# BACKUP AND RESTORE

# BACKING UP EXT2, EXT3, OR EXT4 FILE SYSTEMS

This procedure describes how to back up the content of an ext4, ext3, or ext2 file system into a file.

### Prerequisites

* If the system has been running for a long time, run the e2fsck utility on the partitions before backup:

# e2fsck /dev/*device*

**Procedure 5.1. Backing up ext2, ext3, or ext4 File Systems**

1. Back up configuration information, including the content of the /etc/fstab file and the output of the fdisk -l command. This is useful for restoring the partitions.

To capture this information, run the sosreport or sysreport utilities. For more information about sosreport, see the [What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later?](https://access.redhat.com/solutions/3592) Knowledge base article.

1. Depending on the role of the partition:
   * If the partition you are backing up is an operating system partition, boot your system into the rescue mode. See the [Booting to Rescue Mode](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/system_administrators_guide/index#sec-Booting_to_Rescue_Mode) section of the *System Administrator's Guide*.
   * When backing up a regular, data partition, unmount it.

Although it is possible to back up a data partition while it is mounted, the results of backing up a mounted data partition can be unpredictable.

If you need to back up a mounted file system using the dump utility, do so when the file system is not under a heavy load. The more activity is happening on the file system when backing up, the higher the risk of backup corruption is.

1. Use the dump utility to back up the content of the partitions:

# dump -0uf *backup-file* /dev/*device*

Replace *backup-file* with a path to a file where you want the to store the backup. Replace *device* with the name of the ext4 partition you want to back up. Make sure that you are saving the backup to a directory mounted on a different partition than the partition you are backing up.

**Example 5.2. Backing up Multiple ext4 Partitions**

To back up the content of the /dev/sda1, /dev/sda2, and /dev/sda3 partitions into backup files stored in the /backup-files/ directory, use the following commands:

# dump -0uf /backup-files/sda1.dump /dev/sda1

# dump -0uf /backup-files/sda2.dump /dev/sda2

# dump -0uf /backup-files/sda3.dump /dev/sda3

To do a remote backup, use the ssh utility or configure a password-less ssh login. For more information on ssh and password-less login, see the [Using the ssh Utility](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/s1-ssh-clients#s2-ssh-clients-ssh) and [Using Key-based Authentication](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/s1-ssh-configuration#s2-ssh-configuration-keypairs) sections of the *System Administrator's Guide*.

For example, when using ssh:

**Example 5.3. Performing a Remote Backup Using ssh**

# dump -0u -f - /dev/*device* | ssh root@*remoteserver.example.com* dd of=*backup-file*

# RESTORING EXT2, EXT3, OR EXT4 FILE SYSTEMS

This procedure describes how to restore an ext4, ext3, or ext2 file system from a file backup.

### Prerequisites

* You need a backup of partitions and their metadata, as described in [Section 5.4, “Backing up ext2, ext3, or ext4 File Systems”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ext4Backup).

**Procedure 5.2. Restoring ext2, ext3, or ext4 File Systems**

1. If you are restoring an operating system partition, boot your system into Rescue Mode. See the [Booting to Rescue Mode](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html-single/system_administrators_guide/index#sec-Booting_to_Rescue_Mode) section of the *System Administrator's Guide*.

This step is not required for ordinary data partitions.

1. Rebuild the partitions you want to restore by using the fdisk or parted utilites.

If the partitions no longer exist, recreate them. The new partitions must be large enough to contain the restored data. It is important to get the start and end numbers right; these are the starting and ending sector numbers of the partitions obtained from the fdisk utility when backing up.

For more information on modifying partitions, see [Chapter 13, *Partitions*](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ch-partitions)

1. Use the mkfs utility to format the destination partition:

# mkfs.ext4 /dev/*device*

**Important**

**Do not** format the partition that stores your backup files.

1. If you created new partitions, re-label all the partitions so they match their entries in the /etc/fstab file:

# e2label /dev/*device* *label*

1. Create temporary mount points and mount the partitions on them:
2. # mkdir /mnt/*device*

# mount -t ext4 /dev/*device* /mnt/*device*

1. Restore the data from backup on the mounted partition:
2. # cd /mnt/*device*

# restore -rf *device-backup-file*

If you want to restore on a remote machine or restore from a backup file that is stored on a remote host, you can use the ssh utility. For more information on ssh, see the [Using the ssh Utility](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/s1-ssh-clients#s2-ssh-clients-ssh) section of the *System Administrator's Guide*.

