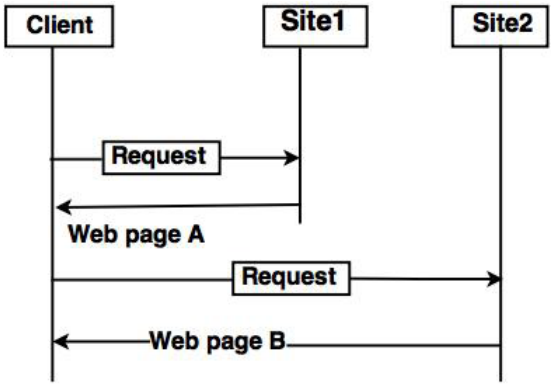
UNIT – 5

APPLICATION LAYER

WWW and HTTP – FTP – Email –Telnet –SSH – DNS – SNMP.

5.0 WWW

- The World Wide Web (WWW) is a collection of documents and other web resources which are identified by URLs, interlinked by hypertext links, and can be accessed and searched by browsers via the Internet.
- World Wide Web is also called the Web and it was invented by Tim Berners-Lee in 1989.
- Website is a collection of web pages belonging to a particular organization.
- The pages can be retrieved and viewed by using browser.



Architecture of WWW

- The client wants to see some information that belongs to site 1.
- It sends a request through its browser to the server at site 2.
- The server at site 1 finds the document and sends it to the client.

Client (Browser):

- Web browser is a program, which is used to communicate with web server on the Internet.

- Each browser consists of three parts: a controller, client protocol and interpreter.
- The controller receives input from input device and use the programs to access the documents.
- After accessing the document, the controller uses one of the interpreters to display the document on the screen.

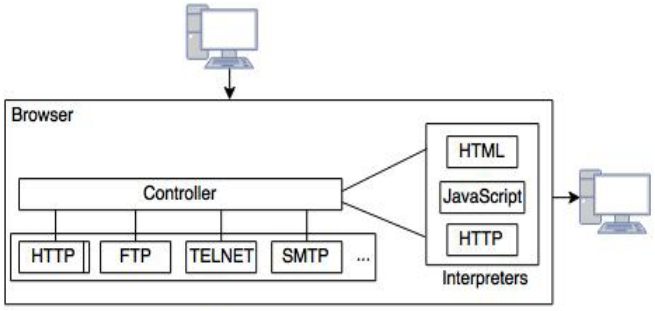


Fig: Cllient (Browser)

Server:

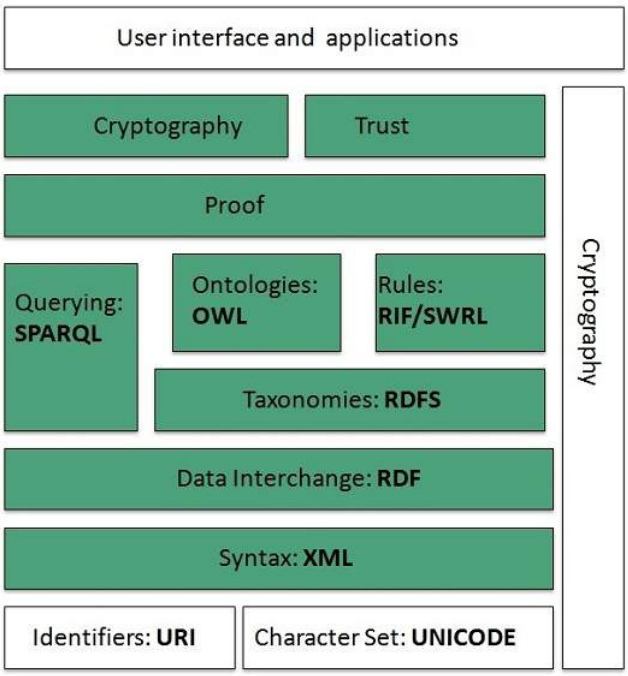
- A computer which is available for the network resources and provides service to the other computer on request is known as server.
- The web pages are stored at the server.
- Server accepts a TCP connection from a client browser.
- It gets the name of the file required.
- Server gets the stored file. Returns the file to the client and releases the top connection.

Uniform Resource Locater (URL):

- The URL is a standard for specifying any kind of information on the Internet.
- The URL consists of four parts: protocol, host computer, port and path.
- The protocol is the client or server program which is used to retrieve the document or file. The protocol can be ftp or http.
- The host is the name of computer on which the information is located.
- The URL can optionally contain the port number and it is separated from the host name by a colon.
- Path is the pathname of the file where the file is stored.

WWW ARCHITECTURE

WWW architecture is divided into several layers as shown in the following diagram:



Identifiers and Character Set

Uniform Resource Identifier (URI) is used to uniquely identify resources on the web and UNICODE makes it possible to built web pages that can be read and write in human languages.

Syntax

XML (Extensible Markup Language) helps to define common syntax in semantic web.

Data Interchange

Resource Description Framework (RDF) framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

RDF Schema (RDFS) allows more standardized description of taxonomies and other ontological constructs.

Ontologies

Web Ontology Language (OWL) offers more constructs over RDFS. It comes in following three versions:

OWL Lite for taxonomies and simple constraints.

OWL DL for full description logic support.

OWL for more syntactic freedom of RDF

Rules

RIF and SWRL offers rules beyond the constructs that are available from RDFs and OWL. Simple Protocol and RDF Query Language (SPARQL) is SQL like language used for querying RDF data and OWL Ontologies.

Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

Cryptography means such as digital signature for verification of the origin of sources is used.

User Interface and Applications

On the top of layer User interface and Applications layer is built for user interaction.

