Unit-I

Introduction

This chapter includes

Introduction to distributed systems, examples of distributed systems, characteristics, goals, hardware and software concepts, design issues, resource sharing and the web, challenges.

System models : Introduction, Architectural models, Fundamental models and client server models.

1.1 Introduction

The literature survey of distributed systems give various definitions of the distributed systems.

1. A distributed system is a collection of independent computers that appears to its users as a single coherent system by

   "Tanenbaum and van steen".

2. A distributed system is one in which components located at networked computers communicate and co-ordinate their actions only by passing messages by

   "Coulouris, Dollimore and Kindberg".

3. Loosely coupled systems are called as distributed systems.

   Loosely coupled systems are those in which the processors do not share memory. Each processor has its own local memory. The loosely coupled system is shown in Fig. 1.1.

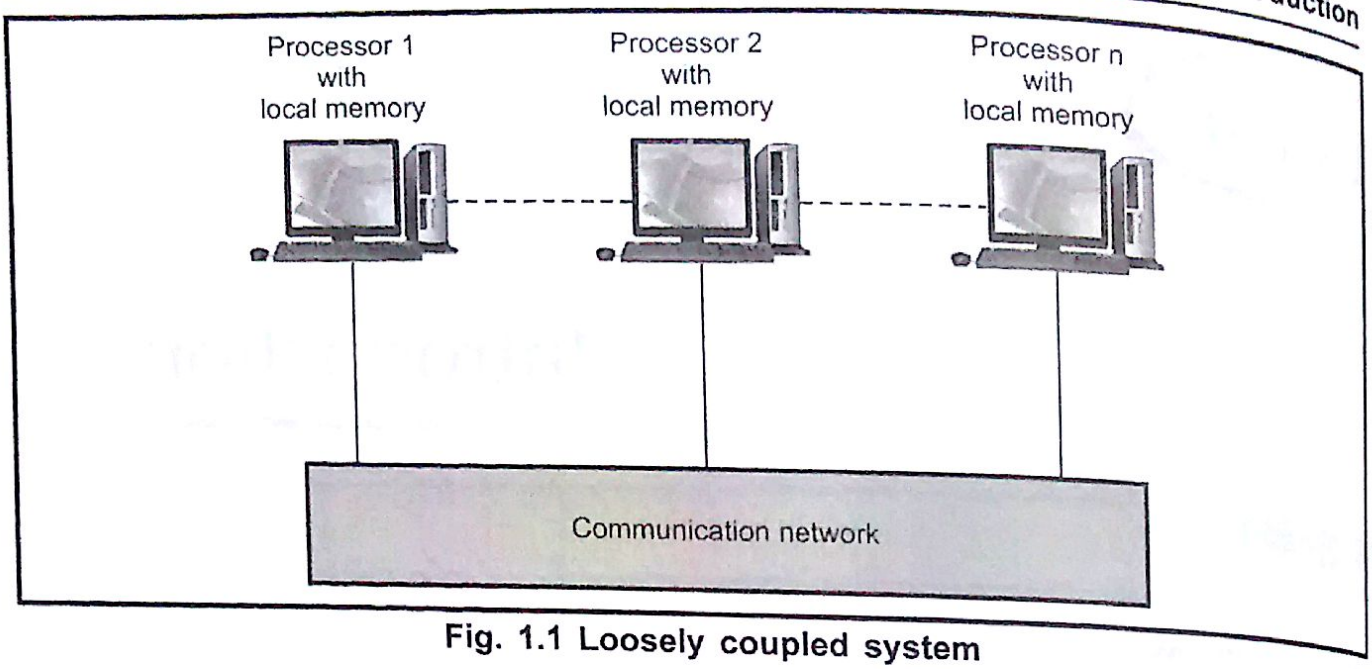

Fig. 1.1 Loosely coupled system

The above definition is given by "P.K. Sinha".

- The first definition has 2 aspects.

a) The first aspect says that the machines used in the distributed systems are autonomous i.e. independent of each other.

b) The second aspect is related to software. The users of the machine think that they are dealing with a single system.

- The second definition leads to following concepts.

a) Concurrent execution

Since there are n computers in the distributed environment each sharing the resources, all resources are available to all users without the feeling of interaction. The capacity of the network can be increased by adding more resources to the network. Thus concurrent program execution is an important feature of the distributed system.

b) Independent failures

A computer may lead to a failure. So also the computer in a network may lead to failure. If a computer in a distributed system fails, this failure is not immediately made known to other components in the system. Fault in the computer isolates it from the system, but the system does not stop running.

Faults in the network, are not made known to other components of the network. Each component of the network can fail independently, still leaving others running successfully.

In brief it can be stated that

A distributed system is a collection of processors. These processors are interconnected by a communication network. In this network each processor will have its own local memory. It will also have its own peripherals.

Since the processors in the distributed system may be located far from each other, the communication between them takes place by message passing over the communication network. For each processor its own resources are local and other resources are remote.

The distributed system gives a single system view. For this feature, a middleware service (software) is the necessity. This middleware service is placed between the higher level layer of users and applications and a lower level consisting of operating system.

