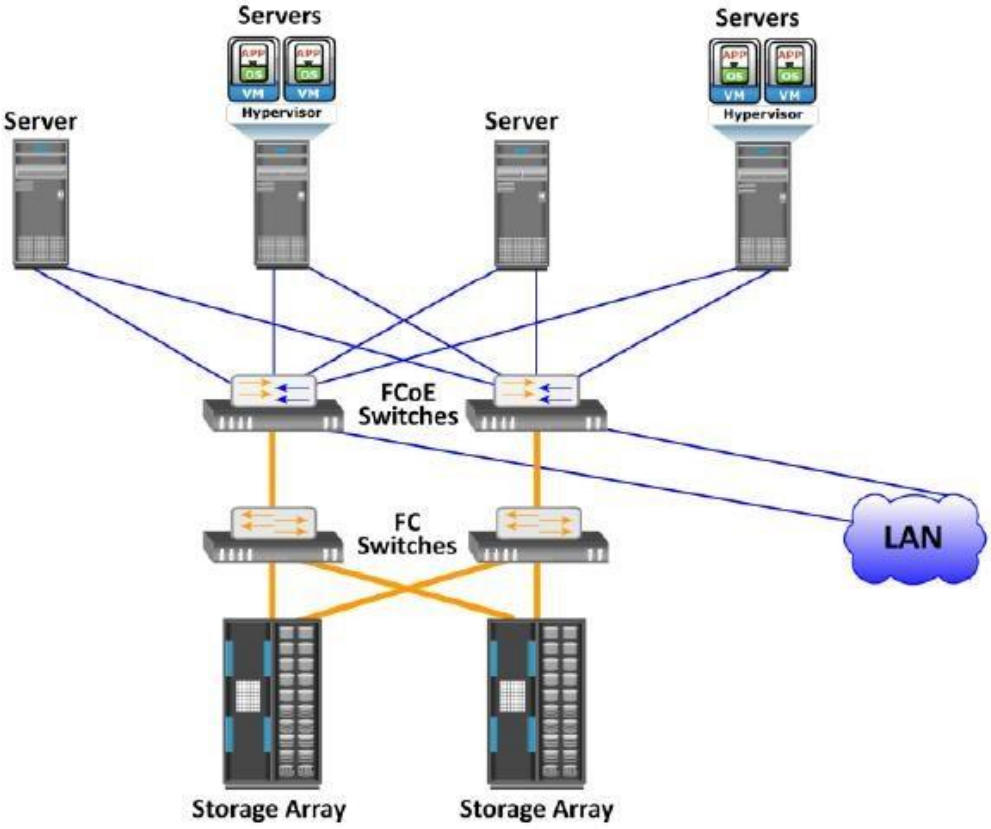


Fig 2.30 Infrastructure After using FCoE

2.11.1 Components ofFCOE

The key components of FCOE are :

- Converged NetworkAdaptors(CNA)
- Cables
- FCOESwitches

Converged Network Adaptors(CNA)

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

- As shown in Fig 2.31, a CNA contains separate modules for 10 Gigabit Ethernet, Fibre Channel, and FCoE Application Specific Integrated Circuits (ASICs). The FCoE ASIC encapsulates FC frames into Ethernet frames. One end of this ASIC is connected to 10GbE and FC ASICs for server connectivity, while the other end provides a 10GbE interface to connect to an FCoE switch.

Fig 2.31 Converged Network Adapter

Cables

- There are two options available for FCoEcabling:
    1. Copper basedTwinax
    2. standard fiber opticalcables.
- 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 coppercables.

FCoE Switches

- An FCoE switch has both **Ethernet switch** and **Fibre Channel switch**functionalities.
- As shown in Fig 2.32, FCoE switch consists of:

  1. *Fibre Channel Forwarder(FCF),*
  2. *EthernetBridge,*
  3. set of Ethernetports
  4. optional FCports

- 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 FCframes.
- 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 thedestination.
  - 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 FCports.
  - If the Ethertype is not FCoE, the switch handles the traffic as usual Ethernet traffic and forwards it over the Ethernetports.

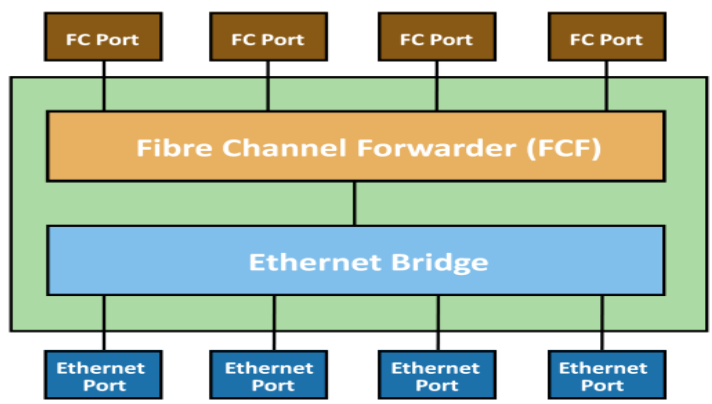


Fig 2.32 FCoE switch generic architecture

FCoE FrameStructure

- AnFCoEframeisanEthernetframethatcontainsanFCoEProtocolDataUnit.
- 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 sourceMACaddress.The32-bitIEEE802.1Qtagsupportsthecreationofmultiple 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.Thenext100-bitsarereservedandarefollowedbythe8-bitStartofFrame andthentheactualFCframe.
- The8-bitEndofFramedelimiterisfollowedby24 reservedbits.Theframeendswiththefinal32-bitsdedicatedtotheFrameCheck Sequence (FCS) function that provides error detection for the Ethernetframe.