5.1 HYPER TEXT TRANSFER PROTOCOL (HTTP)

HTTP is a communication protocol. It defines mechanism for communication between browser and the web server. It is also called request and response protocol because the communication between browser and server takes place in request and response pairs.

- The World Wide Web has been so successful and has made the Internet accessible to so many people that sometimes it seems to be synonymous with the Internet.
- One helpful way to think of the Web is as a set of cooperating clients and servers, all of whom speak the same language: HTTP.
- Most people are exposed to the Web through a graphical client program, or Web browser, like Netscape or Explorer.
- Any Web browser has a function that allows the user to "open a URL."
- URLs (uniform resource locators) provide information about the location of objects on the Web; they look like the following:

  http://www.act.edu.in/index.html
- If you opened that particular URL, your Web browser would open a TCP connection to the Web server at a machine called

  www.act.edu.in

  and immediately retrieve and display the file called index.html.
- Most files on the Web contain images and text, and some have audio and video clips.
- They also include URLs that point to other files, and your Web browser will have some way in which you can recognize URLs and ask the browser to open them. These embedded URLs are called *hypertext links*.

- When you ask your Web browser to open one of these embedded URLs (e.g., by pointing and clicking on it with a mouse), it will open a new connection and retrieve and display a new file. This is called "following a link
- When you select to view a page, your browser (the client) fetches the page from the server using HTTP running over TCP.
- Like SMTP,HTTP is a text-oriented protocol. At its core, each HTTP message has the general form

  START_LINE <CRLF>

  MESSAGE_HEADER <CRLF>

  <CRLF> MESSAGE_BODY <CRLF>

  where as before, <CRLF> stands for carriage-return-line-feed.

HTTP Request

HTTP request comprises of lines which contains:

- Request line.
- Header Fields.
- Message body.

Key Points

- The first line i.e. the **Request line** specifies the request method i.e. **Get** or **Post.**
- The second line specifies the header which indicates the domain name of the server from where index.htm is retrieved.

HTTP Response

Like HTTP request, HTTP response also has certain structure. HTTP response contains:

- Status line.
- Headers.
- Message body.
  - The first line (START LINE) indicates whether this is a request message or a response message.
  - In effect, it identifies the "remote procedure" to be executed (in the case of a request message) or the "status" of the request (in the case of a response message).
  - The next set of lines specifies a collection of options and parameters that qualify the request or response.
  - There are zero or more of these MESSAGE HEADER lines—the set is terminated by a blank line—each of which looks like a header line in an email message.
- HTTP defines many possible header types, some of which pertain to request messages, some to response messages, and some to the data carried in the message body.

- after the blank line comes the contents of the requested message (MESSAGE BODY); this part of the message is typically empty for request messages.

Request Messages

- The first line of an HTTP request message specifies three things:
  - the operation to be performed,
  - the Web page the operation should be performed on, and
  - the version of HTTP being used.
- Although HTTP defines a wide assortment of possible request operations—including "write" operations that allow a Web page to be posted on a server—
- the two most common operations are
  - GET (fetch the specified Web page) and
  - HEAD (fetch status information about the specified Web page).
- The former is obviously used when your browser wants to retrieve and display a Web page.
- The latter is used to test the validity of a hypertext link or to see if a particular page has been modified since the browser last fetched it
- The full set of operations is summarized in Table 9.1

Operation	Description
OPTIONS	request information about available options
GET	retrieve document identified in URL
HEAD	retrieve metainformation about document identified in URL
POST	give information (e.g., annotation) to server
PUT	store document under specified URL
DELETE	delete specified URL
TRACE	loopback request message
CONNECT	for use by proxies

Fig 5.1 HTTP request message

Response Messages

- Like request messages, response messages begin with a single START LINE.
- The line specifies the version of HTTP being used, a three-digit code indicating whether or not the request was successful, and a text string giving the reason for the response.
- For example, the START LINE **HTTP/1.1 202 Accepted** indicates that the server was able to satisfy the request,

- While **HTTP/1.1 404 Not Found** indicates that it was not able to satisfy the request because the page was not found

Code	Type	Example Reasons
1xx	Informational	request received, continuing process
2xx	Success	action successfully received, understood, and accepted
3xx	Redirection	further action must be taken to complete the request
4xx	Client Error	request contains bad syntax or cannot be fulfilled
5xx	Server Error	server failed to fulfill an apparently valid request

Fig 5.2 five types of http result codes

- There are five general types of response codes, with the first digit of the code indicating its type.

Uniform Resource Identifiers

- The URLs that HTTP uses as addresses are one type of uniform resource identifier (URI)
- A URI is a character string that identifies a resource, where a resource can be anything that has identity, such as a document, an image, or a service.
- The format of URIs allows various more-specialized kinds of resource identifiers to be in corporated into the URI space of identifiers.
- The first part of the URI is a scheme that names a particular way of identifying a certain kind of resource, such a **mailto** for email addresses or **file** for the file names.
- The second part, specified from the first part by a colon, is the ***scheme-specific part***. It is a resource identifier consistent with the scheme in the first part , as in the URIs.

  mailto:mail@mydomain.org

  and

  file:///c:/foo/html

- A resource doesn't have to be retrievable or accessible.
- Even human beings and corporations can be resources.
- A more concrete example is the **mid** scheme for message IDs. Hence, URIs are not always some kind of address for locating the resource; they can be purely identifier.
- a particular URI qualifies as a URL only if it is intended to be used to locate the resource.
- Even if a particular URI appears to be an address, such as a URI that uses the **http** scheme,the URI is not considered a URL unless it is *intended* to be used to locate the resource.
- For example, XML namespaces are identified by URIs that use the **http** scheme but are nonetheless not URLs since there is no requirement that the URI give the location of any resource related to the namespace.

