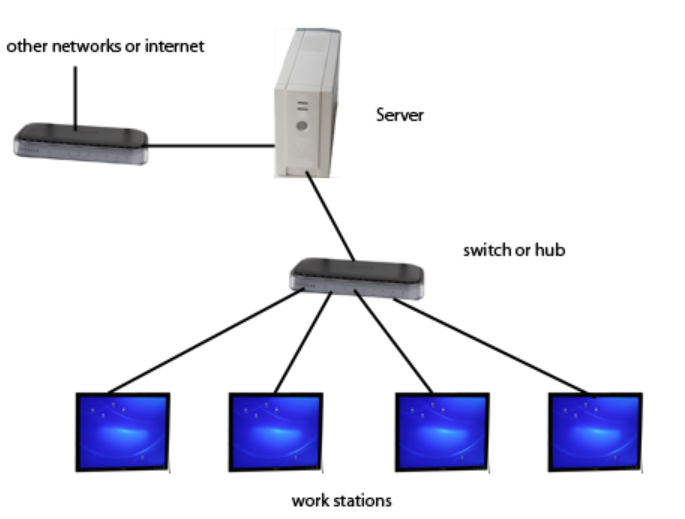
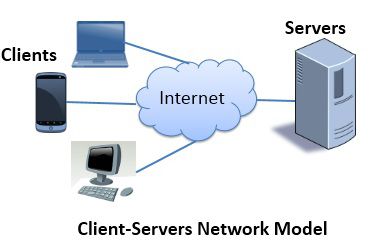
**CLIENT SERVER VS. PEER-TO-PEER COMPUTER NETWORK**

1. **Client-Server network**
   1. **Definitions**

* In the client/server model, all end systems are divided into clients and servers each designed for specific purposes. i.e. **End system** – a device that uses or provides end-user applications or network services E.g., a desktop PC, a Web server, a DNS server. They are labeled ”end systems” because they sit at the edge of a network. End systems that are connected to the Internet are also referred to as ”hosts”; this is because they host (run) Internet applications. End systems can be positioned on a network in different ways relative to each other I.e., they can be made to communicate and share resources according to different interaction models.
* **Clients** have an **active rol**e and initiate a communication session by sending requests to servers. Clients must have knowledge of the available servers and the services they provide. Clients can communicate with servers only; they cannot see each other.
* **Servers** have a **passive role** and respond to their clients by acting on each request and returning results. One server generally supports numerous clients.
* A client server network has a powerful computer known as a **server** which provides all of the services required on the network. These can include directing traffic around the network, managing disk drives and network peripherals such as printers and scanners.
* The '**client**' part of the network are the workstations that people log onto in order to use the network facilities.





* 1. **Hardware Role**
* The terms ”client” and ”server” usually refer to the primary roles played by networked hardware.
* A ”client” is usually something like a PC used by an individual, and primarily initiates conversations by sending requests
* A ”server” is usually a powerful machine dedicated to responding to client requests, sitting in a server room somewhere that nobody but its administrator ever see.
  1. **Software Role**
* TCP/IP uses different pieces of software for many protocols to implement ”client” and ”server” roles.
* Client software is usually found on client hardware and server software on server hardware, but not always
* Some devices may run both client and server software
  1. **Types of Server**
* The purpose of servers is to provide some predefined services for clients.
* There are two types of server.

|  |  |
| --- | --- |
| **Iterative servers** | **Concurrent servers** |
| * Iterative servers iterate through the following steps:  1. Wait for a client request to   arrive   1. Process the request and send the   response back to the client   1. Go back to Step 1  * Thus, iterative servers handle clients **sequentially,** finishing with one client before servicing the next * **Iterative design** is **quite simple** and is most **suitable for short-duration services** that exhibit relatively little variation in their execution time. * Internet services like echo (RFC 862) and daytime (RFC 867) are commonly implemented as iterative servers. * As a rule, **UDP-based servers** are **iterative**. | * Concurrent servers perform the following steps:   1. Wait for a client request to arrive  2. Use a new process/task/thread to  handle the request  3.Go back to Step 1   * Thus, concurrent servers handle client requests **in parallel.** * **Concurrent design** is more **complex** but yields **better performance.** * Internet services like HTTP, telnet, and FTP are commonly implemented as concurrent servers. * As a rule, **TCP-based servers** are **concurrent.** |

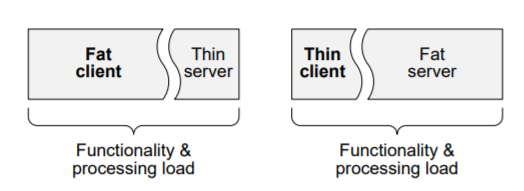
* Different types of Servers used in networks are listed below.