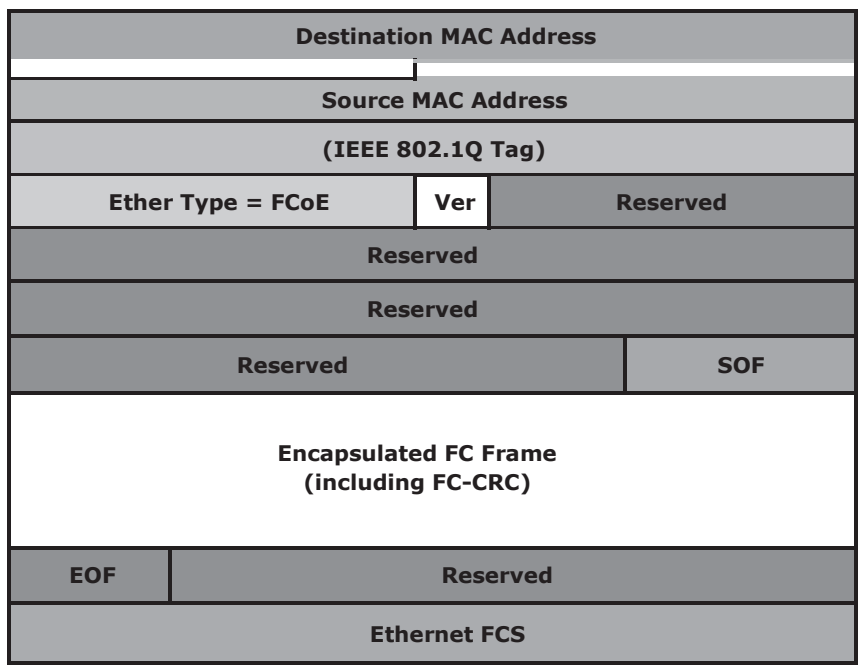


Figure 6-16: FCoE frame structure

- 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 theFCoEframeandtransportedtoFCSANseamlessly.
- Thiscapabilityenables FCoE to integrate with the existing FC SANs without the need for agateway.
- 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 Ethernetframehasadefaultpayloadcapacityof1,500bytes.Tomaintaingood performance, FCoE must use jumbo frames to prevent a Fibre Channel frame from being split into two Ethernet frames.
- The next chapter discusses jumbo framesindetail.FCoErequiresConvergedEnhancedEthernet,whichprovides lossless Ethernet and jumbo framesupport.

FCoE Frame Mapping

- TheencapsulationoftheFibreChannelframeoccursthroughthemappingofthe FCframesontoEthernet,asshowninFigure6-17.
- FibreChannelandtraditional networkshavestacksoflayerswhereeachlayerinthestackrepresentsasetof functionalities.
- TheFCstackconsistsoffivelayers:FC-0throughFC-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 Ethernetlayer.

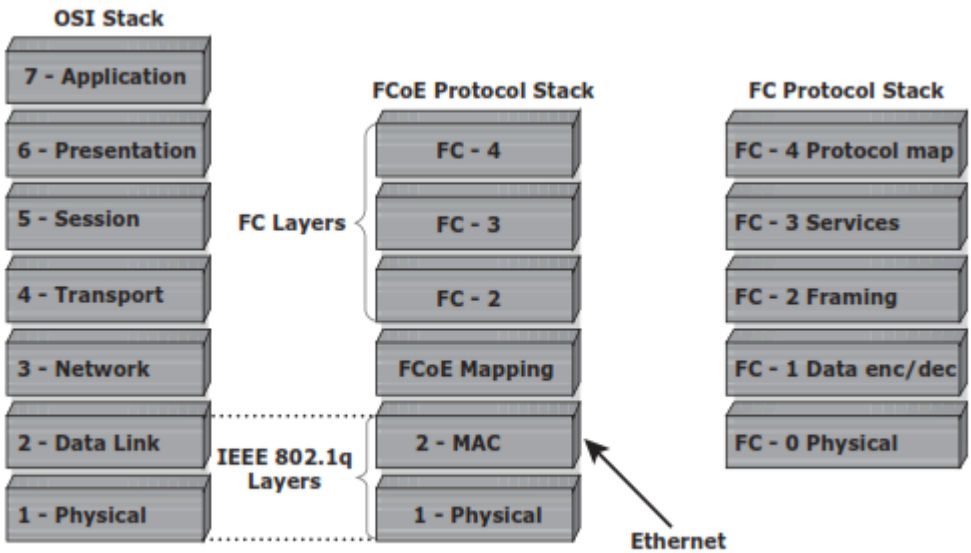


Figure 6-17: FCoE frame mapping

FCoE EnablingTechnologies

- Conventional Ethernet is lossy in nature, which means that frames might be dropped or lost during transmission.
- *Converged Enhanced Ethernet* (CEE), or losslessEthernet,providesanewspecificationtotheexistingEthernetstandard thateliminatesthelossynatureofEthernet.
- Thismakes10GbEthernetaviable 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 theyare:
    - Priority-based flow control
    - Enhanced transmission selection

      Congestion Notification
    - Data center bridging exchange protocol

Priority-Based Flow Control (PFC)

- TraditionalFCmanagescongestionthroughtheuseofalink-level,credit-based flow control that guarantees no loss of frames.
- Typical Ethernet, coupled with TCP/IP,usesapacketdropflowcontrolmechanism.
- Thepacketdropflowcontrol is not lossless. This challenge is eliminated by using an IEEE 802.3x Ethernet PAUSEcontrolframetocreatealosslessEthernet.
- AreceivercansendaPAUSE request to a sender when the receiver's buffer is filling up. Upon receiving a PAUSE frame, the sender stops transmitting frames, which guarantees no loss of frames.
- The downside of using the Ethernet PAUSE frame is that it operates on the entire link, which might be carrying multiple trafficflows.
- PFCprovidesalinklevelflowcontrolmechanism.PFCcreateseightseparate virtuallinksonasinglephysicallinkandallowsanyoftheselinkstobepaused andrestartedindependently.
- PFCenablesthepausemechanismbasedonuser prioritiesorclassesofservice.Enablingthepausebasedonpriorityallowscre-ating lossless links for traffic, such as FCoE traffic.
- This PAUSE mechanism is typicallyimplementedforFCoEwhileregularTCP/IPtrafficcontinuestodrop frames.
- Figure6-18illustrateshowaphysicalEthernetlinkisdividedintoeight virtuallinksandallowsaPAUSEforasinglevirtuallinkwithoutaffectingthe traffic for theothers.

