**System Security**

Security of a computer system is a crucial task. It is a process of ensuring confidentiality and integrity of the OS.  
A system is said to be secure if its resources are used and accessed as intended under all the circumstances, but no system can guarantee absolute security from several of the various malicious threats and unauthorized access.

Security of a system can be threatened via two violations:

* **Threat:** A program which has the potential to cause serious damage to the system.
* **Attack:** An attempt to break security and make unauthorized use of an asset.

Security violations affecting the system can be categorized as malicious and accidental. **Malicious threats**, as the name suggests are a kind of harmful computer code or web script designed to create system vulnerabilities leading to back doors and security breaches. **Accidental Threats**, on the other hand, are comparatively easier to be protected against. Example: Denial of Service DDoS attack.

Security can be compromised via any of the breaches mentioned:

* **Breach of confidentiality:** This type of violation involves the unauthorized reading of data.
* **Breach of integrity:** This violation involves unauthorized modification of data.
* **Breach of availability:** It involves an unauthorized destruction of data.
* **Theft of service:** It involves an unauthorized use of resources.
* **Denial of service:** It involves preventing legitimate use of the system. As mentioned before, such attacks can be accidental in nature.

**Security System Goals –**  
Henceforth, based on the above breaches, the following security goals are aimed:

1. **Integrity:**  
   The objects in the system mustn’t be accessed by any unauthorized user & any user not having sufficient rights should not be allowed to modify the important system files and resources.
2. **Secrecy:**  
   The objects of the system must be accessible only to a limited number of authorized users. Not everyone should be able to view the system files.
3. **Availability:**  
   All the resources of the system must be accessible to all the authorized users i.e only one user/process should not have the right to hog all the system resources. If such kind of situation occurs, denial of service could happen. In this kind of situation, a malware might hog the resources for itself & thus preventing the legitimate processes from accessing the system resources.

Threats can be classified into the following two categories:

1. **Program Threats:**  
   A program written by a cracker to hijack the security or to change the behaviour of a normal process.
2. **System Threats:**  
   These threats involve the abuse of system services. They strive to create a situation in which operating-system resources and user files are misused. They are also used as a medium to launch program threats.

**Types of Program Threats –**

1. **Virus:**  
   An infamous threat, known most widely. It is a self-replicating and a malicious thread which attaches itself to a system file and then rapidly replicates itself, modifying and destroying essential files leading to a system breakdown.

Further, Types of computer viruses can be described briefly as follows:  
– file/parasitic – appends itself to a file  
– boot/memory – infects the boot sector  
– macro – written in a high-level language like VB and affects MS Office files  
– source code – searches and modifies source codes  
– polymorphic – changes in copying each time  
– encrypted – encrypted virus + decrypting code  
– stealth – avoids detection by modifying parts of the system that can be used to detect it, like the read system call  
– tunneling – installs itself in the interrupt service routines and device drivers  
– multipartite – infects multiple parts of the system

1. **Trojan Horse:**  
   A code segment that misuses its environment is called a Trojan Horse. They seem to be attractive and harmless cover program but are a really harmful hidden program which can be used as the virus carrier. In one of the versions of Trojan, User is fooled to enter its confidential login details on an application. Those details are stolen by a login emulator and can be further used as a way of information breaches.

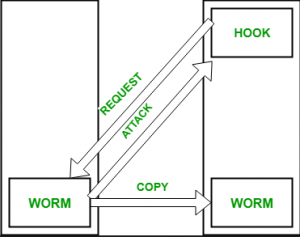
Another variance is Spyware, Spyware accompanies a program that the user has chosen to install and downloads ads to display on the user’s system, thereby creating pop-up browser windows and when certain sites are visited by the user, it captures essential information and sends it over to the remote server. Such attacks are also known as **Covert Channels**.

1. **Trap Door:**  
   The designer of a program or system might leave a hole in the software that only he is capable of using, the Trap Door works on the similar principles. Trap Doors are quite difficult to detect as to analyze them, one needs to go through the source code of all the components of the system.
2. **Logic Bomb:**  
   A program that initiates a security attack only under a specific situation.

**Types of System Threats –**  
Aside from the program threats, various system threats are also endangering the security of our system:

1.**Worm:**  
An infection program which spreads through networks. Unlike a virus, they target mainly LANs. A computer affected by a worm attacks the target system and writes a small program “hook” on it. This hook is further used to copy the worm to the target computer. This process repeats recursively, and soon enough all the systems of the LAN are affected. It uses the spawn mechanism to duplicate itself. The worm spawns copies of itself, using up a majority of system resources and also locking out all other processes.