1. A **print server** which handles all of the print jobs required by network users.
2. An **email server** which provides a centralised email service for all network users
3. **File servers**. There may be several file servers to provide plenty of storage capacity
4. An **authentication server** which handles all log-in and security measures
5. A **web server** to provide an internal intranet service
   1. **Types of Clients**

* Clients are devices/programs that request services from servers.
* Clients (and, hence, servers) can be distinguished according to the functionality they provide and the amount of processing load they carry
* There are two types of clients.

1. Fat (i.e. thick or full)
2. Thin (i.e. slim or lean)

* **Fat clients** are devices/programs that are powerful enough and operate with limited dependence on their server counterparts.
* **Fat clients as devices** – a user workstation that is powerful and fully-featured in its own right. E.g., a desktop PC, a laptop, a netbook.
* **Fat clients as programs** – a client carries a relatively large proportion of the processing load. E.g., the Lineage II gaming client (more than 2 GB in size).
* **Thin clients** are devices/programs that have very limited functionality and depend heavily on their server counterparts.
* **Thin clients as devices** – a user workstation that contains a minimal operating system and little or no data storage. E.g., Sun Ray thin clients in Lintula, room TC215.
* **Thin clients as programs** – a client mainly provides a user interface, while the bulk of processing occurs in the server. E.g., the OnLive gaming client (about 10 MB in size).



* 1. **Advantages and Dis-Advantages of Client-Server Network**

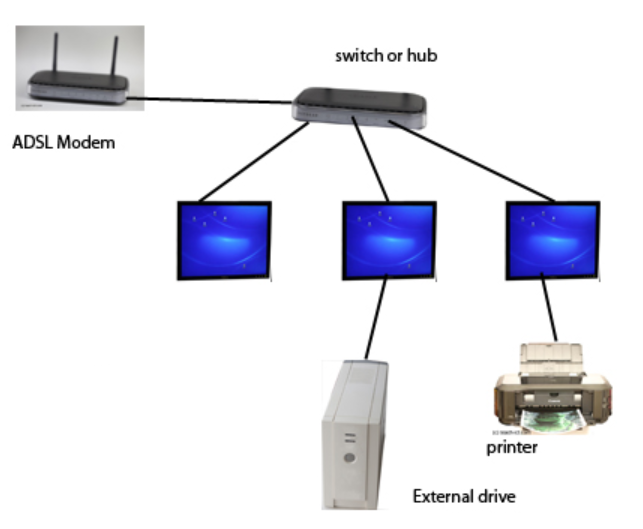
**Advantages**

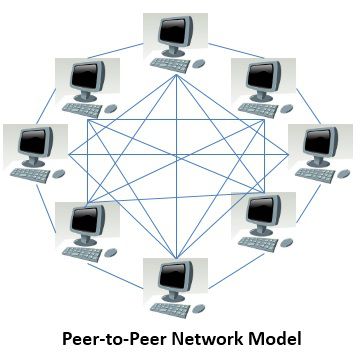
1. All files are stored in a central location.
2. Network peripherals are controlled centrally.
3. **Backups and network security is controlled centrally.** Security is a key advantage, as requests can be monitored and logged. Those with the necessary credentials will get access, so illegal or unauthorized data access can be prevented
4. Users can access shared data which is centrally controlled.

**Dis-Advantages**

1. A specialist network operating system is needed.
2. The server is expensive to purchase. More expensive in setup and maintenance due to specialized server equipment and software for running the server.
3. Specialist staff such as a network manager is needed.
4. If any part of the network fails a lot of disruption can occur.
5. Adding more and more clients, increases the workload of the server and the network speed could reduce. The bandwidth consumption also increases.
6. Client server model is not very robust, if the server fails, the network could collapse. And until the server is restored, the clients cannot function. Also if the server is the only source where data is stored, if the server falls, all such data is lost
7. **Peer-to-Peer Network**
   1. **Definition**

* In this configuration there is no central server. Instead, each workstation is loaded with the appropriate applications needed for that machine and local resources such as an attached printer folder or external hard drive can be shared.





* 1. **Advantages and Dis-Advantages of Peer-to-Peer Network**

**Advantages**

1. As a peer joins the network, it adds resources to the existing network, adding more members to the system, increases the capacity or resources of the system itself. The throughput of the network increases. Such networks also **scale better**, as increase in members **increases efficiency.**
2. Very **robust** as there is **no single point of failure**. If one peer fails, just that connection is lost, the network will go on functioning.
3. Since the **machines are independent of each other**, **operation and set up is easier** and **cheaper** than client-server model machines