This concept can be visualized as follows.

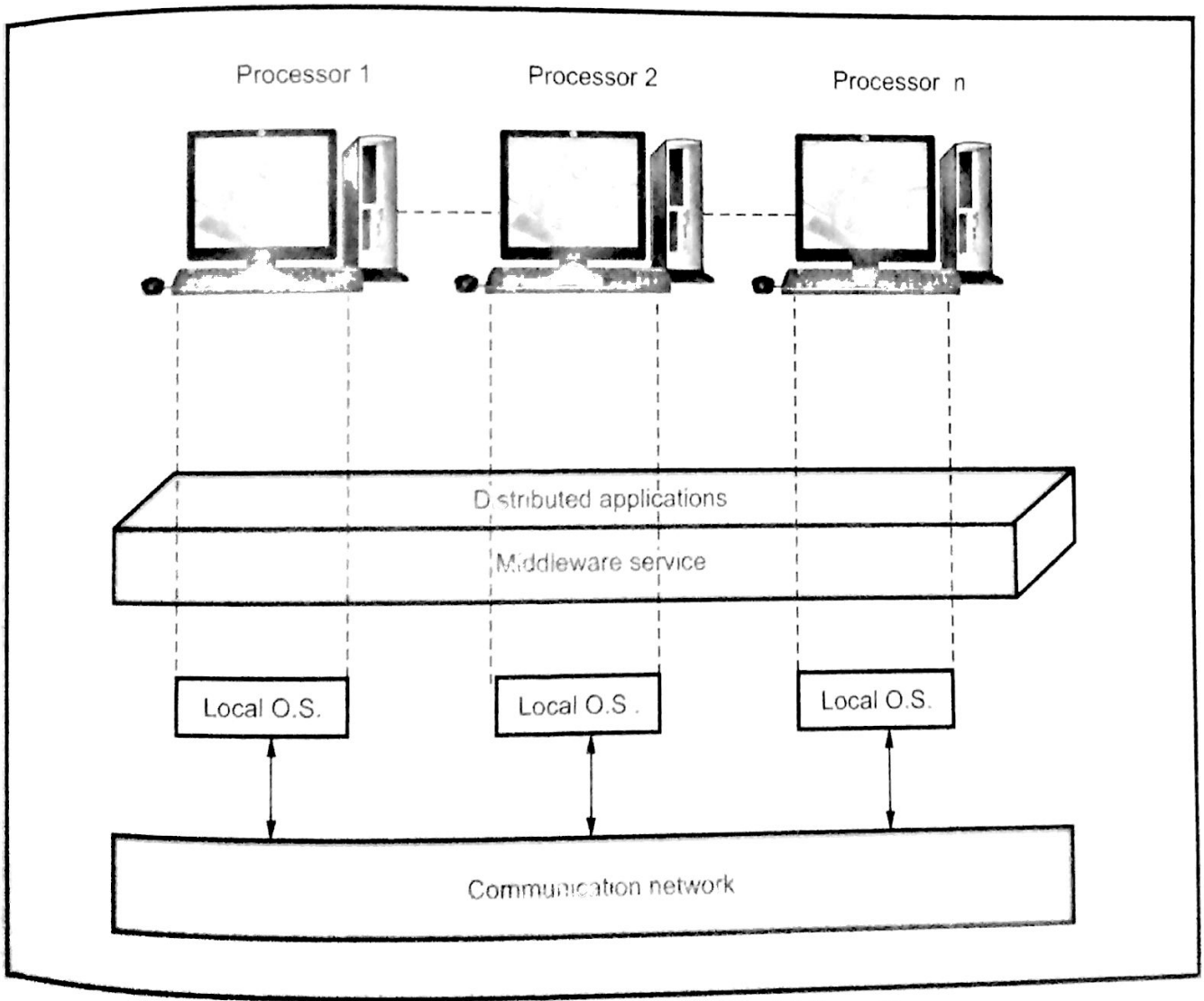

Fig. 1.2 Distributed system as middleware

1.2 Goals of a Distributed System

Four important goals to be achieved by any distributed system are

- Easily connect users and resources.
- Exhibit transparency.
- Support openness.
- Be scalable.

We will see each of these goals in detail.

1.2.1 Easily Connect Users and Resources

With the tremendous progress in the IT industry, a rapid need for sharing resources is raised.

Sharing of resource has its basic pros and cons.

- The pros could be for e.g. : A printer is shared among various computer nodes. This leads towards an economy approach.
- Sharing the web pages and high end server facilities makes it possible to be in touch across the globe communication at far distances become cheaper and faster.
- Sharing makes widely spread people to work under a single group. Internet high end sharing facilities lead to services like teleconferencing, collaborative editing video conferencing etc.
- Sharing through the internet has provided a strong facility like e commerce. All the business transactions could be done staying at their own place without actually moving to the physical location. Buying and selling of goods has also become a simplified procedure.
- Looking at the cons part of the story.

  Connectivity and sharing property has lead to the aspect of security. Security issues are the major concerns of any project involving a network.