The basic functionality of the worm can be represented as:



2. **Port Scanning:**  
It is a means by which the cracker identifies the vulnerabilities of the system to attack. It is an automated process which involves creating a TCP/IP connection to a specific port. To protect the identity of the attacker, port scanning attacks are launched from **Zombie Systems**, that is systems which were previously independent systems that are also serving their owners while being used for such notorious purposes.

3. **Denial of Service:**  
Such attacks aren’t aimed for the purpose of collecting information or destroying system files. Rather, they are used for disrupting the legitimate use of a system or facility.  
These attacks are generally network based. They fall into two categories:  
– Attacks in this first category use so many system resources that no useful work can be performed.

For example, downloading a file from a website that proceeds to use all available CPU time.  
– Attacks in the second category involves disrupting the network of the facility. These attacks are a result of the abuse of some fundamental TCP/IP principles.  
fundamental functionality of TCP/IP.

**Security Measures Taken –**  
To protect the system, Security measures can be taken at the following levels:

* **Physical:**  
  The sites containing computer systems must be physically secured against armed and malicious intruders. The workstations must be carefully protected.
* **Human:**  
  Only appropriate users must have the authorization to access the system. Phishing(collecting confidential information) and Dumpster Diving(collecting basic information so as to gain unauthorized access) must be avoided.
* **Operating system:**  
  The system must protect itself from accidental or purposeful security breaches.
* **Networking System:**  
  Almost all of the information is shared between different systems via a network. Intercepting these data could be just as harmful as breaking into a computer. Henceforth, Network should be properly secured against such attacks.

**Protection**

**14.1 Goals of Protection**

* Obviously to prevent malicious misuse of the system by users or programs. See chapter 15 for a more thorough coverage of this goal.
* To ensure that each shared resource is used only in accordance with system *policies,*which may be set either by system designers or by system administrators.
* To ensure that errant programs cause the minimal amount of damage possible.
* Note that protection systems only provide the *mechanisms* for enforcing policies and ensuring reliable systems. It is up to administrators and users to implement those mechanisms effectively.

**14.2 Principles of Protection**

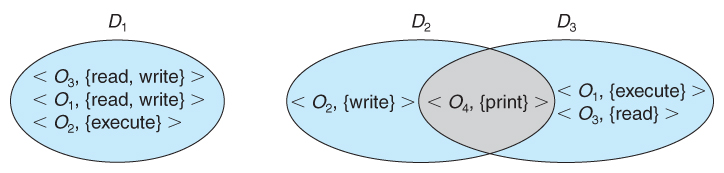
* The ***principle of least privilege***dictates that programs, users, and systems be given just enough privileges to perform their tasks.
* This ensures that failures do the least amount of harm and allow the least of harm to be done.
* For example, if a program needs special privileges to perform a task, it is better to make it a SGID program with group ownership of "network" or "backup" or some other pseudo group, rather than SUID with root ownership. This limits the amount of damage that can occur if something goes wrong.
* Typically each user is given their own account, and has only enough privilege to modify their own files.
* The root account should not be used for normal day to day activities - The System Administrator should also have an ordinary account, and reserve use of the root account for only those tasks which need the root privileges

**14.3 Domain of Protection**

* A computer can be viewed as a collection of *processes* and *objects* ( both HW & SW ).
* The ***need to know principle*** states that a process should only have access to those objects it needs to accomplish its task, and furthermore only in the modes for which it needs access and only during the time frame when it needs access.
* The modes available for a particular object may depend upon its type.

**14.3.1 Domain Structure**

* A ***protection domain***specifies the resources that a process may access.
* Each domain defines a set of objects and the types of operations that may be invoked on each object.
* An ***access right*** is the ability to execute an operation on an object.
* A domain is defined as a set of < object, { access right set } > pairs, as shown below. Note that some domains may be disjoint while others overlap.