Note that you need to configure a password-less login for the following commands. For more information on setting up a password-less ssh login, see the [Using Key-based Authentication](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/s1-ssh-configuration#s2-ssh-configuration-keypairs) section of the *System Administrator's Guide*.

* + To restore a partition on a remote machine from a backup file stored on the same machine:

# ssh *remote-address* "cd /mnt/*device* && cat *backup-file* | /usr/sbin/restore -r -f -"

* + To restore a partition on a remote machine from a backup file stored on a different remote machine:

# ssh *remote-machine-1* "cd /mnt/*device* && RSH=/usr/bin/ssh /usr/sbin/restore -rf *remote-machine-2*:*backup-file*"

1. Reboot:

# systemctl reboot

**Example 5.4. Restoring Multiple ext4 Partitions**

To restore the /dev/sda1, /dev/sda2, and /dev/sda3 partitions from [Example 5.2, “Backing up Multiple ext4 Partitions”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ext4Backup#ext4Backup-Example):

1. Rebuild partitions you want to restore by using the fdisk command.
2. Format the destination partitions:
3. # mkfs.ext4 /dev/sda1
4. # mkfs.ext4 /dev/sda2

# mkfs.ext4 /dev/sda3

1. Re-label all the partitions so they match the /etc/fstab file:
2. # e2label /dev/sda1 Boot1
3. # e2label /dev/sda2 Root

# e2label /dev/sda3 Data

1. Prepare the working directories.

Mount the new partitions:

# mkdir /mnt/sda1

# mount -t ext4 /dev/sda1 /mnt/sda1

# mkdir /mnt/sda2

# mount -t ext4 /dev/sda2 /mnt/sda2

# mkdir /mnt/sda3

# mount -t ext4 /dev/sda3 /mnt/sda3

Mount the partition that contains backup files:

# mkdir /backup-files

# mount -t ext4 /dev/sda6 /backup-files

1. Restore the data from backup to the mounted partitions:
2. # cd /mnt/sda1
3. # restore -rf /backup-files/sda1.dump
4. # cd /mnt/sda2
5. # restore -rf /backup-files/sda2.dump
6. # cd /mnt/sda3

# restore -rf /backup-files/sda3.dump

1. Reboot:

# systemctl reboot

# OTHER EXT4 FILE SYSTEM UTILITIES

Red Hat Enterprise Linux 7 also features other utilities for managing ext4 file systems:

**e2fsck**

Used to repair an ext4 file system. This tool checks and repairs an ext4 file system more efficiently than ext3, thanks to updates in the ext4 disk structure.

**e2label**

Changes the label on an ext4 file system. This tool also works on ext2 and ext3 file systems.

**quota**

Controls and reports on disk space (blocks) and file (inode) usage by users and groups on an ext4 file system. For more information on using quota, refer to man quota and [Section 17.1, “Configuring Disk Quotas”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ch-disk-quotas#s1-disk-quotas-configuring).

**fsfreeze**

To suspend access to a file system, use the command # fsfreeze -f *mount-point* to freeze it and # fsfreeze -u *mount-point* to unfreeze it. This halts access to the file system and creates a stable image on disk.

**Note**

It is unnecessary to use fsfreeze for device-mapper drives.

For more information see the fsfreeze(8) manpage.

As demonstrated in [Section 5.2, “Mounting an ext4 File System”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ext4mount), the tune2fs utility can also adjust configurable file system parameters for ext2, ext3, and ext4 file systems. In addition, the following tools are also useful in debugging and analyzing ext4 file systems:

**debugfs**

Debugs ext2, ext3, or ext4 file systems.

**e2image**

Saves critical ext2, ext3, or ext4 file system metadata to a file.

# BTRFS (TECHNOLOGY PREVIEW)

**Note**

Btrfs is available as a Technology Preview feature in Red Hat Enterprise Linux 7 but has been deprecated since the Red Hat Enterprise Linux 7.4 release. It will be removed in a future major release of Red Hat Enterprise Linux.