  Intruders and hackers have tremendously increased in the upcoming IT era. Sharing makes it possible to share passwords and other sensitive information thru the network.

  It has also led to unwanted information passing like the junk mail or spam.

  Thus the main aim of any distributed system is to easily connect users and resources. The system must handle the security issues with strong servers and high configuration firewalls. It should also provide the unnecessary spreading of the junk data.

1.2.2 Exhibit Transparency

A transparent distributed system looks to its users as if it were only a single computer system. The processes and resources in the distributed system are physically spread but the goal of the distributed system is to hide this fact. There are different types of transparencies in an distributed system.

- **Access transparency**

  There are various ways of representing data. The same data may be represented in two different formats by two companies. A distributed system with access transparency feature allows to access the data hiding the differences in data representation.

  For example :

1. In some format of representation. For example, little endian higher order bytes are transmitted first, whereas in some formats like big endian lower order bytes are transmitted first.

   The D.S. takes into consideration these format differences and hides this feature letting the user to work in his own data format.

2. The distributed systems have computers where different operating systems may be installed. For example, some machine may have windows XP while the other may have UNIX O.S. There is a vast difference in the file naming conventions of both the operating systems.

   The D.S. takes care of these difference and provides the access transparency feature to the users of both the O.S. machines.

- **Concurrency transparency**

  Resources are shared in a distributed environment. This may lead to sharing the same resource by two users at the same time.

  For example, two users may be accessing the same file at the same time. Changes made by one person should not be known to the second person at the same time. This is called as concurrency transparency.

  Another example may be of a database. A same table may be being used by more than one users at the same time. The main feature to be handled in concurrent access is that the data after access must remain in a consistent state. There are features like locking mechanisms and transaction which helps in keeping the data in a consistent format.

- **Location transparency**

  The users of the distributed systems are accessing the resources without knowing their physical location. This feature is called as location transparency.

For example www.google.co.in gives no idea where exactly the google server is placed, but the required information from the google's server can be obtained by typing the url.

Thus location transparency is achieved by assigning logical names to the resources. The logical names, in no sense indicates the physical location of the resource.

- **Migration transparency**

  Resources in the distributed environment are remote. Accessing methods only needs the logical names of the resources, physical location of the resources is immaterial.

  So the physical movement of the resources without affecting the way to access the resources is called as 'Migration transparency'.

- **Relocation transparency**

  The resources in the distributed environment are accessed even when they are relocated. This is an important feature and is called as relocation transparency.

  For example, if a user is working on his laptop and he has a wireless net connection. Though the person is constantly moving, still his machine is in the distributed environment on the network.

- **Replication transparency**

  Since the data is to be shared among various users, several copies of the data exists. As we have seen in the example of automatic banking that several copies of the data do exists. But the replicated data is unknown to the user as it is kept consistent by the distributed environment.
  The feature that the users are being unaware of the replicated copies of the data is called as replication transparency.

- **Failure transparency**

  There are multiple management points in the distributed environment. There is simultaneous access from several users. There may be multiple points of failures. The beauty of a good distributed system is to make the user unaware of the failure of any resource till the system is subsequently recovered from the failure. This feature is called as failure transparency.

- **Persistent transparency**

  This feature allows the user to be unaware of the fact that the server is moving state between primary and secondary memory. This movement of the server is needed is some object oriented databases where direct methods stored on objects can be invoked.

Degree of transparency

The transparency is not always desirable. Since the users are located in different continents they have a certain time difference. There may be situations where all the distribution aspects are not desired. If a person wants to read a particular newspaper everyday at 6:00 p.m. If he travels to the other part of the globe the same time settings will not be effective and hence the transparency aspect is not desirable in this case.

Similarly, it is not always possible to hide the failures. For example the data should remain consistent in a replication environment. Any change done to the data must be made to all the copies of the same data. This requires atleast some fraction of seconds and the user becomes aware of it.

For example, when we are viewing a live score on the internet, change in the score takes a considerable amount of time and the user knows that now the data is changing.

Thus there is a trade-off between high degree of transparency and the performance of the system. Second example may be a failure of the server. The user continuously tries. For example, www.abc.com but with no response indicating that abc's server has a breakdown.

Thus failure transparency may not always be exhibited in the distributed environment. So when designing a distributed system the goals of the distributed system must be achieved taking into account the performance measures of the system.

1.2.3 Support Openness

An open distributed system offer services according to standard rules that describe the syntax and semantics of these services.

The openness of a system is the characteristic that determine whether the system can be extended and reimplemented in various ways.

For example, to send and receive a message in a network, there are certain message formats which are to be followed. These formats are rules are being formalized in the protocols. So also, in a distributed system there is a Interface Definition Language (IDL).

The IDL specifies

- Names of the functions that are available.
- Return values of each function.
- Types of values or parameters to be passed to each function.
- Any exception to be raised on a specific condition etc.

Thus open systems are characterized by the fact that their key interfaces are published. They are based on the provision of a uniform communication mechanism and published interfaces for access to shared resources.