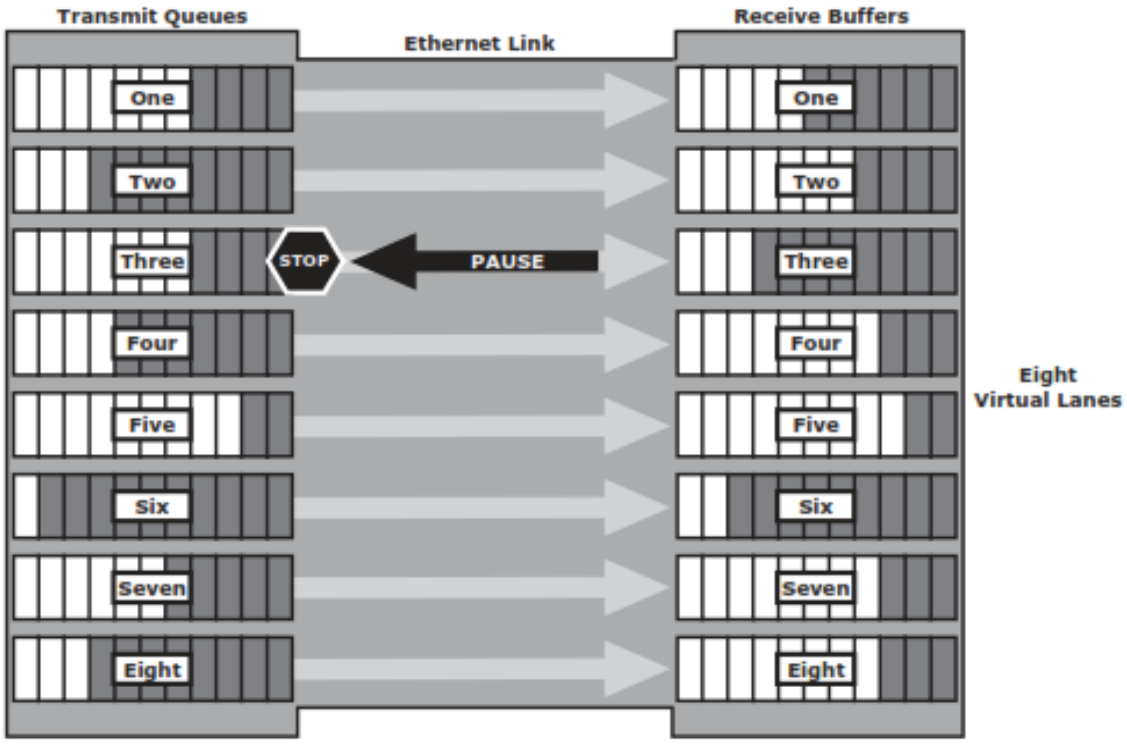


Figure 6-18: Priority-based flow control

Enhanced Transmission Selection (ETS)

- Enhancedtransmissionselectionprovidesacommonmanagementframework fortheassignmentofbandwidthtodifferenttrafficclasses,suchasLAN,SAN, andInterProcessCommunication(IPC).
- Whenaparticularclassoftrafficdoes not use its allocated bandwidth, ETS enables other traffic classes to use the availablebandwidth.

Congestion Notification (CN)

- Congestion notification provides end-to-end congestion management for protocols, such as FCoE, that do not have built-in congestion control mecha- nisms.
- 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.
- TheprocessofcongestionnotificationanditsmanagementisshowninFigure 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 datatransfer.

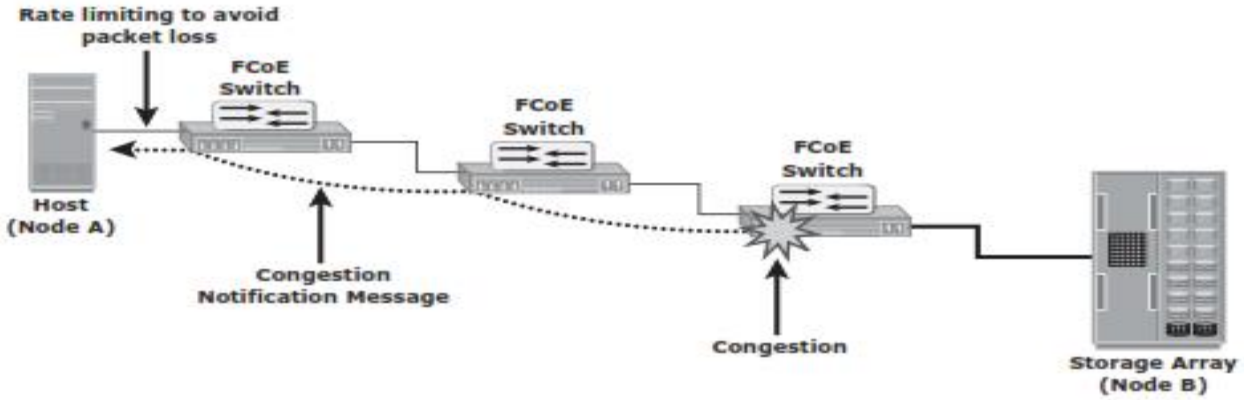


Figure 6-19: Congestion Notification

Data Center Bridging Exchange Protocol (DCBX)

- DCBX protocol is a discovery and capability exchange protocol, which helps Converged Enhanced Ethernet devices to convey and configure their features withtheotherCEEdevicesinthenetwork.
- DCBXisusedtonegotiatecapabilitiesbetween 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.

NETWORK ATTACHED STORAGE(NAS)

- File sharing, as the nameimplies,enables users to share files withotherusers.
- Traditionalmethodsoffilesharinginvolvecopyingfilestoportablemediasuchasfloppy diskette, CD, DVD, or USB drivesanddeliveringthemtootherusers withwhomitisbeing shared.
- However, this approach is not suitable in an enterprise environment in which a large number of users at different locations need access to common files.
- Network-based file sharing provides the flexibility to share files over long distances among a large number of users. File servers use client- server technology to enable file sharing over a network.
- To address the tremendous growth of file data in enterprise environments, organizations have been deploying large numbers of fileservers.
- These servers are either connected to direct-attached storage (DAS) or storage area network (SAN)-attached storage.

- This has resulted in the proliferation of islands of over-utilized and under-utilized file servers and storage. Inaddition, such environments have poor scalability, higher management cost, and greater complexity.
- *Network-attached storage* (NAS) emerged as a solution to these challenges.

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.

File Sharing Environment

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

What is NAS?

