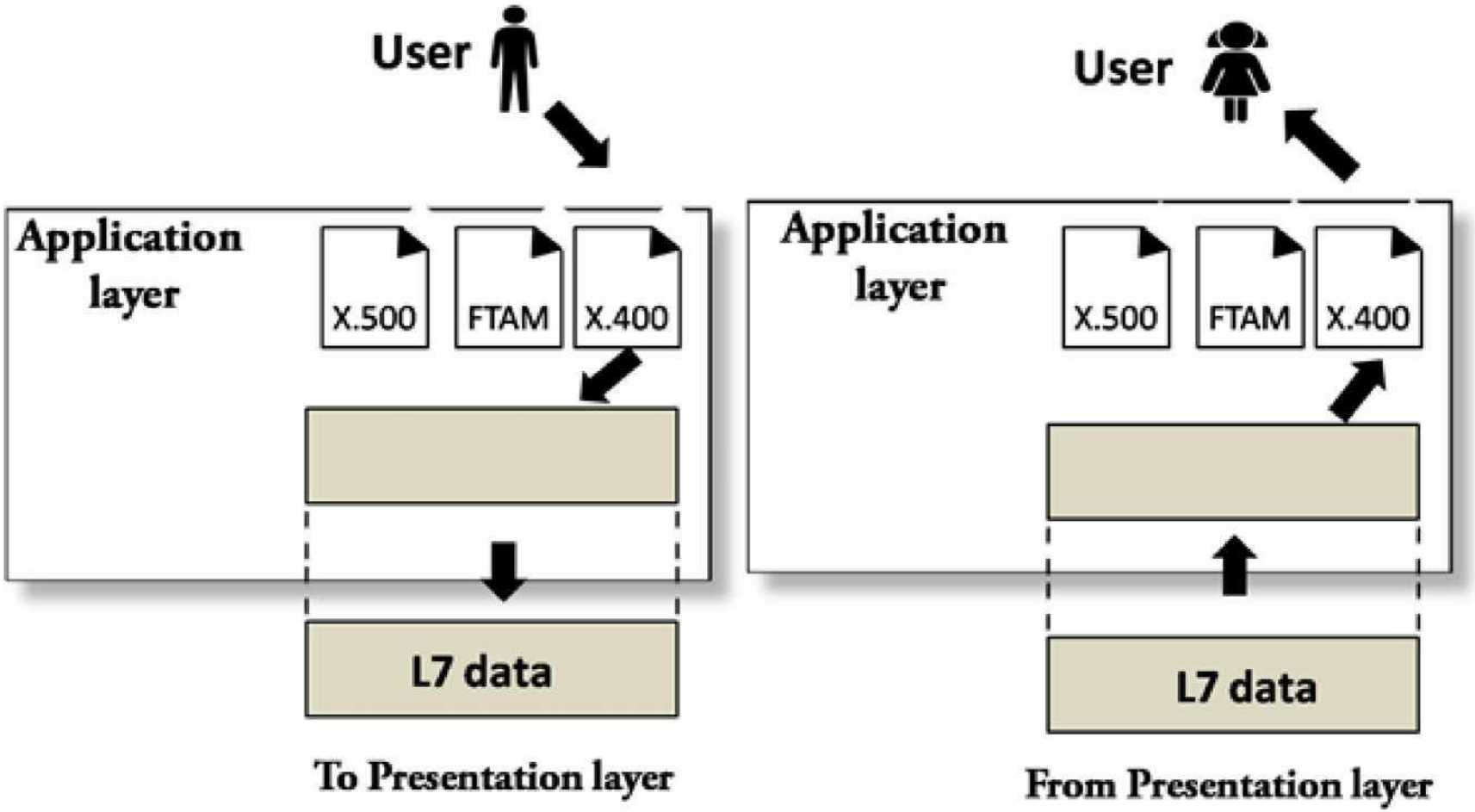
APPLICATION LAYER - OSI MODEL

It is the top most layer of OSI Model. Manipulation of data (information) in various ways is done in this layer which enables user or software to get access to the network. Some services provided by this layer includes: E-Mail, transferring files, distributing the results to user, directory services, network resources, etc.