Advantages of open distributed system

- **Interoperability :** Open systems can work together. This feature allows two implementations of system or components from different manufacturers to co-exist and work together. Both of them rely on each others services or specified by a common standard.
- **Portability :** Portability is an ability to transform an application from one software or hardware platform to another.

Goals of open distributed systems

- **Flexibility :** Different developers provide different components. So a flexible open distributed system is one which allows easy configuration of the system with different components from different developers.
- **Extensibility :** Secondly a flexible open distributed system allows to add new components to the system, replaces the existing ones. This is done without affecting those components. Those are in their original place.

The flexibility is achieved by organizing the system as a collection of small and easily replacable components. There should also be a clear separation between policy and the mechanism used. For example in web caching, the user is provided with the facility of storing the documents. In a flexible open distributed system where web caching feature is used, the user should also be able to decide which documents are stored and for how long. Practically, this can be achieved by offering the users with a huge set of parameters which can be dynamically changed.

1.2.4 Be Scalable

An important goal of a distributed system is to be scalable. According to Neaman 1994, scalability is measured along three different dimensions.

- **Scalability with respect to size :** This means that in a scalable distributed system, users and resources can be easily added to the system.
- **Scalability with respect to geographic area :** In a scalable distributed system, the users and the resources may lie geographically at different locations. But still a communication between them is flexible and addition of new users and new resources is possible.

- **Scalability with respect to administration** : A distributed system with this feature allows easy and proper management of the system even if the system is widely split amongst amongst various organizations.

1.2.4.1 Problems with Scalability

There are various problems to be solved if a distributed system is to be properly scaled. Various types of problems arise in different types of scalability.

- **Problems in scalability with respect to size**

a) In a distributed environment many users are operating at the same time. If there is a single server, then the needs of all the users are to be satisfied by the single server i.e. the services are centralized. If many users are logging on to the server at the same time to get the service, there may be a blocking state to some of the users. Even though there is logically no limit on the number of users, but still providing the service may need communication, message transfers, data exchange etc. This may lead to delay for some users whereas a complete blocking state to the user who is last in the queue.

While designing an distributed system, this feature must be taken into account.

b) A highly secure data like bank accounts, credit card details must be confidentially maintained on the single server. If the number of users accessing the data from the server increases, more servers are designed. Replication of the same data breaks the confidential factor of the distributed system. The data may now be prune to many vulnerable attacks.

Thus keeping centralized service and centralized data both are a difficult task. If huge amount of data is to be stored, the capacity of the high ended servers becomes an issue. For example, electric bill of all the customers in India if decided to be centralized, how do 10000 million records be stored on a single server ?

c) In a distributed system, it is a better practise to adopt decentralized algorithms. This is because the machines in the distributed system are constantly communicating with each other sharing the data and resources. For example, It is decided to collect the load distribution on each machine. Since many messages are routed over various routes, a graph algorithm can be used to find the shortest path. But the algorithm itself would overload the network by passing messages, then processing them and finally making the decision. So it is a good practice to have decentralized algorithms. In a decentralized algorithm, no machine has complete information about the system state. There is no assumption of the existence of a global clock. If any machine feature, the algorithm is not ruined. In such an environment, the machines make decisions based only on local information.

- **Problems in scalability with respect to geographical area**

a) Geographical scalability says that communication between geographically apart users must be flexible. Current distributed systems support synchronous communication. In synchronous communication the client sends a request to the server and remains in the block state till the service is provided. This is shown in Fig. 1.3

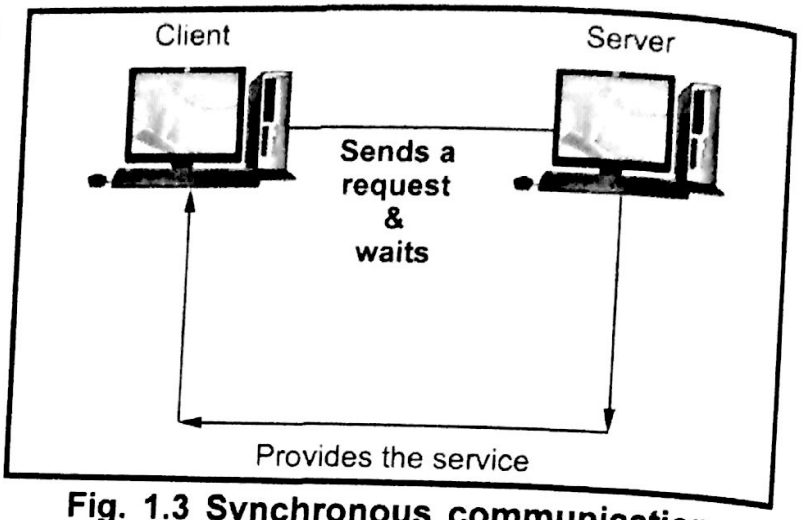

Fig. 1.3 Synchronous communication

This approach is fine in LANs but difficult in an distributed environment.

b) If a service is required by a particular user, then if it is a local area network a message could be broadcasted by enquiring about the name of the service provider but this is a difficult task in the distributed environment. This type of communication in distributed system will impose limitations in geographical scalability as the performance will be degraded and reliability will be on issue of concern.