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

2.12.1 Components ofNAS

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

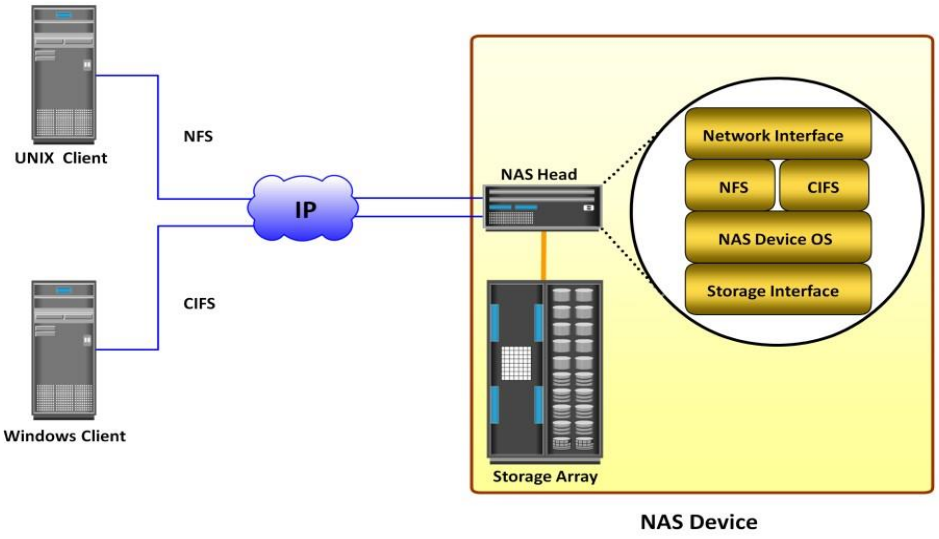


Fig 2.33 Components of NAS

2.12.2 NAS I/OOperation

- NAS provides *file-level data access* to its clients. File I/O is a high-level request that specifies the file to beaccessed.
- Eg: a client may request a file by specifying its name, location, or other attributes. The NAS operating system keeps track of the location of files on the disk volume and converts client file I/O into block-level I/O to retrievedata.
- The process of handling I/Os in a NAS environment is asfollows:
    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 thenetwork.
    2. TheNASdeviceconvertstheI/Orequestintoanappropriatephysicalstoragerequest, 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 protocolresponse.
    4. The NAS device packages this response into TCP/IP again and forwards it to the client through thenetwork.
- Fig 2.34 illustrates the NAS I/O operation

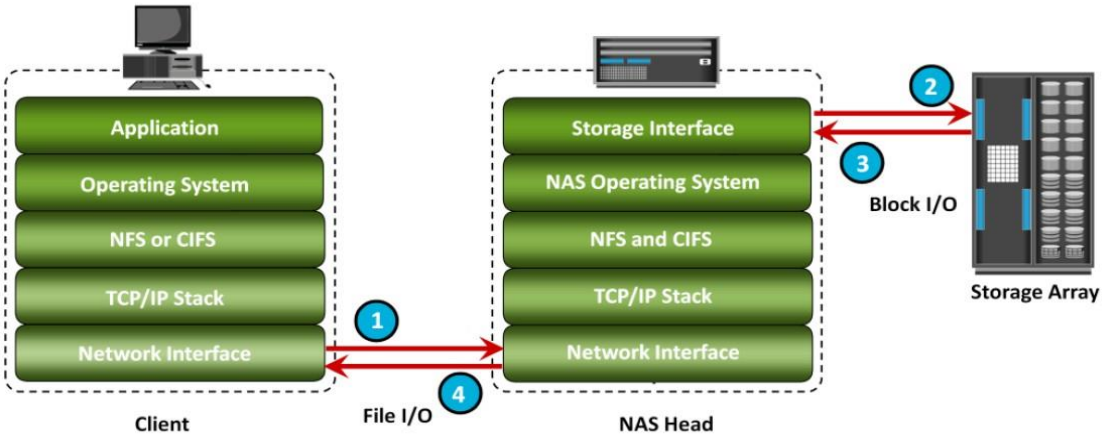


Fig 2.34 NAS I/O Operation

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.

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.

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.

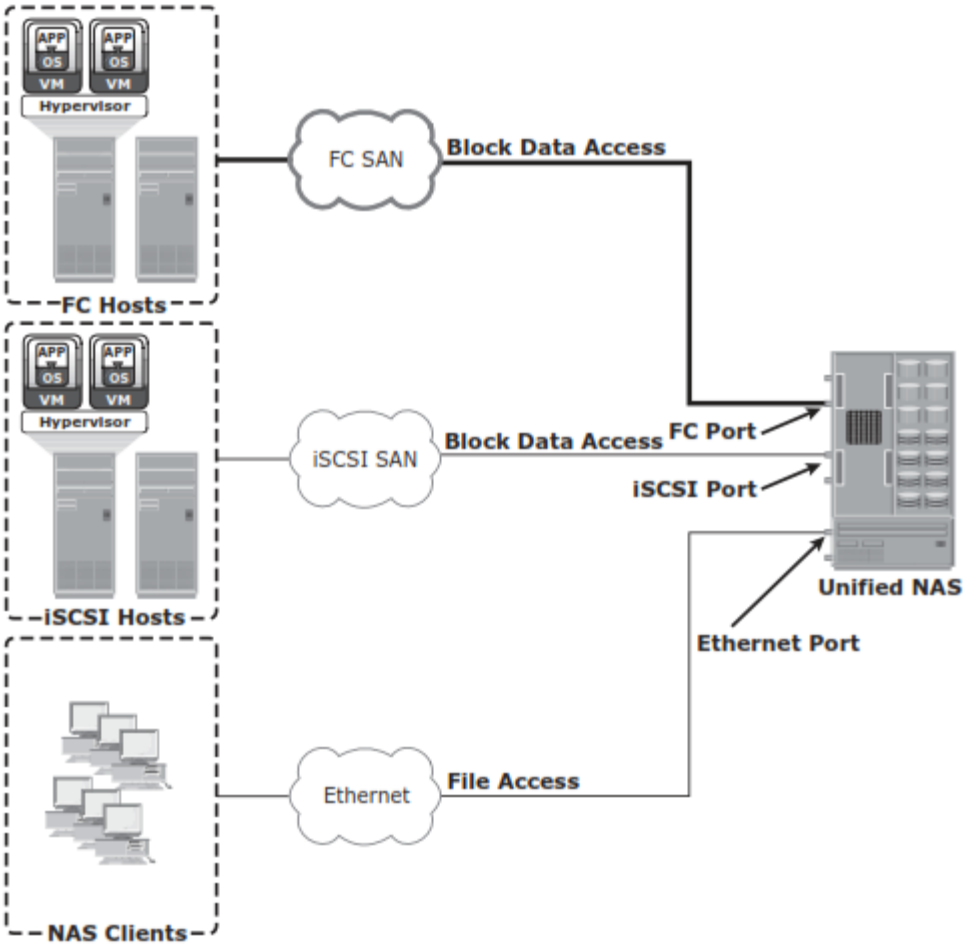


Figure 7-5: Unified NAS connectivity

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 func- tions 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.