TCP Connections

- The original version of HTTP (1.0) established a separate TCP connection for each data item retrieved from the server.
- Retrieving a page that included some text and a dozen icons or other small graphics would result in 13 separate TCP connections being established and closed.
- The most important improvement in the latest version of HTTP (1.1) is to allow *persistent connections*—the client and server can exchange multiple request/response messages over the same TCP connection.
- Persistent connections have two advantages.
    - First, they obviously eliminate the connection setup overhead, thereby reducing the load on the server, the load on the network caused by the additional TCP packets, and the delay perceived by the user.
    - Second, because a client can send multiple request messages down a single TCP connection, TCP's congestion window mechanism is able to operate more efficiently. This is because it's not necessary to go through the slow start phase for each page.
- Persistent connections do not come without a price, however. The problem is that neither the client nor server necessarily knows how long to keep a particular TCP connection opens. This is especially critical on the server, which might be asked to keep connections open on behalf of thousands of clients.
- The solution is that the server must time out and closes a connection if it has received no requests on the connection for a period of time. Also, both the client and server must watch to see if the other side has elected to close the connection, and they must use that information as a signal that they should close their side of the connection as well.

Caching

- One of the most active areas of research (and entrepreneurship) in the Internet today is how to effectively cache Web pages.
- Caching has many benefits.
    - From the client's perspective, a page that can be retrieved from a nearby cache can be displayed much more quickly than if it has to be fetched from across the world.
    - From the server's perspective, having a cache intercept and satisfy a request reduces the load on the server.
- Caching can be implemented in many different places.
    - For example, a user's browser can cache recently accessed pages, and simply display the cached copy if the user visits the same page again.
    - As another example, a site can support a single site wide cache. This allows users to take advantage of pages previously downloaded by other users.
- Closer to the middle of the Internet, ISPs can cache pages. Note that in the second case, the users within the site most likely know what machine is caching pages on behalf of the site, and they configure their browsers to connect directly to the caching host.

This node is sometimes called a *proxy*. In contrast, the sites that connect to the ISP are probably not aware that the ISP is caching pages. It simply happens to be the case that HTTP requests coming out of the various sites pass through a common ISP router. This router can peek inside the request message and look at the URL for the requested page. If it has the page in its cache, it returns it. If not, it forwards the request to the server and watches for the response to fly by in the other direction. When it does, the router saves a copy in the hope that it can use it to satisfy a future request.

- No matter where pages are cached, the ability to cache Web pages is important enough that HTTP has been designed to make the job easier. The trick is that the cache needs to make sure it is not responding with an out-of-date version of the page.
- For example, the server assigns an expiration date (the Expires header field) to each page it sends back to the client (or to a cache between the server and client). The cache remembers this date and knows that it need not revivify the page each time it is requested until after that expiration date has passed.
- After that time (or if that header field is not set) the cache can use the HEAD or conditional GET operation (GET with If-Modified-Since header line) to verify that it has the most recent copy of the page.
- More generally, there is a set of "cache directives" that must be obeyed by all caching mechanisms along the request/response chain. These directives specify whether or not a document can be cached, how long it can be cached, how fresh a document must be, and so on.

5.2 FILE TRANSFER PROTOCOL (FTP)

FTP is used to copy files from one host to another. FTP offers the mechanism for the same in following manner:

- FTP creates two processes such as Control Process and Data Transfer Process at both ends i.e. at client as well as at server.
- FTP establishes two different connections: one is for data transfer and other is for control information.

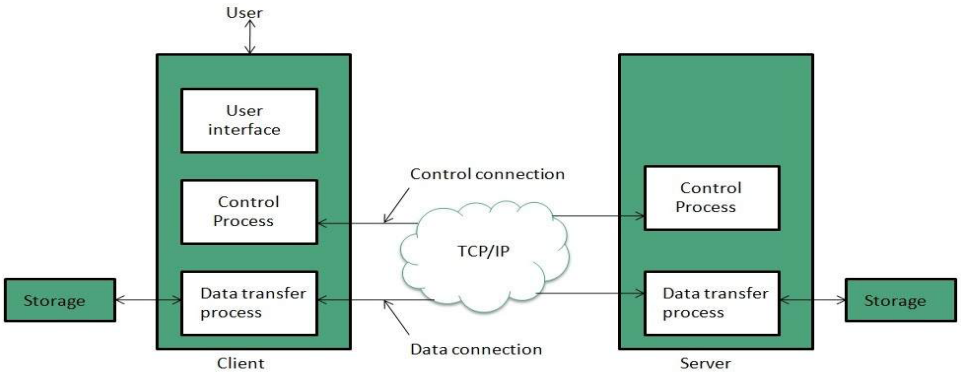


Figure : File Transfer Protocol

- **Control connection** is made between **control processes** while **Data Connection** is made between <="" b="">.

- FTP uses **port 21** for the control connection and **Port 20** for the data connection.

5.3 TRIVIAL FILE TRANSFER PROTOCOL (TFTP).

Trivial File Transfer Protocol is also used to transfer the files but it transfers the files without authentication. Unlike FTP, TFTP does not separate control and data information. Since there is no authentication exists, TFTP lacks in security features therefore it is not recommended to use TFTP.

Key points

- TFTP makes use of UDP for data transport. Each TFTP message is carried in separate UDP datagram.
- The first two bytes of a TFTP message specify the type of message.
- The TFTP session is initiated when a TFTP client sends a request to upload or download a file.
- The request is sent from an ephemeral UDP port to the **UDP port 69** of an TFTP server.