- **Problems in scalability with respect to administration**

Scalability in an administrative domain is an difficult task to be achieved by any distributed system.

Components in an distributed environment does not necessarily reside within a single domain and hence leads to a difficulty in administration. If the components are within a single domain are trusted by the users of the same domain. Mostly preventive measures are already taken by the respective organization handling those components. However such measures cannot be adopted when the domain boundaries are crossed.

- Two types of security measures are required in case the domain boundaries are crossed.

1) Protection has to be done from malicious attacks from other domains.
2) New domain has to protect itself from the malicious attacks of the distributed systems.

Thus scalability with respect to administration is an important issue to be handled by an distributed system.

1.3 Examples of Distributed Systems

Distributed systems are basically networked. So the following examples are basically considered for distributed systems.

1. Intranet
2. Internet
3. Mobile computing

1.3.1 Intranet

Fig. 1.4 shows a typical intranet. The intranet may be composed of several local area networks.

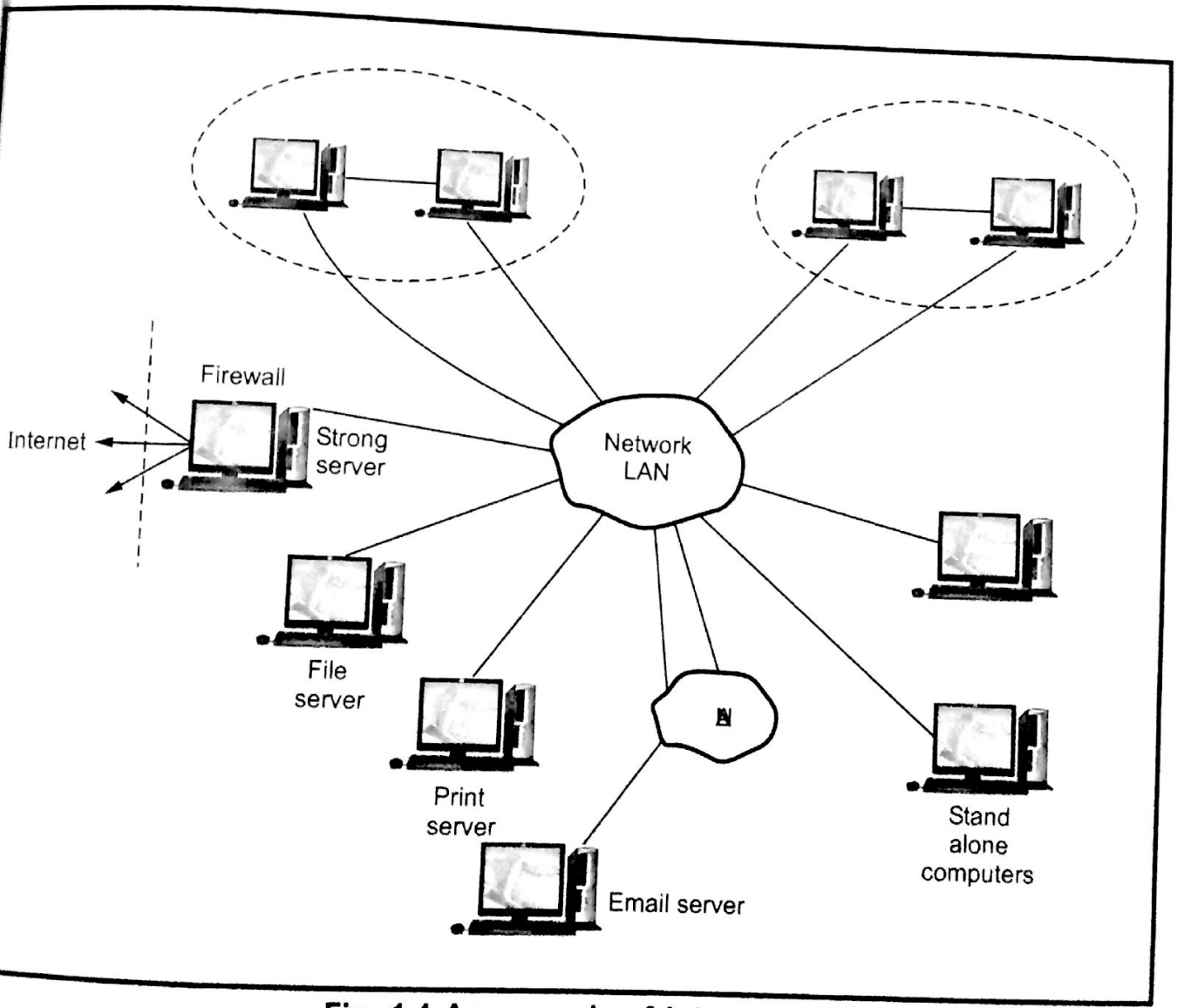

Fig. 1.4 An example of intranet

The capacity of the number of nodes or computers in an intranet depends on the organization setting it up. It may be limited to a single LAN on a single site or many LAN's internally connected together.