For more information, see [Deprecated Functionality](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/7.4_Release_Notes/chap-Red_Hat_Enterprise_Linux-7.4_Release_Notes-Deprecated_Functionality.html) in the Red Hat Enterprise Linux 7.4 Release Notes.

Btrfs is a next generation Linux file system that offers advanced management, reliability, and scalability features. It is unique in offering snapshots, compression, and integrated device management.

## 6.1. Creating a btrfs File System

In order to make a basic btrfs file system, use the following command:

# mkfs.btrfs /*dev*/*device*

For more information on creating btrfs file systems with added devices and specifying multi-device profiles for metadata and data, refer to [Section 6.4, “Integrated Volume Management of Multiple Devices”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/btrfs-integrated_volume_management).

# NFS

**NFS** is an abbreviation of the **Network File System**. It is a protocol of a distributed file system. This protocol was developed by the [**Sun Microsystems**](https://www.javatpoint.com/sun-microsystems) in the year of 1984.

It is an architecture of the client/server, which contains a client program, server program, and a protocol that helps for communication between the client and server.

It is that protocol which allows the users to access the data and files remotely over the network. Any user can easily implement the NFS protocol because it is an open standard. Any user can manipulate files as same as if they were on like other protocols. This protocol is also built on the ONC RPC system.

This protocol is mainly implemented on those computing environments where the centralized management of resources and data is critical. It uses the [Transmission Control Protocol (TCP)](https://www.javatpoint.com/tcp) and [User Datagram Protocol (UDP)](https://www.javatpoint.com/udp-protocol) for accessing and delivering the data and files.

Network File System is a protocol that works on all the networks of IP-based. It is implemented in that client/server application in which the server of NFS manages the authorization, authentication, and clients. This protocol is used with Apple Mac OS, Unix, and Unix-like operating systems such as Solaris, [Linux](https://www.javatpoint.com/linux-tutorial), FreeBSD, AIX.

Difference Between NFS and CIFS

|  |  |  |
| --- | --- | --- |
| **NFS** | | **CIFS** |
| 1. NFS is an abbreviation of the Network File System. | 1. CIFS is an abbreviation of the Common Internet File system. | |
| 2. This protocol is used for sharing the files by Unix and Linux Operating systems. | 2. This protocol is used for sharing the files by Windows Operating systems. | |
| 3. It is highly scalable. | 3. It is low scalable. | |
| 4. The speed of communication is fast. | 4. The speed of communication is medium. | |
| 5. The network File system is not a secure protocol. | 5. Common Internet File System is more secure than the Network File System. | |
| 6. NFS is not a reliable protocol. | 6. CIFS is a reliable protocol. | |
| 7. This protocol does not provide the session. | 7. This protocol provides the sessions. | |
| 8. This protocol is easy to implement and set up. | 8. Its implementation is complex. | |
| 9. This protocol uses 111 port for both TCP and UDP. | 9. This protocol uses 139 and 445 TCP ports and 137 and 138 UDP ports. | |

# FTP

* FTP stands for File transfer protocol.
* FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
* It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
* It is also used for downloading the files to computer from other servers.