Difference between FTP and TFTP

S.N.	Parameter	FTP	TFTP
1	Operation	Transferring Files	Transferring Files
2	Authentication	Yes	No
3	Protocol	TCP	UDP
4	Ports	21 – Control, 20 – Data	Port 3214, 69, 4012
5	Control and Data	Separated	Separated
6	Data Transfer	Reliable	Unreliable

Table : Difference between FTP and TFTP

4.3 EMAIL

One of the most popular Internet services is electronic mail (e-mail).

User Agent

- The first component of an electronic mail system is the user agent *(UA)*.
- It provides service to the user to make the process of sending and receiving a message easier.

Services Provided by a User Agent

- A user agent is a software package that composes, reads, replies to, and forwards messages.
- It also handles mailboxes.

- Figure shows the services of a typical user agent.

Composing Messages

- A user agent helps the user compose the e-mail message to be sent out.
- Most user agents provide a template on the screen to be filled in by the user.

Reading Messages

- The second duty of the user agent is to read the incoming messages.
- When a user invokes a user agent, it first checks the mail in the incoming mailbox Most user agents show a one-line summary of each received mail.
- Each e-mail contains the following fields.
- A number field.
- A flag field that shows the status of the mail such as new, already read but not replied to, or read and replied to.
- The size of the message.
- The sender.
- The optional subject field.

Replying to Messages

- After reading a message, a user can use the user agent to reply to a message.
- The reply message may contain the original message and the new message.

Forwarding Messages

- Forwarding is defined as sending the message to a third party.
- A user agent allows the receiver to forward the message, with or without extra comments, to a third party.

Handling Mailboxes

- A user agent normally creates two mailboxes: an inbox and an outbox.
- The inbox keeps all the received e-mails until they are deleted by the user.
- The outbox keeps all the sent e-mails until the user deletes them.

User Agent Types

- There are two types of user agents: command-driven and GUI-based. Command-Driven
- A command-driven user agent normally accepts a one-character command from the keyboard to perform its task.
- For example, a user can type the character r, at the command prompt, to reply to the sender of the message, or type the character R to reply to the sender and all recipients.

GUI-Based

- They contain graphical-user interface (GUI) components that allow the user to interact with the software by using both the keyboard and the mouse.
- They have graphical components such as icons, menu bars, and windows that make the services easy to access.
- Some examples of GUI-based user agents are Eudora, Microsoft's Outlook, and Netscape.

Sending Mail

- To send mail, the user, through the UA, creates mail that looks very similar to postal mail.
- It has an envelope and a message.

Local Part

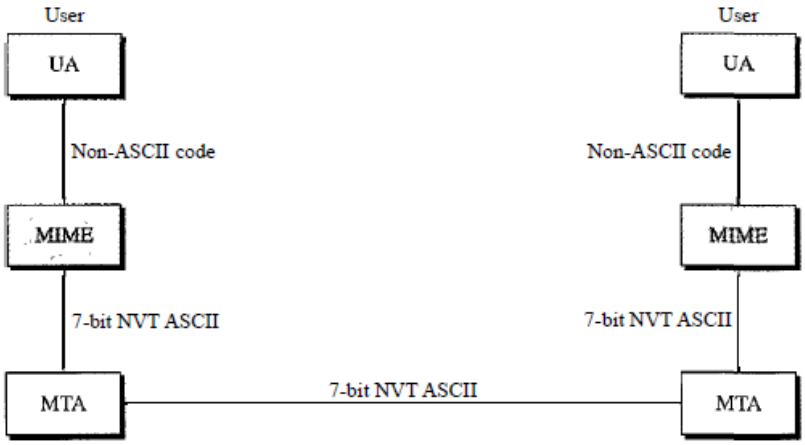
- The local part defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.

Domain Name

- An organization usually selects one or more hosts to receive and send e-mail; the hosts are sometimes called mail servers or exchangers.

MIME (Multipurpose Internet Mail Extensions)

- Electronic mail has a simple structure.
- It can send messages only in NVT 7-bit ASCII format.
- For example, it cannot be used for languages that are not supported by 7-bit
- ASCII characters (such as French, German, Hebrew, Russian, Chinese, and Japanese).
- Also, it cannot be used to send binary files or video or audio data.
- Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non- ASCII data to be sent through e-mail.
- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet.
- The message at the receiving side is transformed back to the original data.



MIME defines five headers that can be added to the original e-mail header section to define the transformation parameters:

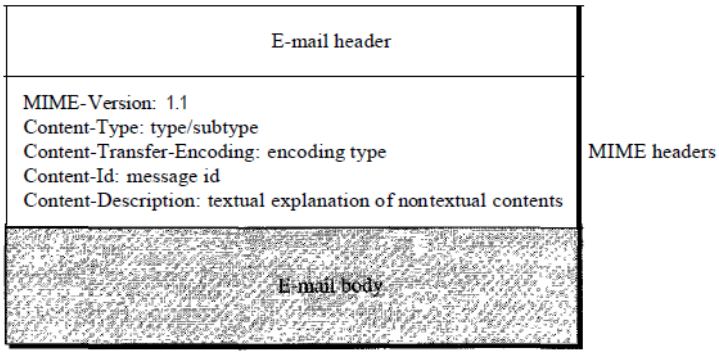
(1) MIME-Version

(2) Content-Type

(3) Content-Transfer-Encoding

(4) Content-Id

Content-Description



SMTP Architecture:

- The actual mail transfer is done through message transfer agents.
- To send mail, a system must have the client MTA, and to receive mail, a system must have a server
- MTA.
- The formal protocol that defines the MTA client and server in the Internet is called the Simple MailTransfer Protocol (SMTP).