  
 **System with three protection domains.**

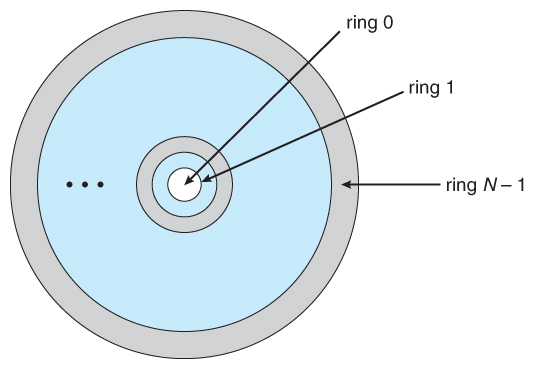
* The association between a process and a domain may be *static*or *dynamic.*
  + If the association is static, then the need-to-know principle requires a way of changing the contents of the domain dynamically.
  + If the association is dynamic, then there needs to be a mechanism for ***domain switching.***
* Domains may be realized in different fashions - as users, or as processes, or as procedures. E.g. if each user corresponds to a domain, then that domain defines the access of that user, and changing domains involves changing user ID.

**14.3.2 An Example: UNIX**

* UNIX associates domains with users.
* Certain programs operate with the SUID bit set, which effectively changes the user ID, and therefore the access domain, while the program is running. ( and similarly for the SGID bit. ) Unfortunately this has some potential for abuse.
* An alternative used on some systems is to place privileged programs in special directories, so that they attain the identity of the directory owner when they run. This prevents crackers from placing SUID programs in random directories around the system.
* Yet another alternative is to not allow the changing of ID at all. Instead, special privileged daemons are launched at boot time, and user processes send messages to these daemons when they need special tasks performed.

**14.3.3 An Example: MULTICS**

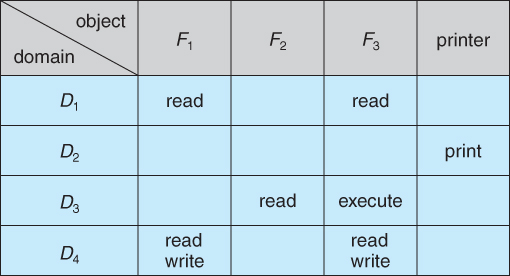
* The MULTICS system uses a complex system of rings, each corresponding to a different protection domain, as shown below:

  
**Figure MULTICS ring structure.**

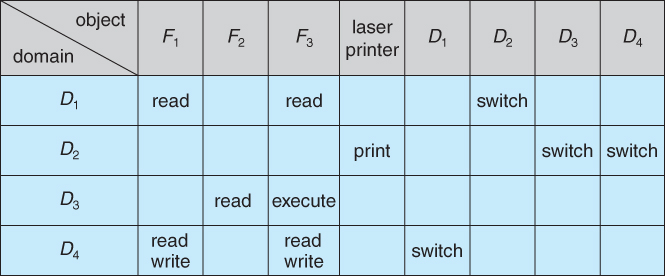
* Rings are numbered from 0 to 7, with outer rings having a subset of the privileges of the inner rings.
* Each file is a memory segment, and each segment description includes an entry that indicates the ring number associated with that segment, as well as read, write, and execute privileges.
* Each process runs in a ring, according to the *current-ring-number,*a counter associated with each process.
* A process operating in one ring can only access segments associated with higher ( farther out ) rings, and then only according to the access bits. Processes cannot access segments associated with lower rings.
* Domain switching is achieved by a process in one ring calling upon a process operating in a lower ring, which is controlled by several factors stored with each segment descriptor:
  + An ***access bracket***, defined by integers b1 <= b2.
  + A ***limit*** b3 > b2
  + A ***list of gates,***identifying the entry points at which the segments may be called.
* If a process operating in ring i calls a segment whose bracket is such that b1 <= i <= b2, then the call succeeds and the process remains in ring i.
* Otherwise a trap to the OS occurs, and is handled as follows:
  + If i < b1, then the call is allowed, because we are transferring to a procedure with fewer privileges. However if any of the parameters being passed are of segments below b1, then they must be copied to an area accessible by the called procedure.
  + If i > b2, then the call is allowed only if i <= b3 and the call is directed to one of the entries on the list of gates.
* Overall this approach is more complex and less efficient than other protection schemes.