The intranet may be connected to the internet. There may be a strong server with a high end firewall to protect the intranet from any malicious scripts coming from the internet. There may be different servers in the intranet like the file server, print server, internal email server etc. These servers provide the respective services to the nodes on the intranet. For example the file servers allow the users to share the data.

A typical organization having intranet facility may be a college. The college compromises of various departments. Each department may have its own LAN setup.

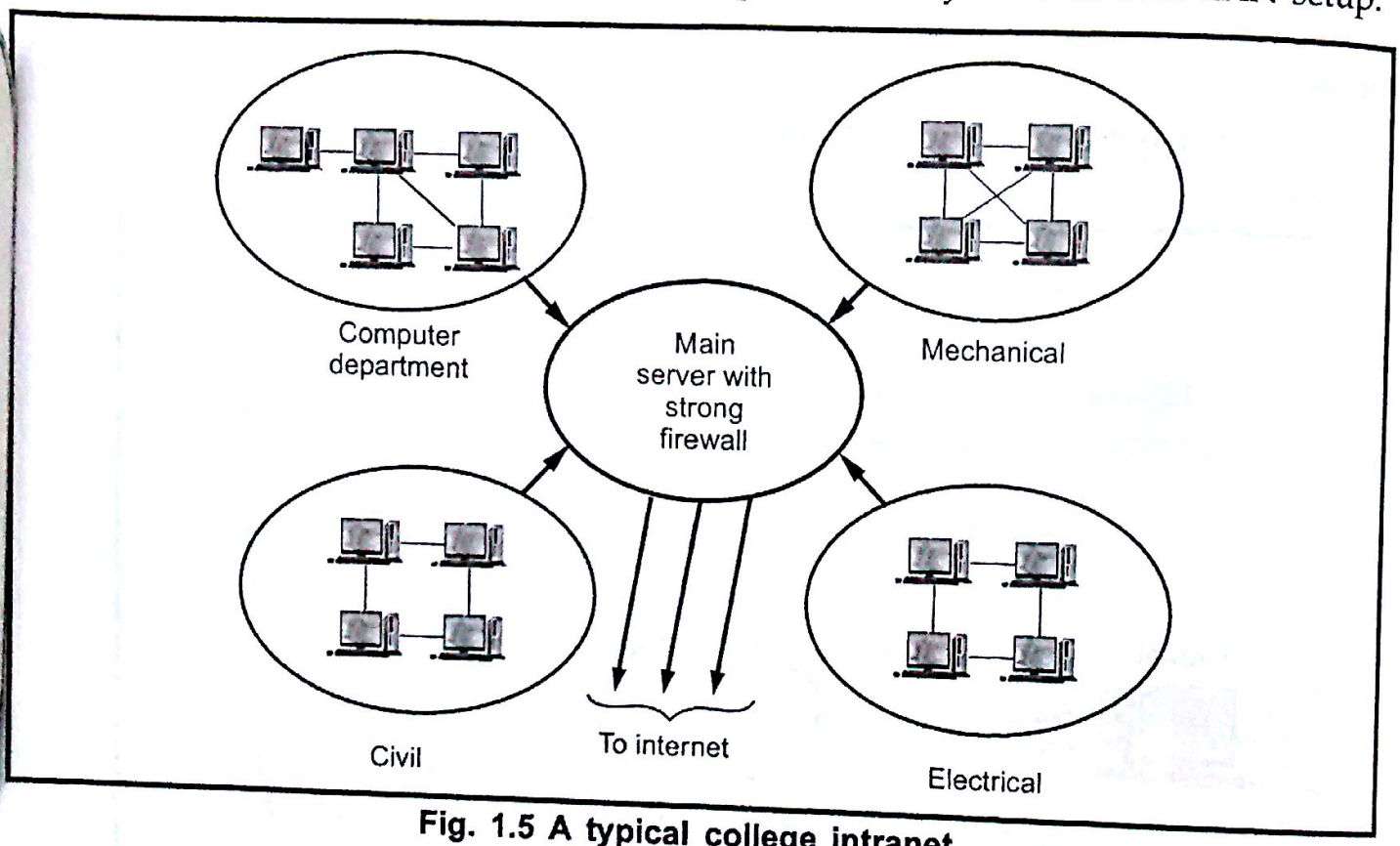

Fig. 1.5 A typical college intranet

Some organizations do not connect their internal networks to the internet. The example of such organizations are security agencies or military organizations.

The distributed system where intranets may be used are

- University computer network.
- Workflow information system of any organization.

1.3.2 Internet

In an internet different types of computer networks are interconnected. The internet itself is a very huge distributed system. Fig. 1.6 shows an typical portion of the internet.