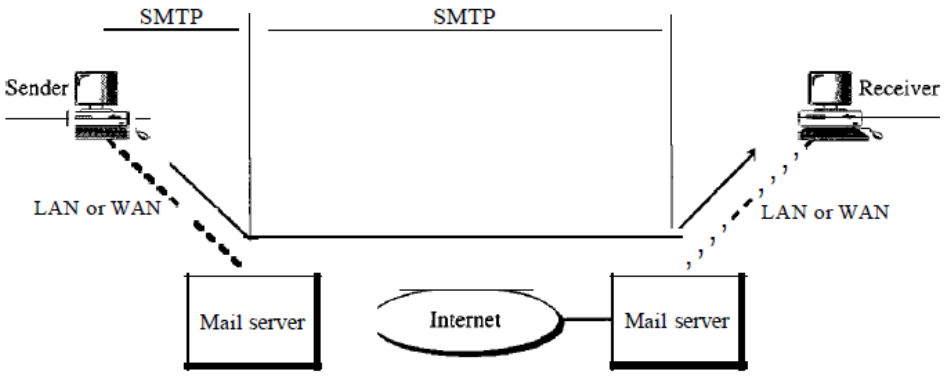
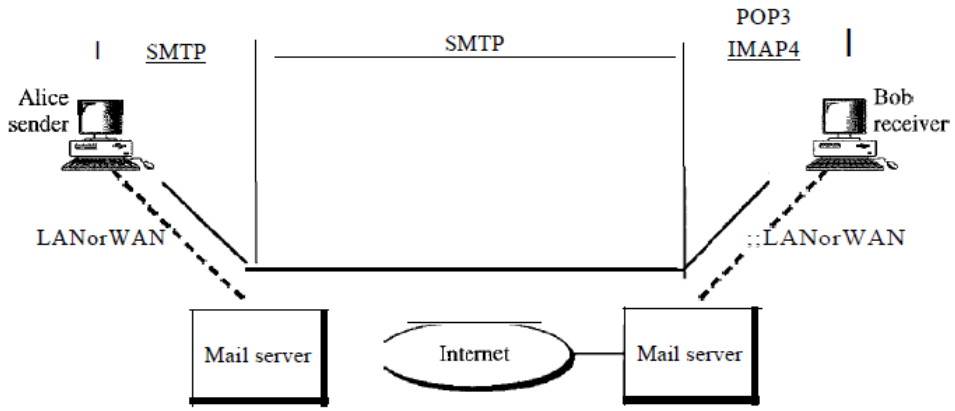


Figure 26.16 shows the range of the SMTP protocol in this scenario.

Message Access Agent: POP and IMAP

- The first and the second stages of mail delivery use SMTP.
- However, SMTP is not involved in the third stage because SMTP is a *push* protocol; it pushes the message from the client to the server.
- On the other hand, the third stage needs a *pull* protocol; the client must pull messages from the server.
- The third stage uses a message access agent.
- Currently two message access protocols are available: Post Office Protocol, version 3 (POP3) and Internet Mail Access Protocol, version 4 (IMAP4).



POP3

- Post Office Protocol, version 3 (POP3) is simple and limited in functionality.
- The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download e-mail from the mailbox on the mail server.

- The client opens a connection to the server on TCP port 110.
- It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

POP3 has two modes: the delete mode and the keep mode.

- In the delete mode, the mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval.
- The delete mode is normally used when the user is working at her permanent computer and can
- save and organize the received mail after reading or replying.
- The keep mode is normally used when the user accesses her mail away from her primary
- computer (e.g., a laptop).

Limitations of POP3

- It does not allow the user to organize her mail on the server; the user cannot have different folders on the server.
- POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4

- Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4).
- IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.

IMAP4 provides the following extra functions:

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the email contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage

Advantages

E-mail has prooved to be powerful and reliable medium of commmunication. Here are the benefits of E-mail:

- Reliable
- Convenience

- Speed
- Inexpensive
- Printable
- Global
- Generality

Reliable

Many of the mail systems notify the sender if e-mail message was undeliverable.

Convenience

There is no requirement of stationary and stamps. One does not have to go to post office. But all these things are not required for sending or receiving an mail.

Speed

E-mail is very fast. However, the speed also depends upon the underlying network.

Inexpensive

The cost of sending e-mail is very low.

Printable

It is easy to obtain a hardcopy of an e-mail. Also an electronic copy of an e-mail can also be saved for records.

Global

E-mail can be sent and received by a person sitting across the globe.

Disadvantages

Apart from several benefits of E-mail, there also exists some disadvantages as discussed below:

- Forgery
- Overload
- Misdirection
- Junk
- No response

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.

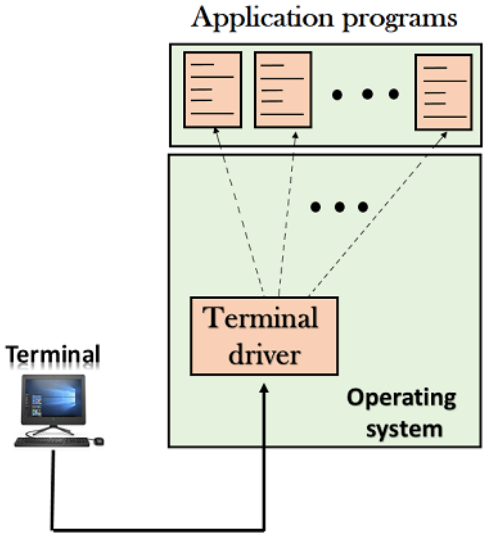
- Telnet provides a connection to the remote computer in such a way that a local terminal appears to be at the remote side.