**14.4 Access Matrix**

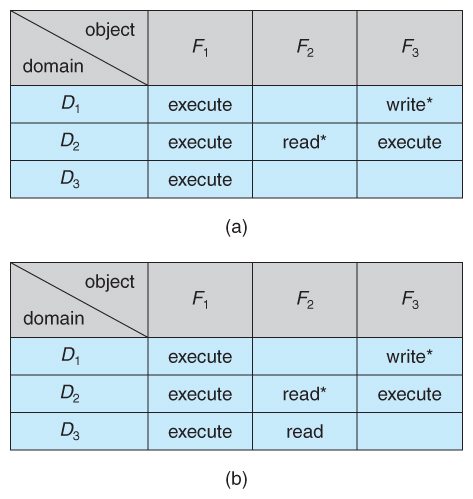
* The model of protection that we have been discussing can be viewed as an ***access matrix,***in which columns represent different system resources and rows represent different protection domains. Entries within the matrix indicate what access that domain has to that resource.

  
**Figure Access matrix.**

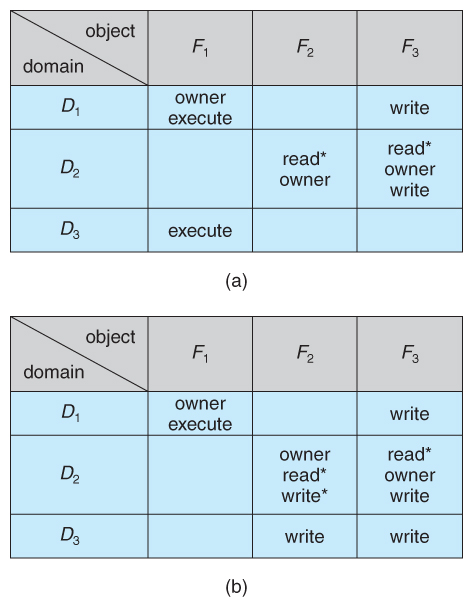
* Domain switching can be easily supported under this model, simply by providing "switch" access to other domains:

  
**Figure - Access matrix of Figure with domains as objects.**

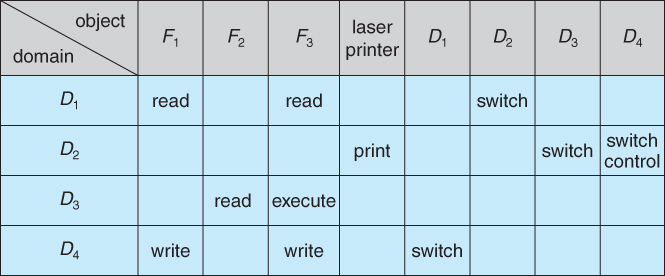
* The ability to ***copy***rights is denoted by an asterisk, indicating that processes in that domain have the right to copy that access within the same column, i.e. for the same object. There are two important variations:
  + If the asterisk is removed from the original access right, then the right is ***transferred,***rather than being copied. This may be termed a ***transfer*** right as opposed to a ***copy*** right.
  + If only the right and not the asterisk is copied, then the access right is added to the new domain, but it may not be propagated further. That is the new domain does not also receive the right to copy the access. This may be termed a ***limited copy*** right, as shown in Figure 14.5 below:

  
**Figure - Access matrix with *copy* rights.**

* The ***owner*** right adds the privilege of adding new rights or removing existing ones:

  
**Figure - Access matrix with *owner* rights.**

* Copy and owner rights only allow the modification of rights within a column. The addition of ***control rights***, which only apply to domain objects, allow a process operating in one domain to affect the rights available in other domains. For example in the table below, a process operating in domain D2 has the right to control any of the rights in domain D4.

  
**Figure - Modified access matrix of Figure 14.4**

**14.5 Implementation of Access Matrix**

**14.5.1 Global Table**

* The simplest approach is one big global table with < domain, object, rights > entries.
* Unfortunately this table is very large ( even if sparse ) and so cannot be kept in memory ( without invoking virtual memory techniques. )
* There is also no good way to specify groupings - If everyone has access to some resource, then it still needs a separate entry for every domain.

**14.5.2 Access Lists for Objects**

* Each column of the table can be kept as a list of the access rights for that particular object, discarding blank entries.
* For efficiency a separate list of default access rights can also be kept, and checked first.