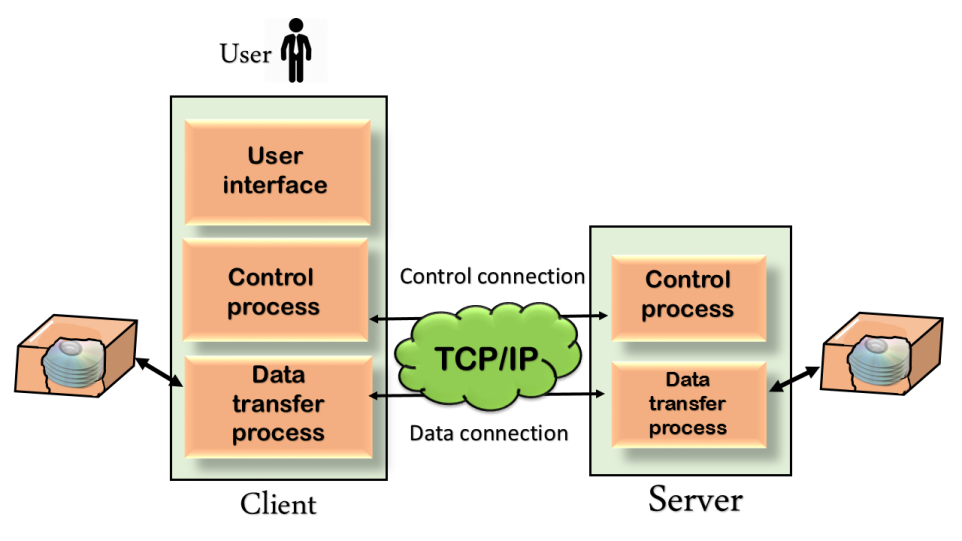
## Objectives of FTP

* It provides the sharing of files.
* It is used to encourage the use of remote computers.
* It transfers the data more reliably and efficiently.

Why FTP?

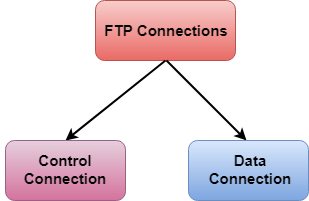
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

**There are two types of connections in FTP:**



* **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.
* **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

* FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
* It allows a user to connect to a remote host and upload or download the files.
* It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
* The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP:

* **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.
* **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
* **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.
* **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:

* The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
* FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
* Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
* It is not compatible with every system.

# Telnet

* The main task of the internet is to provide services to users. For example, users want to run different application programs at the remote site and transfers a result to the local site. This requires a client-server program such as FTP, SMTP. But this would not allow us to create a specific program for each demand.
* The better solution is to provide a general client-server program that lets the user access any application program on a remote computer. Therefore, a program that allows a user to log on to a remote computer. A popular client-server program Telnet is used to meet such demands. Telnet is an abbreviation for **Terminal Network**.
* Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

## There are two types of login:

### Local Login

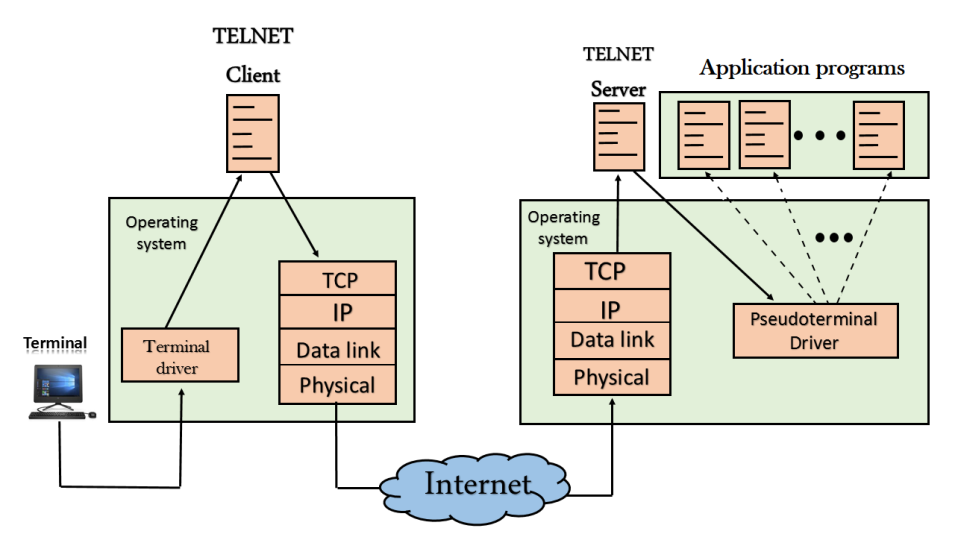
### Computer Network Telnet

### When a user logs into a local computer, then it is known as local login.

### When the workstation running terminal emulator, the keystrokes entered by the user are accepted by the terminal driver. The terminal driver then passes these characters to the operating system which in turn, invokes the desired application program.

### However, the operating system has special meaning to special characters. For example, in UNIX some combination of characters have special meanings such as control character with "z" means suspend. Such situations do not create any problem as the terminal driver knows the meaning of such characters. But, it can cause the problems in remote login.