There are two types of login:

- Local Login
- Remote Login

Local Login

- 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.



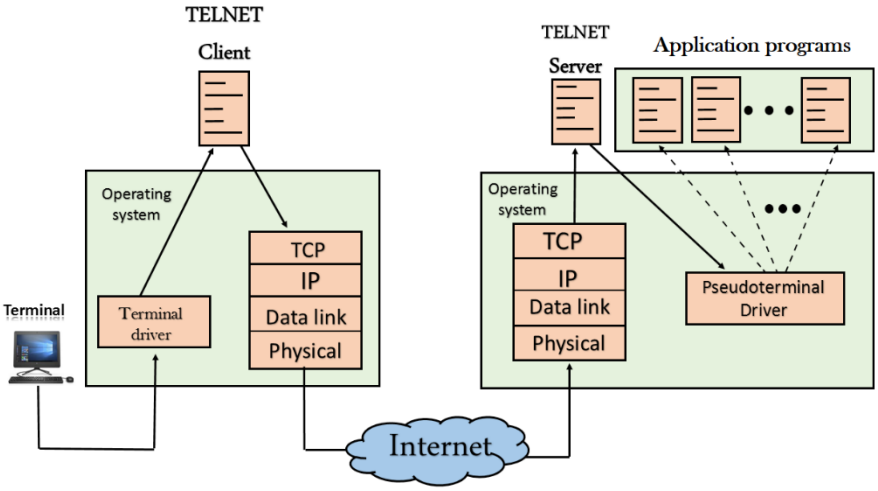
Remote login

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

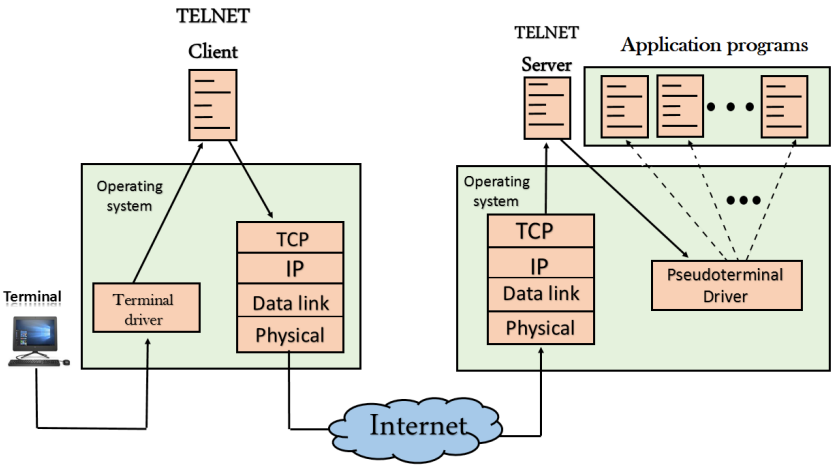
The user sends the keystrokes to the terminal driver, the characters are then sent to the TELNET client. The TELNET client which in turn, transforms the characters to a universal character set known as network virtual terminal characters and delivers them to the local TCP/IP stack



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.

SSH PROTOCOL

The SSH protocol (also referred to as Secure Shell) is a method for secure remote login from one computer to another. It provides several alternative options for strong authentication, and it protects the communications security and integrity with strong encryption. It is a secure alternative to the non-protected login protocols (such as telnet, rlogin) and insecure file transfer methods (such as FTP).

USES OF THE SSH PROTOCOL

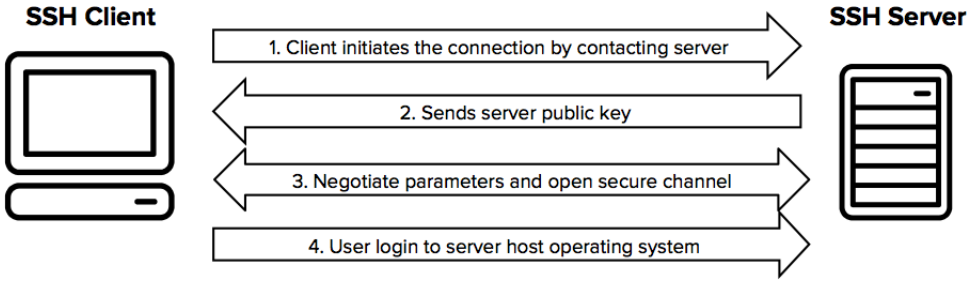
The protocol is used in corporate networks for:

- providing secure access for users and automated processes
- interactive and automated file transfers
- issuing remote commands
- managing network infrastructure and other mission-critical system components**.**

THE SSH PROTOCOL WORK

The protocol works in the client-server model, which means that the connection is established by the SSH client connecting to the SSH server. The SSH client drives the connection setup process and uses public key cryptography to verify the identity of the SSH server. After the setup phase the SSH protocol uses strong symmetric encryption and hashing algorithms to ensure the privacy and integrity of the data that is exchanged between the client and server.

The figure below presents a simplified setup flow of a secure shell connection.



Strong Authentication with SSH Keys

There are several options that can be used for user authentication. The most common ones are passwords and public key authentication.

The public key authentication method is primarily used for automation and sometimes by system administrators for single sign-on. It has turned out to be much more widely used than we ever anticipated. The idea is to have a cryptographic key pair - public key and private key - and configure the public key on a server to authorize access and grant anyone who has a copy of the private key access to the server. The keys used for authentication are called SSH keys. Public key authentication is also used with smartcards, such as the CAC and PIV cards used by US government.