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

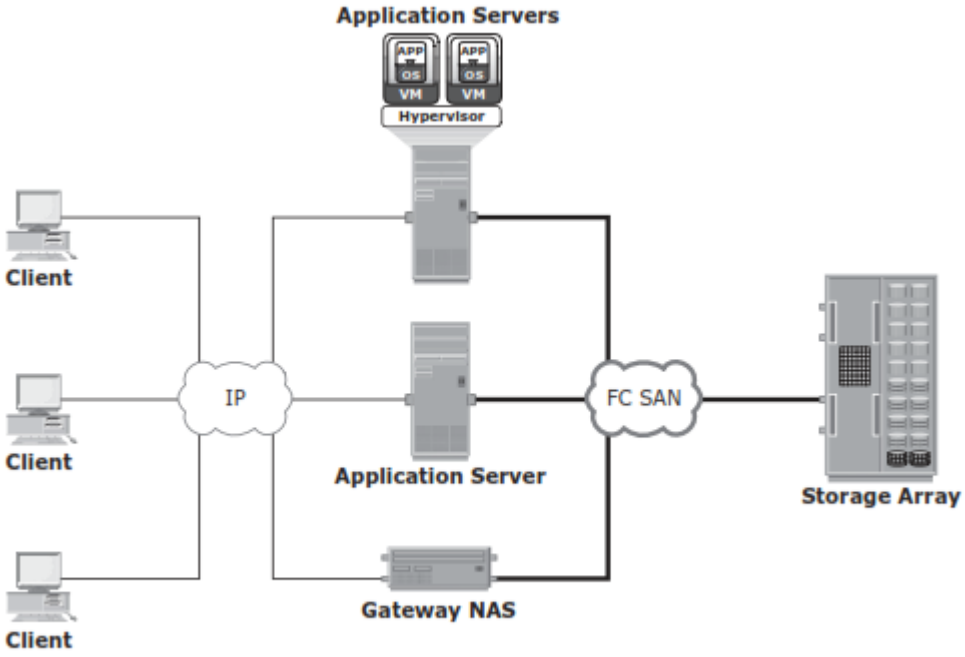


Figure 7-6: Gateway NAS connectivity

- 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.

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.

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 con- nections 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.

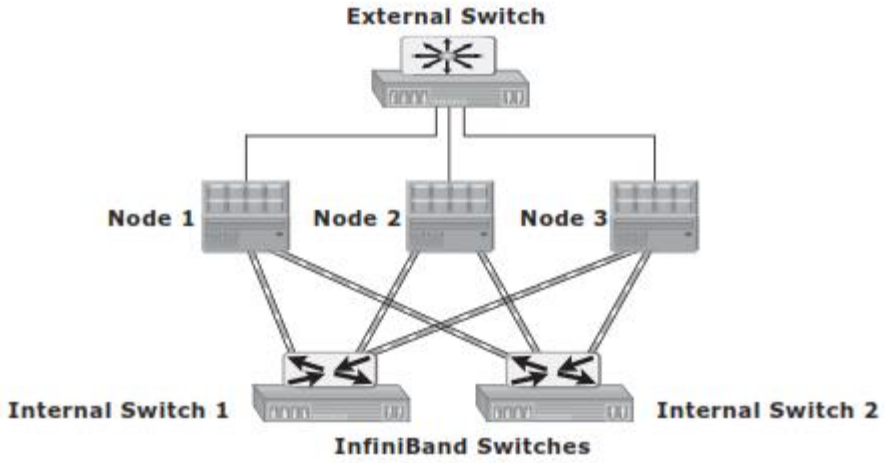


Figure 7-7: Scale-out NAS with dual internal and single external networks

2.12.3 NAS File Sharing Protocols

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

Network File System (NFS)

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

Common Internet File System (CIFS)

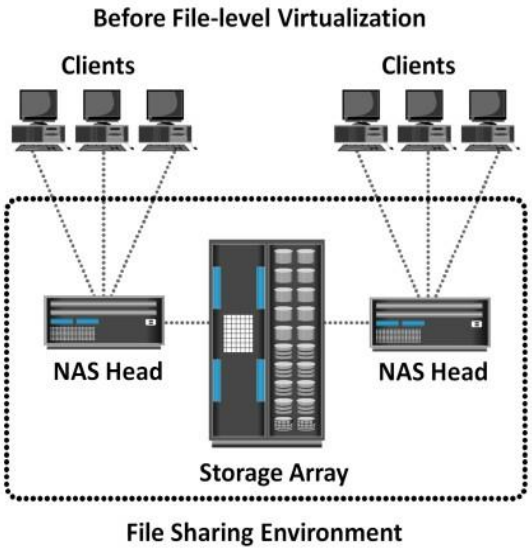
- CIFS is a *client-server application* protocol
- It enables clients to access files and services on remote computers over**TCP/IP**.
- It is a public, or open, variation of **Server Message Block (SMB)**protocol.
- It provides following features to ensure dataintegrity:
    - It uses file and record locking to prevent users from overwriting the work of another user on a file or arecord.
    - It supports fault tolerance and can automatically restore connections and reopen files that were open prior to an interruption. This feature depends on whether an application is written to take advantage ofthis.
    - 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-namingscheme:
- Eg: \\server\share or\\servername.domain.suffix\share

2.13 File-level Virtualization

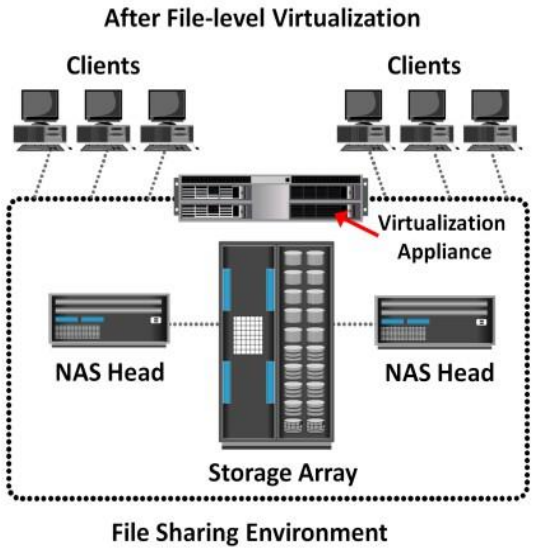
- File-level virtualization, implemented in NAS or the file server environment, provides a simple, non disruptive file-mobilitysolution.
- It eliminates the dependencies between data accessed at the file level and the location where the files are physicallystored.
- It creates a logical pool of storage,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

level virtualization enables the movement of files across NAS devices, even if the files are being accessed.