### Remote login



* + When the user wants to access an application program on a remote computer, then the user must perform remote login.

## How remote login occurs

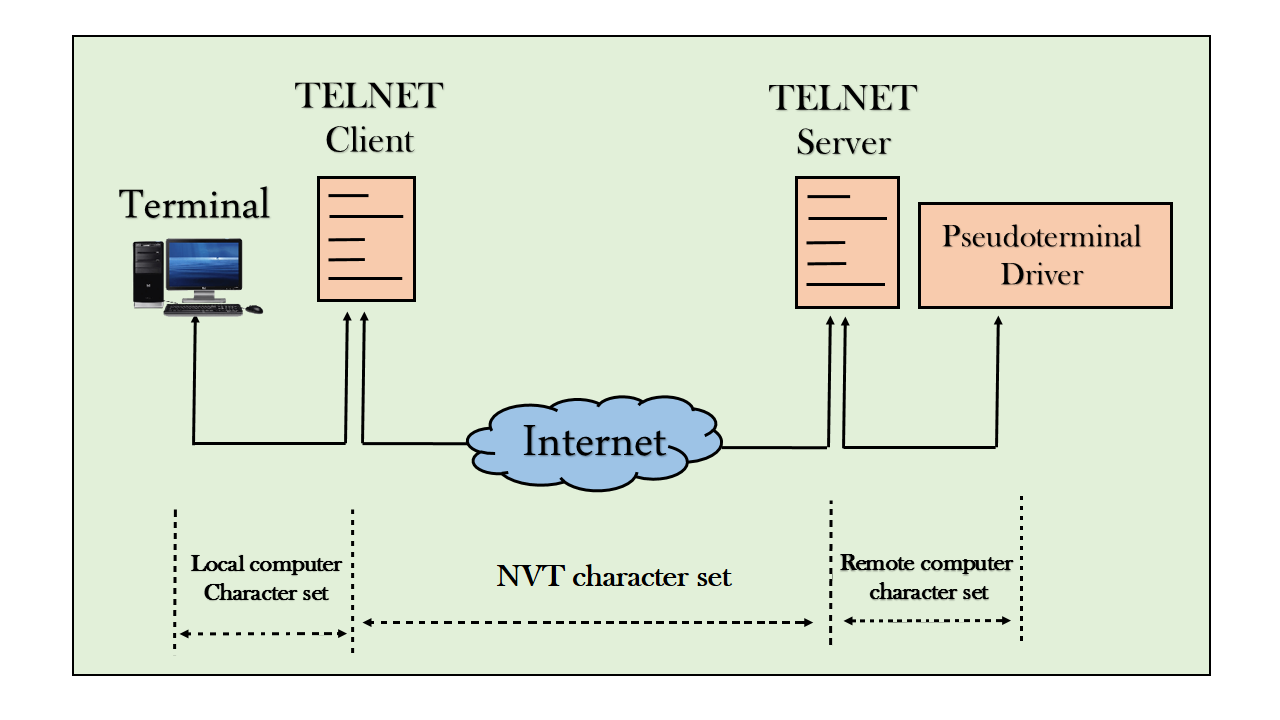
### At the local site

The user sends the keystrokes to the terminal driver, the characters are then sent to the TELNET client. The TELNET client which in turn, transforms the characters to a universal character set known as network virtual terminal characters and delivers them to the local TCP/IP stack

### At the remote site

The commands in NVT forms are transmitted to the TCP/IP at the remote machine. Here, the characters are delivered to the operating system and then pass to the TELNET server. The TELNET server transforms the characters which can be understandable by a remote computer. However, the characters cannot be directly passed to the operating system as a remote operating system does not receive the characters from the TELNET server. Therefore it requires some piece of software that can accept the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