The main use of key-based authentication is to enable secure automation. Automated secure shell file transfers are used to seamlessly integrate applications and also for automated systems & configuration management.

We have found that large organizations have way more SSH keys than they imagine, and managing SSH keys has become very important. SSH keys grant access as user names and passwords do. They require a similar provisioning and termination processes.

In some cases, we have found several million SSH keys authorizing access into production servers in customer environments, with 90% of the keys actually being unused and representing access that was provisioned but never terminated. Ensuring proper policies, processes, and audits also for SSH usage is critical for proper identity and access management. Traditional identity management projects have overlooked as much as 90% of all credentials by ignoring SSH keys. We provide services and tools for implementing SSH key management.

SSH Provides Strong Encryption and Integrity Protection

Once a connection has been established between the SSH client and server, the data that is transmitted is encrypted according to the parameters negotiated in the setup. During the negotiation the client and server agree on the symmetric encryption algorithm to be used and generate the encryption key that will be used.

The traffic between the communicating parties is protected with industry standard strong encryption algorithms (such as AES (Advanced Encryption Standard)), and the SSH protocol also includes a mechanism that ensures the integrity of the transmitted data by using standard hash algorithms (such as SHA-2 (Standard Hashing Algorithm)

SSH Advantages

SSH allows for the encryption of data so that those malicious would-be attackers cannot access your user information and passwords. SSH also allows for the tunnelling of other protocols such as FTP. Below is a list of specific things SSH protects you from.

IP source routing

While source routing is normally used for good purposes such as altering the path of a network signal if it originally fails, it can also be used by malicious users on to make a machine think it is talking to a different one.

DNS Spoofing

This is a type of hacking attack where data is inserted into a Domain Name System name server's cache database. This causes the name server to return an incorrect IP address so it can divert traffic to another computer. This is often the attacker's computer. From there they can obtain sensitive information.

Data manipulation at things like routers along the network.

This is fairly self explanatory, the attacker obtains or changes data at intermediaries along the network route. This is often performed at routers where data enters a sort of gateway or checkpoint on the way to its destination.

Eavesdropping or sniffing of the transmitted data.

If using an unsecure connection, an attacker can watch the data that goes through, collecting all sorts of sensitive or private information for their own malicious uses.

IP address spoofing

IP spoofing is the when a malicious user creates packets with a forged source IP address. This way it keeps the source computer's identity and location a secret and appears to be another computer that the receiver trusts.

5.6 DOMAIN NAME SYSTEM

- Domain Name System is an Internet service that translates domain names into IP addresses.
- The DNS has a distributed database that resides on multiple machines on the Internet.
- DNS has some protocols that allow the client and servers to communicate with each other.
- When the Internet was small, mapping was done by using hosts.txt file.
- The host file was located at host's disk and updated periodically from a master host file.
- When any program or any user wanted to map domain name to an address, the host consulted the host file and found the mapping.
- Now Internet is not small, it is impossible to have only one host file to relate every address with a name and vice versa.
- The solution used today is to divide the host file into smaller parts and store each part on a different computer.
- In this method, the host that needs mapping can call the closest computer holding the needed information.
- This method is used in Domain Name System (DNS).

Name space

- The names assigned to the machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

- There are two types of name spaces: Flat name spaces and Hierarchical names.

Flat name spaces

- In a flat name space, a name is a sequence of characters without structure.
- A name in this space is assigned to an address.
- The names were convenient and short.
- A flat name space cannot be used in a large system such as the internet because it must be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space

- In hierarchical name space, each name consists of several parts.
- First part defines the nature of the organization, second part defines the name of an organization, third part defines department of the organization, and so on.
- In hierarchical name space, the authority to assign and control the name spaces can be decentralized.
- Authority for names in each partition is passed to each designated agent.

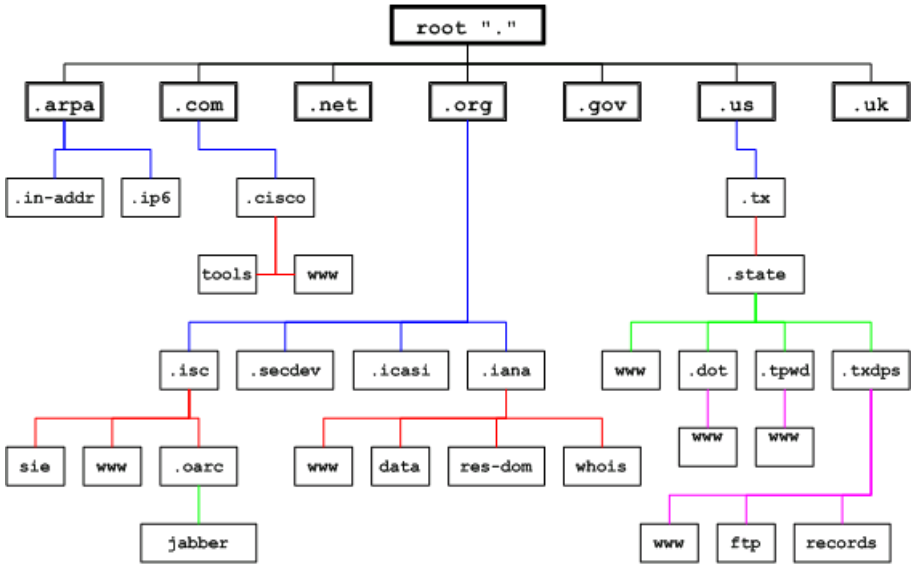


Figure 5.23: Hierarchical Name Space

DNS in the Internet

- DNS is a protocol that can be used in different platform.
- Domain Name Space is divided into different sections in the Internet: Generic domain, country domain and inverse domain.