**Dis-Advantages**

1. **P2P** networks have **high bandwidth consumption rates**, due to multiple request and responses taking place at the same time from different peers.
2. **Lack of security,** no checking of authentication takes place. So anyone can send and receive data from anybody
3. **Difference Between Client-Server and Peer-to-Peer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Parameter** | **Client-Server Networking** | **Peer-to-Peer Networking** |
| 1 | Physical Structure | There is a specific server and specific clients connected to the server.  Peers can be arranged in any network topology but in small networks, are located near to each other physically. The computers are similar in software content and protocols used for networking | Clients and server are not distinguished; each node act as client and server.  Physical structure is divided. Servers are powerful machines, designed for a dedicated purpose and should be robust to handle multiple transactions. Clients are ordinary workstations, accessed by different users. They have their own data |
| 2 | Service | The client request for service and server respond with the service. | Each node can request for services and can also provide the services. |
| 3 | Data | The data is stored in a centralized server. | Each peer has its own data. |
| 4 | Expense | The client-server are expensive to implement. | Peer-to-peer are less expensive to implement. |
| 5 | OS | Need a special operating system. | Can operate on a basic peer operating system |
| 6 | Security | Network access and security are centrally controlled, hence more secure. | Access to the network is not centrally controlled, hence less secure. |
| 7 | Focus | Sharing the information. | Connectivity. |
| 8 | Backup | Centralized on the server; managed by network administrator. Backup by device and media only required at server. | Decentralized; managed by users. Backup devices and media are required at each workstation. |
| 9 | Server down | If the server has a problem or the network connection fails then no one can do any work. This is why servers are designed to be reliable. | Here there is no dedicated server, so any peer goes down, there will be no problem. |
| 10 | User Roaming | People can log in to any client computer and their desktop, applications and data will be available as it is all stored on a remote server | User data is one one of the networked machines which must be on for access. User data must be in a 'share' folder if it is to be visible on the other machines |
|  | Explanation | Computer A is the server. Computers B, C and D are the clients. B wants to print a page. The printer is attached to Comp A. B will send a request to A, asking to print a page. A will print the page and respond to B. C wants to access a file, it will send a request to A, asking for the file. A will check C's credentials, C is not authorized to access the data, A will reject the request and respond to C by turning down its request | Computers A, B, C and D are connected in a P2P network. Comp A wants a file from Comp C, it sends a request to C. C decides to accept the request, finds the file and sends it to A. B and D are ignorant to what is going on but function normally. There is a network printer to which all computers are connected to. A sends a request to print and B sends one too. A's request reached first, so it is granted. Then the printer will grant B's request |

* 1. **Differences Between Client-Server and Peer-to-Peer network**
* The key difference between Client-Server and Peer-to-Peer network is that there is **a dedicated server and specific clients in the client-server network model** whereas, in **peer-to-peer each node can act as both server and client.**
* In the client-server model, the **server provides services to the client**. However, in peer-to-peer, **each peer can provide services and can also request for the services**.
* In the client-server model, **sharing information is more important** whereas, in peer-to-peer model **connectivity between peers is more important.**
* In the client-server model, **data is stored on a centralized server** whereas, in peer-to-peer each peer has its own data.
* In peer-to-peer model, the servers are distributed in a system, so there are fewer chances of server getting bottlenecked, but in the client-server model, there is a single server serving the clients, so there are more chances of server getting bottlenecked.
* The client-server model is more expensive to implement than peer-to-peer.
* The client-server model is more scalable and stable than peer-to-peer.

|  |
| --- |
| File Server: File servers are used to store the user documents and files centrally. An ideal file server should have a large amount of memory and storage space, fast hard-disks, multiple processors, fast network adapters, redundant power supplies etc.  A File server runs FTP (File Transfer Protocol) in Windows, Linux or Unix Networks, or SMBP (Server Message Block Protocol) in Windows Networks. Well known FTP software products are Micrsoft IIS, vsftpd, Apache FTP Server etc.  The main advantage of keeping network user files and electronic documents centrally in a file server is that the network user files and documents can be managed (backup'd) easily. Think about managing network user files and electronic documents kept distributed inside user workstations in a network consists of thousands of computers! Nearly impossible.  Print Server: Print Server, which redirects print jobs from client computers to specific printers.  Mail Server: Mail Servers are used to transmit emails using email protocols. Most widely used email transmission protocol is SMTP (Simple Mail Transfer Protocol). Mail Servers exchange emails between different domains.  Most widely used Mail Server software products are Microsoft Exchange Server, SENDMAIL (now proofpoint), qmail, Postfix etc.  Application Server: Common computer applications or programs which are required by different network users can be run in a central server, which enables multiple network users to access common network applications from the network. Typically Application Servers run business logic. Which means, every business is different and the Application Server is the Server Software which controls the business process. Some examples for Application Server Software are SAP BASIS, WebLogic, WebSphere etc.  Database Server: Database Server allows authorized network clients to create, view, modify and/or delete an organization's data, stored in a common database.  Examples of Database Management Systems are Oracle 10g/11g, Microsoft SQL Server 2000/2005/2008/2012, PostgreSQL, IBM DB2, MySQL, Sybase, Informix etc.  Directory Servers: Directory Servers allows the central administration and management of network users and network resources. Directory Servers provide the basic functions of network security, Authentication, Authorization and Accounting.  Examples of Directory Servers are Microsoft Active Directory, NetIQ eDirectory, Fedora Directory Server, OpenLDAP etc. |