## Network Virtual Terminal (NVT)

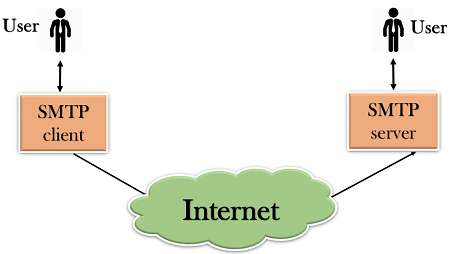


* + The network virtual terminal is an interface that defines how data and commands are sent across the network.
  + In today's world, systems are heterogeneous. For example, the operating system accepts a special combination of characters such as end-of-file token running a DOS operating system *ctrl+z* while the token running a UNIX operating system is *ctrl+d*.
  + TELNET solves this issue by defining a universal interface known as network virtual interface.
  + The TELNET client translates the characters that come from the local terminal into NVT form and then delivers them to the network. The Telnet server then translates the data from NVT form into a form which can be understandable by a remote computer.

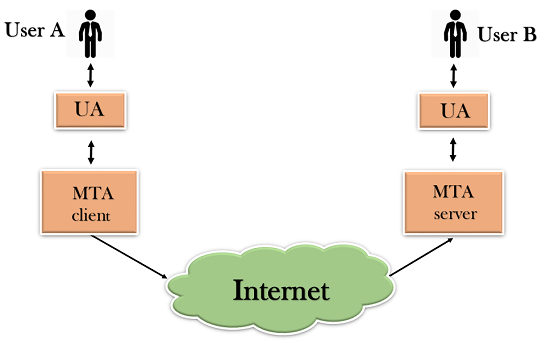
# SMTP

* SMTP stands for Simple Mail Transfer Protocol.
* SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called **Simple Mail Transfer Protocol**.
* It is a program used for sending messages to other computer users based on e-mail addresses.
* It provides a mail exchange between users on the same or different computers, and it also supports:
  + It can send a single message to one or more recipients.
  + Sending message can include text, voice, video or graphics.
  + It can also send the messages on networks outside the internet.
* The main purpose of SMTP is used to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address. For example, if the recipient address is wrong, then receiving server reply with an error message of some kind.

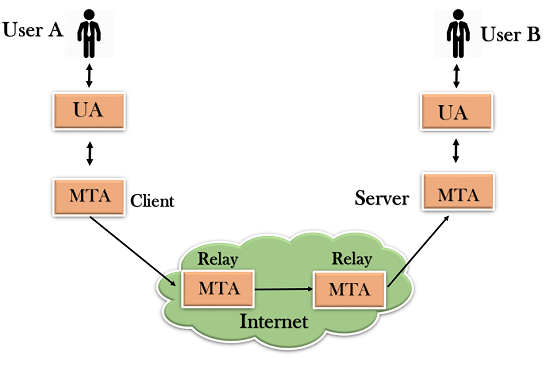
## Components of SMTP



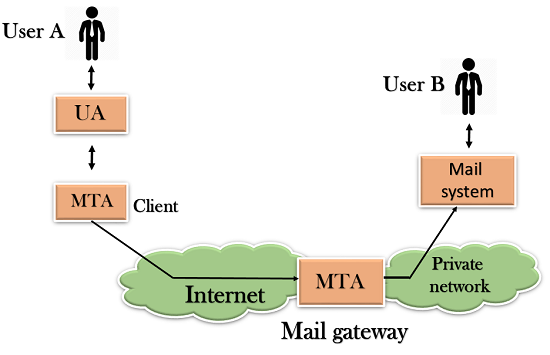
* First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.



* SMTP allows a more complex system by adding a relaying system. Instead of just having one MTA at sending side and one at receiving side, more MTAs can be added, acting either as a client or server to relay the email.



* The relaying system without TCP/IP protocol can also be used to send the emails to users, and this is achieved by the use of the mail gateway. The mail gateway is a relay MTA that can be used to receive an email.



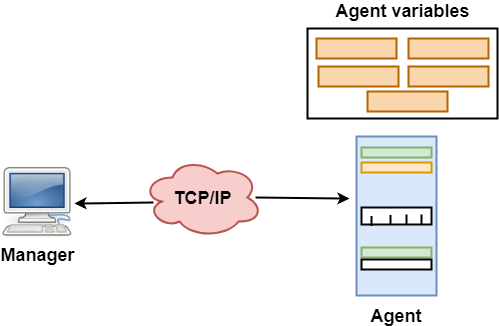
## Working of SMTP

1. **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and header is like an envelope that contains the recipient's address.
2. **Submission of Mail:** After composing an email, the mail client then submits the completed e-mail to the SMTP server by using SMTP on TCP port 25.
3. **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name. For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name.  
   If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.
4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

# SNMP

* SNMP stands for **Simple Network Management Protocol**.
* SNMP is a framework used for managing devices on the internet.
* It provides a set of operations for monitoring and managing the internet.