**14.5.3 Capability Lists for Domains**

* In a similar fashion, each row of the table can be kept as a list of the capabilities of that domain.
* Capability lists are associated with each domain, but not directly accessible by the domain or any user process.
* Capability lists are themselves protected resources, distinguished from other data in one of two ways:
  + A ***tag,***possibly hardware implemented, distinguishing this special type of data. ( other types may be floats, pointers, booleans, etc. )
  + The address space for a program may be split into multiple segments, at least one of which is inaccessible by the program itself, and used by the operating system for maintaining the process's access right capability list.

**14.5.4 A Lock-Key Mechanism**

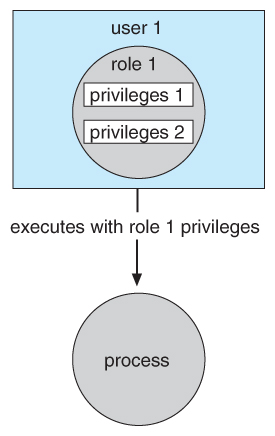
* Each resource has a list of unique bit patterns, termed locks.
* Each domain has its own list of unique bit patterns, termed keys.
* Access is granted if one of the domain's keys fits one of the resource's locks.
* Again, a process is not allowed to modify its own keys.

**14.5.5 Comparison**

* Each of the methods here has certain advantages or disadvantages, depending on the particular situation and task at hand.
* Many systems employ some combination of the listed methods.

**14.6 Access Control**

* ***Role-Based Access Control, RBAC,***assigns privileges to users, programs, or roles as appropriate, where "privileges" refer to the right to call certain system calls, or to use certain parameters with those calls.
* RBAC supports the principle of least privilege, and reduces the susceptibility to abuse as opposed to SUID or SGID programs.

  
**Figure - Role-based access control in Solaris 10.**

**14.7 Revocation of Access Rights**

* The need to revoke access rights dynamically raises several questions:
  + Immediate versus delayed - If delayed, can we determine when the revocation will take place?
  + Selective versus general - Does revocation of an access right to an object affect *all* users who have that right, or only some users?
  + Partial versus total - Can a subset of rights for an object be revoked, or are all rights revoked at once?
  + Temporary versus permanent - If rights are revoked, is there a mechanism for processes to re-acquire some or all of the revoked rights?
* With an access list scheme revocation is easy, immediate, and can be selective, general, partial, total, temporary, or permanent, as desired.
* With capabilities lists the problem is more complicated, because access rights are distributed throughout the system. A few schemes that have been developed include:
  + Reacquisition - Capabilities are periodically revoked from each domain, which must then re-acquire them.
  + Back-pointers - A list of pointers is maintained from each object to each capability which is held for that object.
  + Indirection - Capabilities point to an entry in a global table rather than to the object. Access rights can be revoked by changing or invalidating the table entry, which may affect multiple processes, which must then re-acquire access rights to continue.
  + Keys - A unique bit pattern is associated with each capability when created, which can be neither inspected nor modified by the process.
    - A master key is associated with each object.
    - When a capability is created, its key is set to the object's master key.
    - As long as the capability's key matches the object's key, then the capabilities remain valid.
    - The object master key can be changed with the set-key command, thereby invalidating all current capabilities.
    - More flexibility can be added to this scheme by implementing a ***list*** of keys for each object, possibly in a global table.

## Encryption/Decryption

**Encryption:** Encryption means that the sender converts the original information into another form and sends the unintelligible message over the network.

**Decryption:** Decryption reverses the Encryption process in order to transform the message back to the original form.

The data which is to be encrypted at the sender site is known as plaintext, and the encrypted data is known as ciphertext. The data is decrypted at the receiver site.

**There are two types of Encryption/Decryption techniques:**

* Privacy with secret key Encryption/Decryption
* Privacy with public key Encryption/Decryption

## Secret Key Encryption/Decryption technique

* In Secret Key Encryption/Decryption technique, the same key is used by both the parties, i.e., the sender and receiver.
* The sender uses the secret key and encryption algorithm to encrypt the data; the receiver uses this key and decryption algorithm to decrypt the data.
* In Secret Key Encryption/Decryption technique, the algorithm used for encryption is the inverse of the algorithm used for decryption. It means that if the encryption algorithm uses a combination of addition and multiplication, then the decryption algorithm uses a combination of subtraction and division.
* The secret key encryption algorithm is also known as symmetric encryption algorithm because the same secret key is used in bidirectional communication.
* In secret key encryption/decryption algorithm, the secret code is used by the computer to encrypt the information before it is sent over the network to another computer.
* The secret key requires that we should know which computers are talking to each other so that we can install the key on each computer.