Before and After File-level Virtualization



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



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

Object-Based Storage & Unified Storage Platform

- **Object-based storage** is a way to store file data in the form of objects based on its *content and other attributes* rather than the name andlocation.
- Recent studies have shown that more than 90 percent of data generated isunstructured.
- Traditional NAS, which is the dominant solution for storing unstructured data, has become inefficient.
- This demands a smarter approach to manage unstructured data based on its content rather than metadata about its name, location, and soon.

- Due to varied application requirements, organizations have been deploying storage area networks (SANs), NAS, and object-based storage devices (OSDs) in their datacenters.
- Deploying these disparate storage solutions adds management complexity, cost and environmentaloverhead.
- An ideal solution is to have an integrated storage solution that supports block, file, and object access.
- **Unified storage** has emerged as a solution that consolidates *block, file,* and *object-based access* within one unifiedplatform.
- It supports multiple protocols for data access and can be managed using a single management interface.

2.14 Object-Based Storage Devices (OSD)

- An OSD is a device that *organizes and stores* unstructured data, such as movies, office documents, and graphics, asobjects.
- **Object-based storage** provides a scalable, self-managed, protected, and shared storage option. OSD stores data in the form ofobjects.
- 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 Fig2.35).

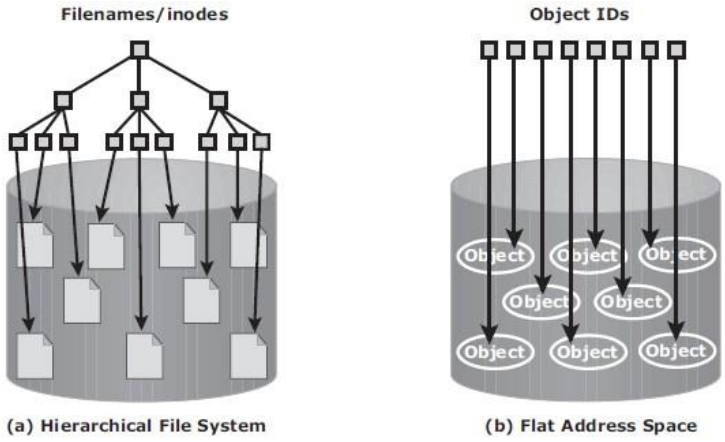


Fig 2.35 Hierarchical file system versus flat address space

- 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 Fig2.36.
- Each object stored in the system is identified by a unique ID called the **objectID**.
- The object ID is generated using specialized algorithms such as hash function on the data and guarantees that every object is uniquelyidentified.

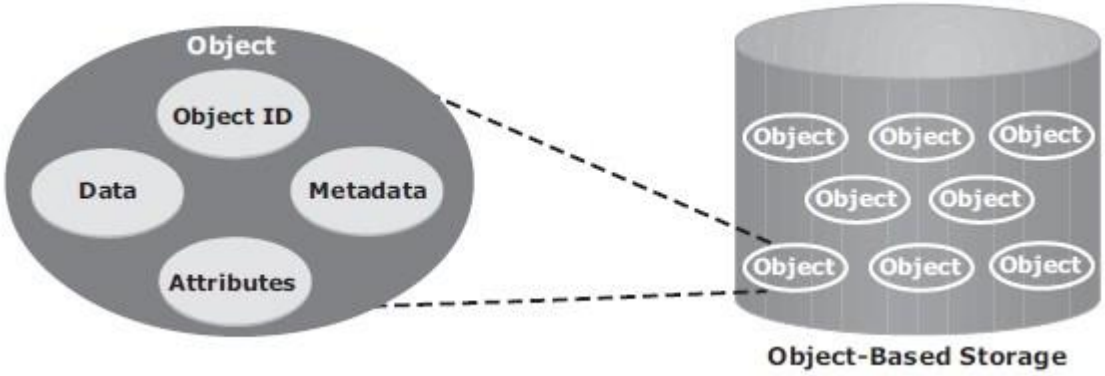


Fig 2.36 ObjectStructure

2.14.1 Object-Based StorageArchitecture

- An I/O in the traditional block access method passes through various layers in the I/Opath.
- The I/O generated by an application passes through the file system, the channel, or network and reaches the diskdrive.
- 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 diskdrive.
- Fig 2.37 (a) illustrates the block-level access.
- The file system has two components: *user component* and *storagecomponent*.

    1. The user component of the file system performs functions such as hierarchy management, naming, and user accesscontrol.
    2. The storage component maps the files to the physical location on the diskdrive.

- 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 storagedevice.
- The storage device has the OSD storage component responsible for managing the access to the object on a storagedevice.
- Fig 2.37 (b) illustrates the object-levelaccess.
- After the object is stored, the OSD sends an acknowledgment to the applicationserver.
- The OSD storage component manages all the required low-level storage and space management functions. It also manages security and access control functions for theobjects.

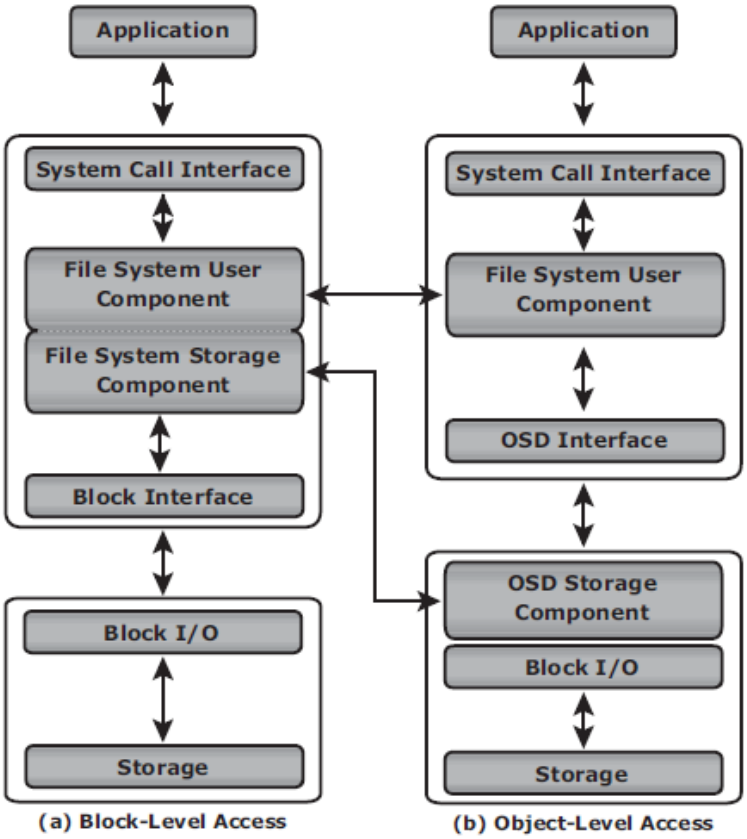


Fig 2.37 Block-level access versus object-level access

2.14.2 Components ofOSD

- The OSD system is typically composed of three keycomponents:
    1. nodes
    2. privatenetwork
    3. storage
- Fig 2.38 illustrates the components ofOSD.
- 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 system is composed of one or morenodes.
- The OSD node has *two* keyservices:
    1. The ***metadata service*** is responsible for generating the object ID from the contents of a file. It also maintains the mapping of the object IDs and the file systemnamespace.
    2. The ***storage service*** manages a set of disks on which the user data isstored.
- The OSD nodes connect to the storage via an **internal network (private network)**. The internal network provides node-to-node connectivity and node-to-storageconnectivity.
- The application server accesses the node to store and retrieve data over an *externalnetwork*.
- For **storage,** 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 thesystem.