## SNMP Concept



* SNMP has two components Manager and agent.
* The manager is a host that controls and monitors a set of agents such as routers.
* It is an application layer protocol in which a few manager stations can handle a set of agents.
* The protocol designed at the application level can monitor the devices made by different manufacturers and installed on different physical networks.
* It is used in a heterogeneous network made of different LANs and WANs connected by routers or gateways.

## Managers & Agents

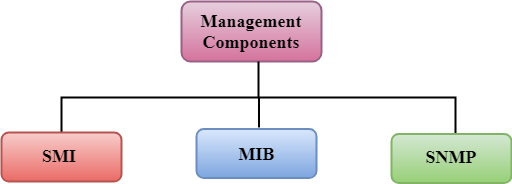
* A manager is a host that runs the SNMP client program while the agent is a router that runs the SNMP server program.
* Management of the internet is achieved through simple interaction between a manager and agent.
* The agent is used to keep the information in a database while the manager is used to access the values in the database. For example, a router can store the appropriate variables such as a number of packets received and forwarded while the manager can compare these variables to determine whether the router is congested or not.
* Agents can also contribute to the management process. A server program on the agent checks the environment, if something goes wrong, the agent sends a warning message to the manager.

## Management with SNMP has three basic ideas:

* A manager checks the agent by requesting the information that reflects the behavior of the agent.
* A manager also forces the agent to perform a certain function by resetting values in the agent database.
* An agent also contributes to the management process by warning the manager regarding an unusual condition.

## Management Components

* Management is not achieved only through the SNMP protocol but also the use of other protocols that can cooperate with the SNMP protocol. Management is achieved through the use of the other two protocols: SMI (Structure of management information) and MIB(management information base).
* Management is a combination of SMI, MIB, and SNMP. All these three protocols such as abstract syntax notation 1 (ASN.1) and basic encoding rules (BER).

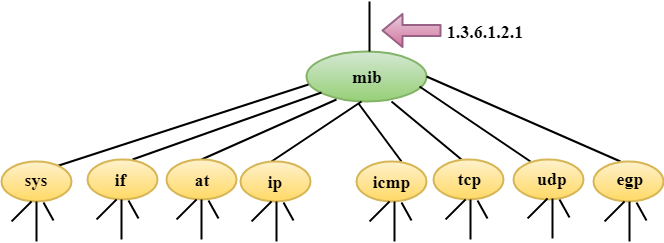


### SMI

The SMI (Structure of management information) is a component used in network management. Its main function is to define the type of data that can be stored in an object and to show how to encode the data for the transmission over a network.

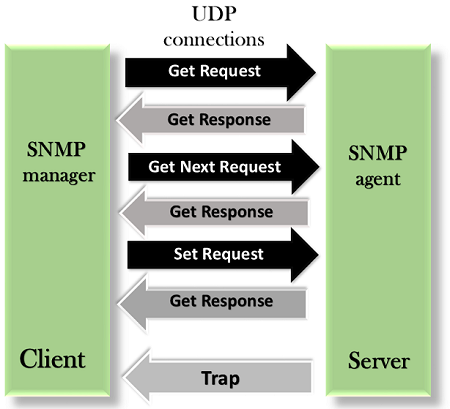
### MIB

* The MIB (Management information base) is a second component for the network management.
* Each agent has its own MIB, which is a collection of all the objects that the manager can manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.



### SNMP

SNMP defines five types of messages: GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap.



**GetRequest:** The GetRequest message is sent from a manager (client) to the agent (server) to retrieve the value of a variable.

**GetNextRequest:** The GetNextRequest message is sent from the manager to agent to retrieve the value of a variable. This type of message is used to retrieve the values of the entries in a table. If the manager does not know the indexes of the entries, then it will not be able to retrieve the values. In such situations, GetNextRequest message is used to define an object.