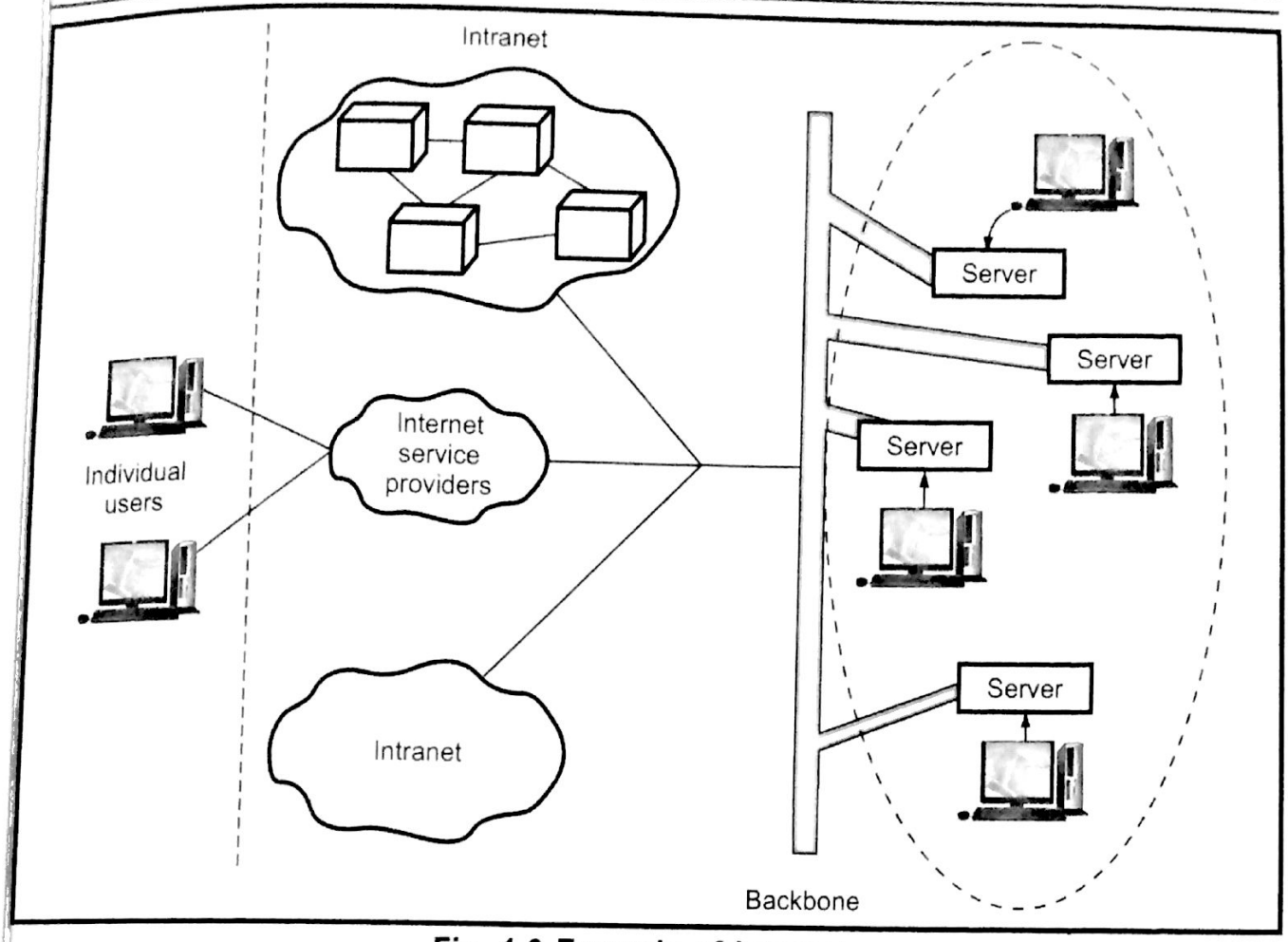

Fig. 1.6 Example of internet

Some common features of the internet are

- The individual users are connected to internet service providers. The internet service providers are companies that provide modem links and other internet facilities to the individual users.
- The intranets may be linked by a backbone. The backbone network provides high transmission capacities. This makes fast communication between various high end servers providing services and the individual nodes on the intranet.
- In an internet, the programs running on individual machines communicate with each other via different message passing mechanisms.
- The internet protocols enables the distributed feature allowing any program running anywhere on the network to address messages to programs anywhere else on the network.

1.3.3 Mobile Computing

This is a era of wireless networks. The devices enabling wireless features are

- Mobile phones.
- Handheld devices like video cameras and digital cameras.
- Laptops etc.

Mobile computing means users who are away from their working environment may get connected to their office intranet. This connection is wireless. This connection allows them to use the official resources at the place where they are using the wireless facility.

The user can connect to the internet using a

- Laptop which has connection to wireless LAN. The wireless LAN connects to internet.
- A mobile phone which connects to the WAP gateway. The WAP gateway connects to the internet.

Thus portable and handled devices are used in distributed computing.

Some more examples of distributed systems are

- WWW
- Air-traffic control
- Stock brokerage systems
- Banking
- Distributed supercomputing, simulation
- Distributed file systems/network storage/file sharing
- Streaming media distribution
- Event notification
- Gaming
- Embedded distributed systems (factory automation, vehicles).