### Data Encryption Standard (DES)

* The Data Encryption Standard (DES) was designed by IBM and adopted by the U.S. government as the standard encryption method for nonmilitary and nonclassified use.
* The Data Encryption Standard is a standard used for encryption, and it is a form of Secret **Key Cryptography**.

### Advantage

**Efficient:** The secret key algorithms are more efficient as it takes less time to encrypt the message than to encrypt the message by using a public key encryption algorithm. The reason for this is that the size of the key is small. Due to this reason, Secret Key Algorithms are mainly used for encryption and decryption.

### Disadvantages of Secret Key Encryption

**The Secret Key Encryption/Decryption has the following disadvantages:**

* Each pair of users must have a secret key. If the number of people wants to use this method in the world is N, then there are N(N-1)/2 secret keys. For example, for one million people, then there are half billion secret keys.
* The distribution of keys among different parties can be very difficult. This problem can be resolved by combining the Secret Key Encryption/Decryption with the Public Key Encryption/Decryption algorithm.

## Public Key Encryption/Decryption technique

* There are two keys in public key encryption: a private key and a public key.
* The private key is given to the receiver while the public key is provided to the public.

In the above figure, we see that A is sending the message to user B. 'A' uses the public key to encrypt the data while 'B' uses the private key to decrypt the data.

* In public key Encryption/Decryption, the public key used by the sender is different from the private key used by the receiver.
* The public key is available to the public while the private key is kept by each individual.
* The most commonly used public key algorithm is known as RSA.

### Advantages of Public Key Encryption

* The main restriction of private key encryption is the sharing of a secret key. A third party cannot use this key. In public key encryption, each entity creates a pair of keys, and they keep the private one and distribute the public key.
* The number of keys in public key encryption is reduced tremendously. For example, for one million users to communicate, only two million keys are required, not a half-billion keys as in the case of secret key encryption.

### Disadvantages of Public Key Encryption

* **Speed:** One of the major disadvantage of the public-key encryption is that it is slower than secret-key encryption. In secret key encryption, a single shared key is used to encrypt and decrypt the message which speeds up the process while in public key encryption, different two keys are used, both related to each other by a complex mathematical process. Therefore, we can say that encryption and decryption take more time in public key encryption.
* **Authentication:** A public key encryption does not have a built-in authentication. Without authentication, the message can be interpreted or intercepted without the user's knowledge.
* **Inefficient:** The main disadvantage of the public key is its complexity. If we want the method to be effective, large numbers are needed. But in public key encryption, converting the plaintext into ciphertext using long keys takes a lot of time. Therefore, the public key encryption algorithms are efficient for short messages not for long messages.

## Differences b/w Secret Key Encryption & Public Key Encryption

|  |  |  |
| --- | --- | --- |
| **Basis for Comparison** | **Secret Key Encryption** | **Public Key Encryption** |
| Define | Secret Key Encryption is defined as the technique that uses a single shared key to encrypt and decrypt the message. | Public Key Encryption is defined as the technique that uses two different keys for encryption and decryption. |
| Efficieny | It is efficient as this technique is recommended for large amounts of text. | It is inefficient as this technique is used only for short messages. |
| Other name | It is also known as Symmetric Key encryption. | It is also known as Asymmetric Key Encryption. |
| Speed | Its speed is high as it uses a single key for encryption and decryption. | Its speed is slow as it uses two different keys, both keys are related to each other through the complicated mathematical process. |
| Algorithms | The Secret key algorithms are DES, 3DES, AES & RCA. | The Public key algorithms are Diffie-Hellman, RSA. |
| Purpose | The main purpose of the secret key algorithm is to transmit the bulk data. | The main purpose of the public key algorithm is to share the keys securely. |

**TOPOLOGIES**

# What is Topology?

Topology defines the structure of the network of how all the components are interconnected to each other. There are two types of topology: physical and logical topology.

Physical topology is the geometric representation of all the nodes in a network.

## Bus Topology

* The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
* Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable.
* When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
* The bus topology is mainly used in 802.3 (ethernet) and 802.4 standard networks.
* The configuration of a bus topology is quite simpler as compared to other topologies.
* The backbone cable is considered as a **"single lane"** through which the message is broadcast to all the stations.
* The most common access method of the bus topologies is **CSMA** (Carrier Sense Multiple Access).