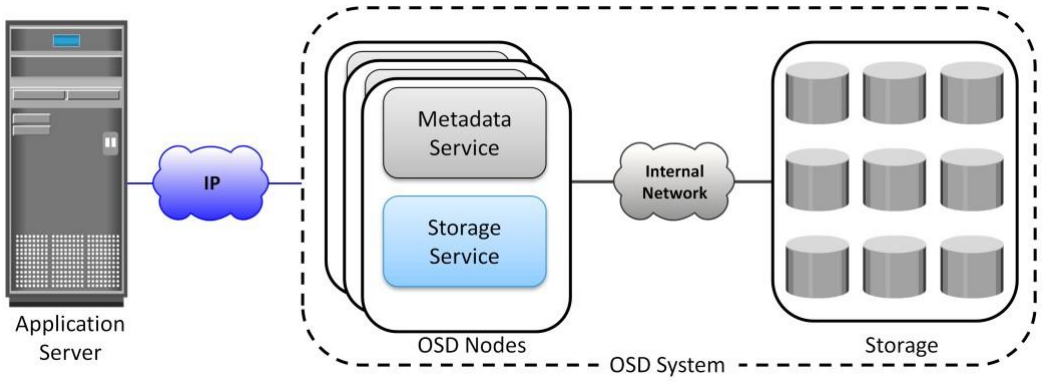


Fig 2.38: OSD Components

2.14.3 Object Storage and Retrieval in OSD

- The process of storing objects in OSD is illustrated in Fig2.39.
- The data storage process in an OSD system is asfollows:

1. The application server presents the file to be stored to the OSDnode.
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 thisdata.
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 isstored.

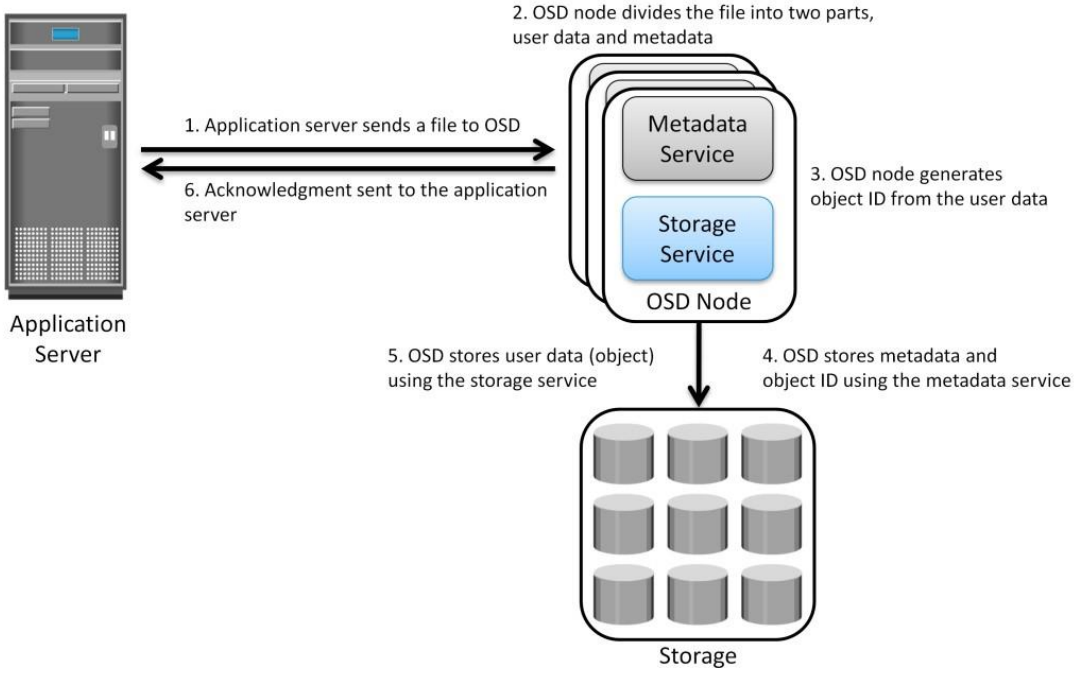


Fig 2.39: Storing objects on OSD

- A user accesses the data stored on OSD by the samefilename.
- The application server retrieves the stored content using the **object ID**. This process is transparent to theuser.
- The process of retrieving objects in OSD is illustrated in Fig 2.40. The process of data retrieval from OSD is asfollows:

1. The application server sends a ***read request*** to the OSDsystem.
2. The metadata service retrieves the object ID for the requestedfile.
3. The metadata service sends the object ID to the applicationserver.
4. The application server sends the object ID to the OSD storage service for objectretrieval.
5. The OSD storage service retrieves the object from the storagedevice.
6. The OSD storage service sends the file to the applicationserver.