1.4 Characteristics of Distributed Systems

The main characteristics of distributed system are

- **Multiple autonomous components (computers)**

  In a distributed system there are multiple components or computer. Each computer is an individual autonomous node having its own software and resources. It can also share components and resources from the distributed environment.

- **Components not shared by all users**

  Since a distributed system is a collection of autonomous computers linked by a computer network that appear to the users of the system at a single user, the user might not share some of the private resources in the distributed environment. So also, some of the resources may not be accessible by all the users.

- **Software architecture**

  In a distributed environment the software runs in concurrent processes on different processors. The software architecture and the operating system must be such that it necessarily supports this feature. Therefore the distributed systems must possibly have heterogeneous machines and networks or clusters of workstations or servers. Modern operating systems like UNIX or NT allows an easy construction of a basic clusters of workstations.

- **Multiple management points**

  The important characteristic of a distributed system is that there are multiple points of control.

  For example : Automatic banking.

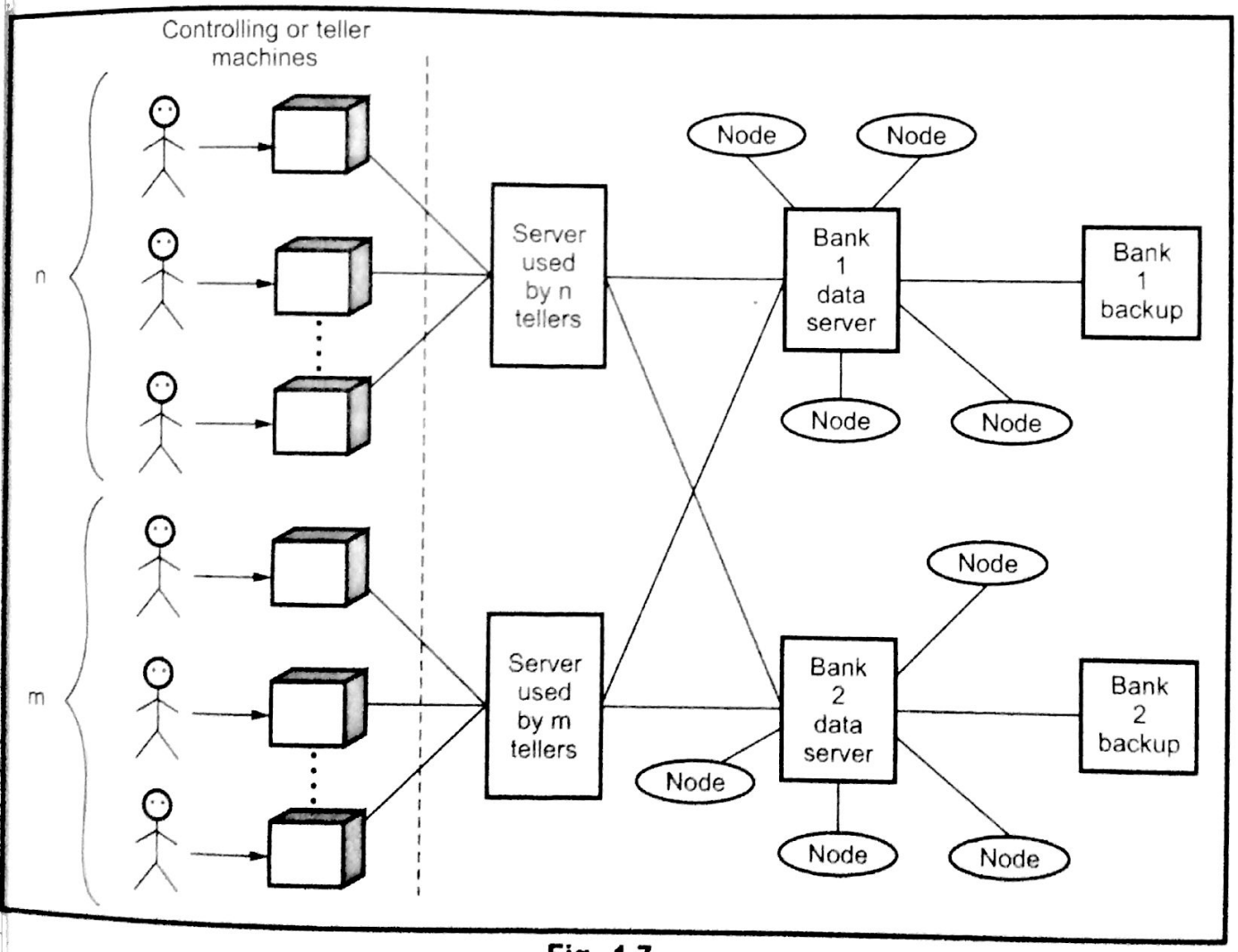

Fig. 1.7

In a distributed environment of automatic banking the primary requirements are security and reliability. Since there are multiple points of control, it is a challenge to maintain consistency of replicated data. Also, concurrent transactions which have operations involving accounts in different banks is an important feature.