**CSMA:** It is a media access control used to control the data flow so that data integrity is maintained, i.e., the packets do not get lost. There are two alternative ways of handling the problems that occur when two nodes send the messages simultaneously.

* **CSMA CD:** CSMA CD (**Collision detection**) is an access method used to detect the collision. Once the collision is detected, the sender will stop transmitting the data. Therefore, it works on "**recovery after the collision**".
* **CSMA CA:** **CSMA CA (Collision Avoidance)** is an access method used to avoid the collision by checking whether the transmission media is busy or not. If busy, then the sender waits until the media becomes idle. This technique effectively reduces the possibility of the collision. It does not work on "recovery after the collision".

### Advantages of Bus topology:

* **Low-cost cable:** In bus topology, nodes are directly connected to the cable without passing through a hub. Therefore, the initial cost of installation is low.
* **Moderate data speeds:** Coaxial or twisted pair cables are mainly used in bus-based networks that support upto 10 Mbps.
* **Familiar technology:** Bus topology is a familiar technology as the installation and troubleshooting techniques are well known, and hardware components are easily available.
* **Limited failure:** A failure in one node will not have any effect on other nodes.

### Disadvantages of Bus topology:

* **Extensive cabling:** A bus topology is quite simpler, but still it requires a lot of cabling.
* **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
* **Signal interference:** If two nodes send the messages simultaneously, then the signals of both the nodes collide with each other.
* **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
* **Attenuation:** Attenuation is a loss of signal leads to communication issues. Repeaters are used to regenerate the signal.

## Ring Topology

* Ring topology is like a bus topology, but with connected ends.
* The node that receives the message from the previous computer will retransmit to the next node.
* The data flows in one direction, i.e., it is unidirectional.
* The data flows in a single loop continuously known as an endless loop.
* It has no terminated ends, i.e., each node is connected to other node and having no termination point.
* The data in a ring topology flow in a clockwise direction.
* The most common access method of the ring topology is **token passing**.
  + **Token passing:** It is a network access method in which token is passed from one node to another node.
  + **Token:** It is a frame that circulates around the network.

### Working of Token passing

* A token moves around the network, and it is passed from computer to computer until it reaches the destination.
* The sender modifies the token by putting the address along with the data.
* The data is passed from one device to another device until the destination address matches. Once the token received by the destination device, then it sends the acknowledgment to the sender.
* In a ring topology, a token is used as a carrier.

### Advantages of Ring topology:

* **Network Management:** Faulty devices can be removed from the network without bringing the network down.
* **Product availability:** Many hardware and software tools for network operation and monitoring are available.
* **Cost:** Twisted pair cabling is inexpensive and easily available. Therefore, the installation cost is very low.
* **Reliable:** It is a more reliable network because the communication system is not dependent on the single host computer.

### Disadvantages of Ring topology:

* **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
* **Failure:** The breakdown in one station leads to the failure of the overall network.
* **Reconfiguration difficult:** Adding new devices to the network would slow down the network.
* **Delay:** Communication delay is directly proportional to the number of nodes. Adding new devices increases the communication delay.

## Star Topology

* Star topology is an arrangement of the network in which every node is connected to the central hub, switch or a central computer.
* The central computer is known as a **server**, and the peripheral devices attached to the server are known as **clients**.
* Coaxial cable or RJ-45 cables are used to connect the computers.
* Hubs or Switches are mainly used as connection devices in a **physical star topology**.
* Star topology is the most popular topology in network implementation.

### Advantages of Star topology

* **Efficient troubleshooting:** Troubleshooting is quite efficient in a star topology as compared to bus topology. In a bus topology, the manager has to inspect the kilometers of cable. In a star topology, all the stations are connected to the centralized network. Therefore, the network administrator has to go to the single station to troubleshoot the problem.
* **Network control:** Complex network control features can be easily implemented in the star topology. Any changes made in the star topology are automatically accommodated.
* **Limited failure:** As each station is connected to the central hub with its own cable, therefore failure in one cable will not affect the entire network.
* **Familiar technology:** Star topology is a familiar technology as its tools are cost-effective.
* **Easily expandable:** It is easily expandable as new stations can be added to the open ports on the hub.
* **Cost effective:** Star topology networks are cost-effective as it uses inexpensive coaxial cable.
* **High data speeds:** It supports a bandwidth of approx 100Mbps. Ethernet 100BaseT is one of the most popular Star topology networks.