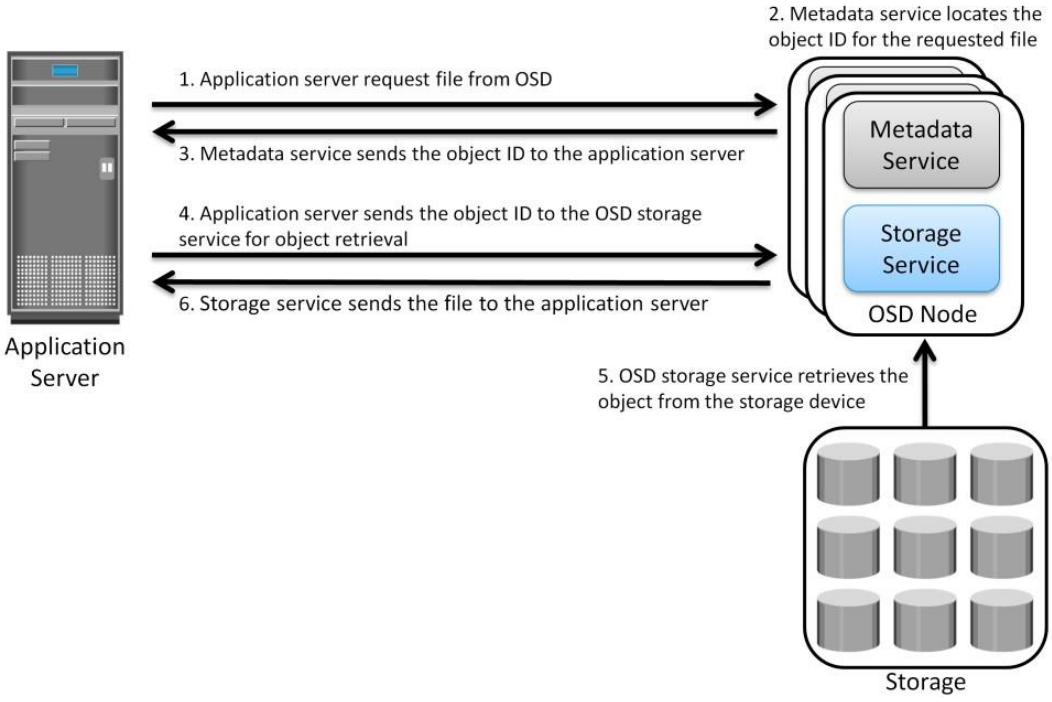


Fig 2.40: Object retrieval from an OSD system

2.14.4 Benefits of Object-BasedStorage

- The key benefits of object-based storage are asfollows:
    - **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 authenticationmechanism.
    - **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 makes object-based storage the best candidate for cloud computing environments.
    - **Scalability:** Due to the use of flat address space, object-based storage can handle large amounts of data without impacting performance. Both storage and OSD nodes can be scaled independently in terms of performance andcapacity.
    - **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 jobsautomatically.

2.14.5 Common Use Cases for Object-BasedStorage

- 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 fixedcontent.
- Cloud-based storage is another use case of OSD. OSD uses a web interface to access storage resources. OSD provides inherent security, scalability, and automated data management. It also enablesdatasharingacrossheterogeneousplatformsortenantswhileensuringintegrityofdata.

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 theweb.

2.15 UnifiedStorage

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

2.15.1 Components of Unified Storage

- A unified storage system consists of the following keycomponents:
    - storagecontroller,
    - NAShead,
    - OSDnode,
    - storage.
- Fig 2.41 illustrates the block diagram of a unified storageplatform.
- The **storage controller or storage processor** provides block-level access to application servers through iSCSI, FC, or FCoE protocols. It contains the corresponding front-end ports for direct block access. The storage controller is also responsible for managing the back-end storage pool in the storagesystem.
- 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 storingdata.

- **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 forredundancy.
- 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 NASclients.
- **The OSD node** also 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 applicationservers.

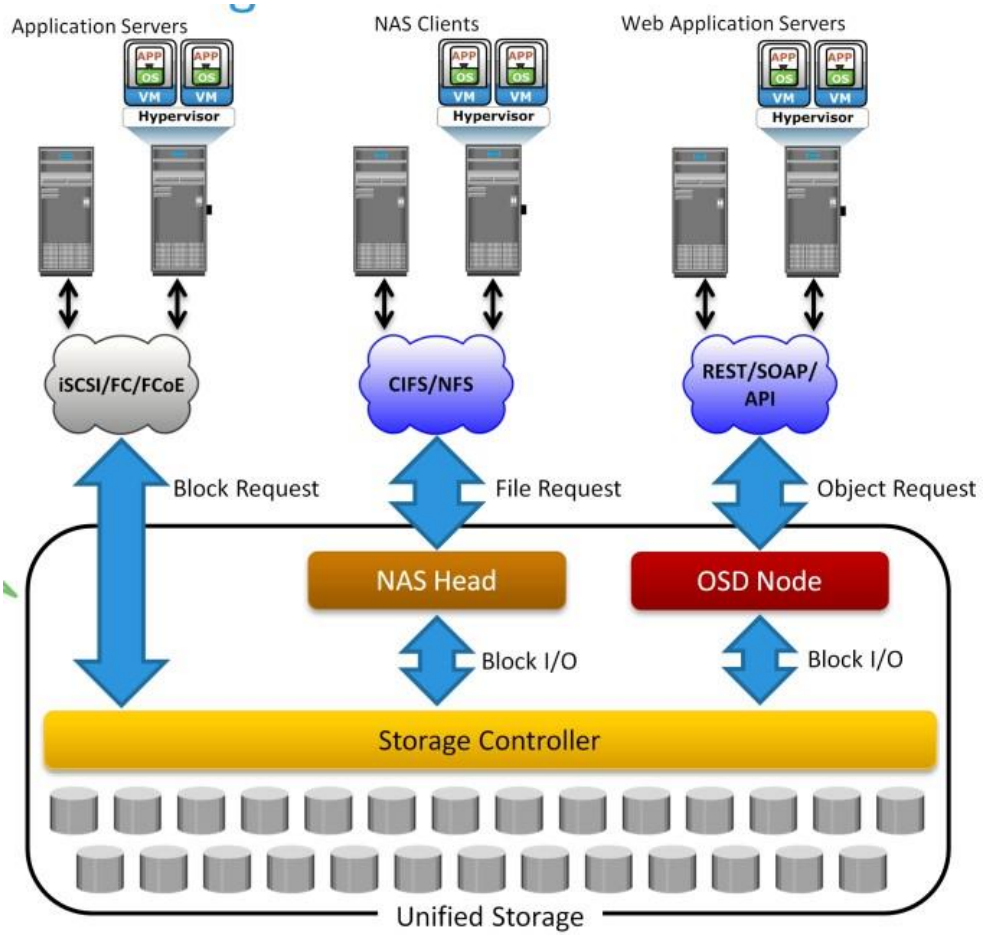


Fig 2.41 Unified Storage Platform

Data Access from Unified Storage

- In a unified storage system, block, file, and object requests to the storage travel through different I/O paths. Fig 2.41 also illustrates the different I/O paths for block, file, and objectaccess.
- **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 and the storage processor (SP) processes the I/O and responds to the applicationserver.
- **File I/O request:** The NAS clients send a file request to the NAS head using 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, the NAS head again converts the block request back to the file request and sends back it to theclients.
- **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 applicationserver.