**GetResponse:** The GetResponse message is sent from an agent to the manager in response to the GetRequest and GetNextRequest message. This message contains the value of a variable requested by the manager.

**SetRequest:** The SetRequest message is sent from a manager to the agent to set a value in a variable.

**Trap:** The Trap message is sent from an agent to the manager to report an event. For example, if the agent is rebooted, then it informs the manager as well as sends the time of rebooting.

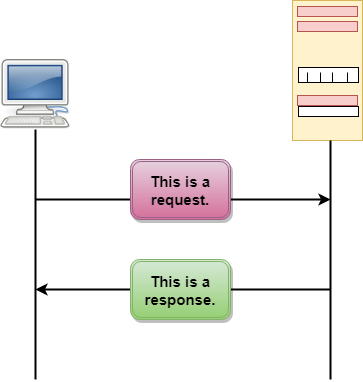
# HTTP

* HTTP stands for **HyperText Transfer Protocol**.
* It is a protocol used to access the data on the World Wide Web (www).
* The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
* This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
* HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
* HTTP is used to carry the data in the form of MIME-like format.
* HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

## Features of HTTP:

* **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exist only during the current request and response time only.
* **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.
* **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

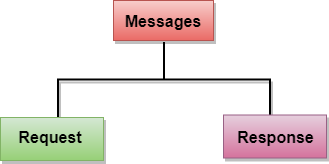
## HTTP Transactions



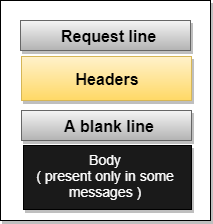
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

## Messages

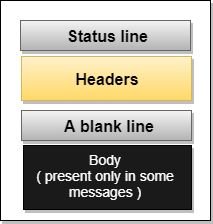
HTTP messages are of two types: request and response. Both the message types follow the same message format.



**Request Message:** The request message is sent by the client that consists of a request line, headers, and sometimes a body.



**Response Message:** The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



## Uniform Resource Locator (URL)

* A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
* The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
* The URL defines four parts: method, host computer, port, and path.



* **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
* **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
* **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
* **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files.

# Computer Network Security

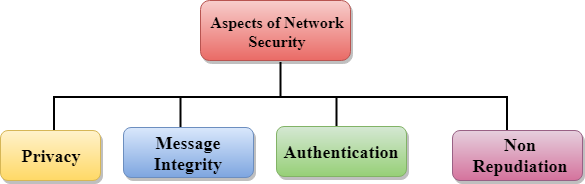
Computer network security consists of measures taken by business or some organizations to monitor and prevent unauthorized access from the outside attackers.

Different approaches to computer network security management have different requirements depending on the size of the computer network. For example, a home office requires basic network security while large businesses require high maintenance to prevent the network from malicious attacks.

Network Administrator controls access to the data and software on the network. A network administrator assigns the user ID and password to the authorized person.

## Aspects of Network Security:

Following are the desirable properties to achieve secure communication:



* **Privacy:** Privacy means both the sender and the receiver expects confidentiality. The transmitted message should be sent only to the intended receiver while the message should be opaque for other users. Only the sender and receiver should be able to understand the transmitted message as eavesdroppers can intercept the message. Therefore, there is a requirement to encrypt the message so that the message cannot be intercepted. This aspect of confidentiality is commonly used to achieve secure communication.
* **Message Integrity:** Data integrity means that the data must arrive at the receiver exactly as it was sent. There must be no changes in the data content during transmission, either maliciously or accident, in a transit. As there are more and more monetary exchanges over the internet, data integrity is more crucial. The data integrity must be preserved for secure communication.
* **End-point authentication:** Authentication means that the receiver is sure of the sender?s identity, i.e., no imposter has sent the message.
* **Non-Repudiation:** Non-Repudiation means that the receiver must be able to prove that the received message has come from a specific sender. The sender must not deny sending a message that he or she send. The burden of proving the identity comes on the receiver. For example, if a customer sends a request to transfer the money from one account to another account, then the bank must have a proof that the customer has requested for the transaction.