### Disadvantages of Star topology

* **A Central point of failure:** If the central hub or switch goes down, then all the connected nodes will not be able to communicate with each other.
* **Cable:** Sometimes cable routing becomes difficult when a significant amount of routing is required.

## Tree topology

* Tree topology combines the characteristics of bus topology and star topology.
* A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
* The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
* There is only one path exists between two nodes for the data transmission. Thus, it forms a parent-child hierarchy.

### Advantages of Tree topology

* **Support for broadband transmission:** Tree topology is mainly used to provide broadband transmission, i.e., signals are sent over long distances without being attenuated.
* **Easily expandable:** We can add the new device to the existing network. Therefore, we can say that tree topology is easily expandable.
* **Easily manageable:** In tree topology, the whole network is divided into segments known as star networks which can be easily managed and maintained.
* **Error detection:** Error detection and error correction are very easy in a tree topology.
* **Limited failure:** The breakdown in one station does not affect the entire network.
* **Point-to-point wiring:** It has point-to-point wiring for individual segments.

### Disadvantages of Tree topology

* **Difficult troubleshooting:** If any fault occurs in the node, then it becomes difficult to troubleshoot the problem.
* **High cost:** Devices required for broadband transmission are very costly.
* **Failure:** A tree topology mainly relies on main bus cable and failure in main bus cable will damage the overall network.
* **Reconfiguration difficult:** If new devices are added, then it becomes difficult to reconfigure.

## Mesh topology

* Mesh technology is an arrangement of the network in which computers are interconnected with each other through various redundant connections.
* There are multiple paths from one computer to another computer.
* It does not contain the switch, hub or any central computer which acts as a central point of communication.
* The Internet is an example of the mesh topology.
* Mesh topology is mainly used for WAN implementations where communication failures are a critical concern.
* Mesh topology is mainly used for wireless networks.
* Mesh topology can be formed by using the formula:  
  **Number of cables = (n\*(n-1))/2;**

Where n is the number of nodes that represents the network.

**Mesh topology is divided into two categories:**

* Fully connected mesh topology
* Partially connected mesh topology
* **Full Mesh Topology:** In a full mesh topology, each computer is connected to all the computers available in the network.
* **Partial Mesh Topology:** In a partial mesh topology, not all but certain computers are connected to those computers with which they communicate frequently.

### Advantages of Mesh topology:

**Reliable:** The mesh topology networks are very reliable as if any link breakdown will not affect the communication between connected computers.

**Fast Communication:** Communication is very fast between the nodes.

**Easier Reconfiguration:** Adding new devices would not disrupt the communication between other devices.

### Disadvantages of Mesh topology

* **Cost:** A mesh topology contains a large number of connected devices such as a router and more transmission media than other topologies.
* **Management:** Mesh topology networks are very large and very difficult to maintain and manage. If the network is not monitored carefully, then the communication link failure goes undetected.
* **Efficiency:** In this topology, redundant connections are high that reduces the efficiency of the network.

## Hybrid Topology

* The combination of various different topologies is known as **Hybrid topology**.
* A Hybrid topology is a connection between different links and nodes to transfer the data.
* When two or more different topologies are combined together is termed as Hybrid topology and if similar topologies are connected with each other will not result in Hybrid topology. For example, if there exist a ring topology in one branch of ICICI bank and bus topology in another branch of ICICI bank, connecting these two topologies will result in Hybrid topology.

### Advantages of Hybrid Topology

* **Reliable:** If a fault occurs in any part of the network will not affect the functioning of the rest of the network.
* **Scalable:** Size of the network can be easily expanded by adding new devices without affecting the functionality of the existing network.
* **Flexible:** This topology is very flexible as it can be designed according to the requirements of the organization.
* **Effective:** Hybrid topology is very effective as it can be designed in such a way that the strength of the network is maximized and weakness of the network is minimized.

### Disadvantages of Hybrid topology

* **Complex design:** The major drawback of the Hybrid topology is the design of the Hybrid network. It is very difficult to design the architecture of the Hybrid network.
* **Costly Hub:** The Hubs used in the Hybrid topology are very expensive as these hubs are different from usual Hubs used in other topologies.
* **Costly infrastructure:** The infrastructure cost is very high as a hybrid network requires a lot of cabling, network devices, etc.