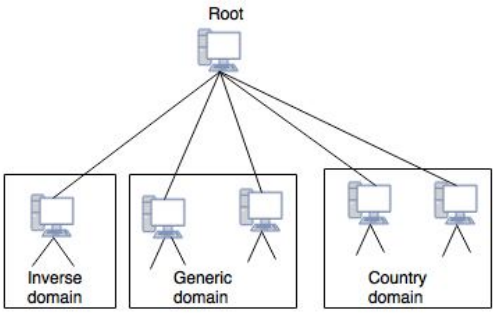


Figure 5.24: DNS in the Internet

Generic Domains

The generic domains define registered hosts according to their generic behavior. **Generic domain labels are as stated below:**

(1) Country Domains

- Country domain uses two character country abbreviations.
- Second labels can be more specific, national designation.
- **For example,** for Australia the country domain is "au", India is .in, UK is .uk etc.

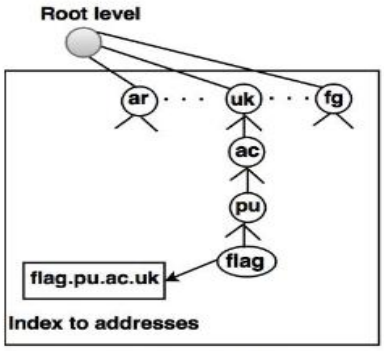


Figure : Country Domain

(2) Inverse Domains

- Inverse domain is used to map an address to a name.
- **For example,** a client send a request to the server for performing a particular task, server finds a list of authorized client. The list contains only IP addresses of the client.
- The server sends a query to the DNS server to map an address to a name to determine if the client is on the authorized list.
- This query is called an inverse query.
- This query is handled by first level node called ARPA.

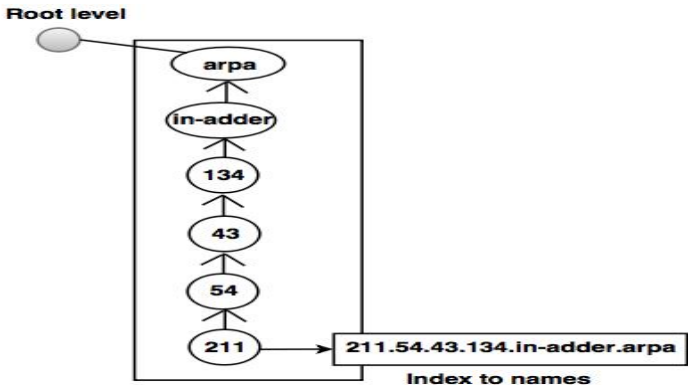
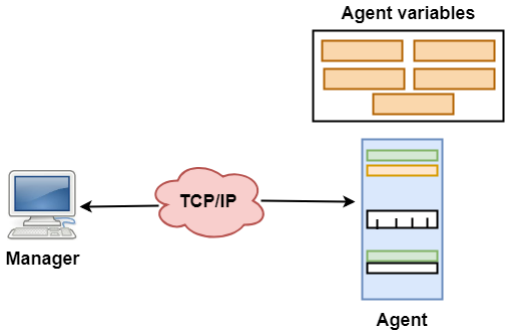


Figure : Inverse domain

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.

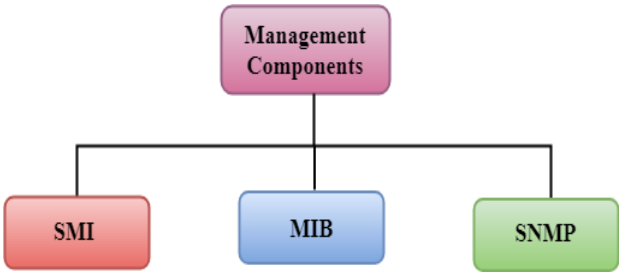
Management with SNMP has three basic ideas:

- 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).



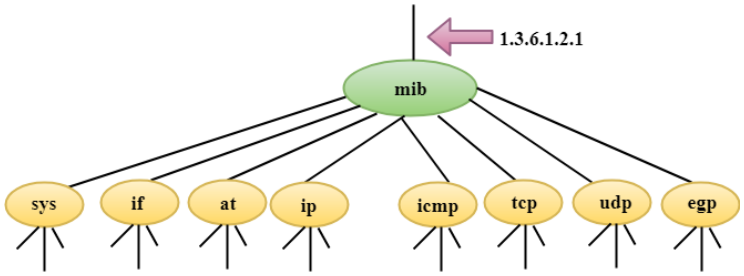
SMI

- 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

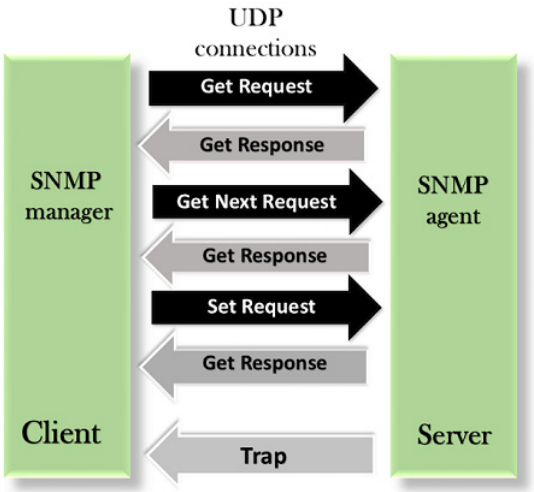
manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.



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.