The Application Layer contains a variety of protocols that are commonly needed by users. One widely-used application protocol is **HTTP (HyperText Transfer Protocol)**, which is the basis for the World Wide Web. When a browser wants a web page, it sends the name of the page it wants to the server using HTTP. The server then sends the page back.

Other Application protocols that are used are: **File Transfer Protocol (FTP)**, **Trivial File Transfer Protocol (TFTP)**, **Simple Mail Transfer Protocol (SMTP)**, **TELNET**, **Domain Name System (DNS)** etc.

APPLICATION LAYER - OSI MODEL



APPLICATION LAYER - OSI MODEL

Functions of Application Layer

1. **Network Virtual Terminal:** It allows a user to log on to a remote host. The application creates software emulation of a terminal at the remote host. User's computer talks to the software terminal which in turn talks to the host and vice versa. Then the remote host believes it is communicating with one of its own terminals and allows user to log on.
2. **Mail Services:** This layer provides the basis for E-mail forwarding and storage.
3. **Addressing:** To obtain communication between client and server, there is a need for addressing. When a client made a request to the server, the request contains the server address and its own address. The server response to the client request, the request contains the destination address, i.e., client address. To achieve this kind of addressing, DNS is used.
4. **Directory Services:** This layer provides access for global information about various services.

APPLICATION LAYER - OSI MODEL

5.**File Transfer, Access and Management (FTAM):** It is a standard mechanism to access files and manages it. Users can access files in a remote computer and manage it. They can also retrieve files from a remote computer.

Design Issues with Application Layer

There are commonly reoccurring problems that occur in the design and implementation of Application Layer protocols and can be addressed by patterns from several different pattern languages:

. Pattern Language for Application-level Communication Protocols . Service Design Patterns . Patterns of Enterprise Application Architecture . Pattern-Oriented Software Architecture

APPLICATION LAYER - OSI MODEL

Network Application Architecture

Application architecture is different from the network architecture. The network architecture is fixed and provides a set of services to applications. The application architecture, on the other hand, is designed by the application developer and defines how the application should be structured over the various end systems.

Application architecture is of two types:

1. **Client-server architecture:** An application program running on the local machine sends a request to another application program is known as a client, and a program that serves a request is known as a server. For example, when a web server receives a request from the client host, it responds to the request to the client host.

APPLICATION LAYER - OSI MODEL

Characteristics of Client-server architecture:

. In Client-server architecture, clients do not directly communicate with each other. For example, in a web application, two browsers do not directly communicate with each other.

. A server is fixed, well-known address known as IP address because the server is always on while the client can always contact the server by sending a packet to the sender's IP address.

Disadvantage of Client-server architecture:

It is a single-server based architecture which is incapable of holding all the requests from the clients. For example, a social networking site can become overwhelmed when there is only one server exists.

APPLICATION LAYER - OSI MODEL

2. P2P (peer-to-peer) architecture: It has no dedicated server in a data center. The peers are the computers which are not owned by the service provider. Most of the peers reside in the homes, offices, schools, and universities. The peers communicate with each other without passing the information through a dedicated server, this architecture is known as peer-to-peer architecture. The applications based on P2P architecture includes file sharing and internet telephony.

Features of P2P architecture

. **Self scalability:** In a file sharing system, although each peer generates a workload by requesting the files, each peer also adds a service capacity by distributing the files to the peer.

. **Cost-effective:** It is cost-effective as it does not require significant server infrastructure and server bandwidth.

Client and Server processes

. A network application consists of a pair of processes that send the messages to each other over a network.

APPLICATION LAYER - OSI MODEL

. In P2P file-sharing system, a file is transferred from a process in one peer to a process in another peer. We label one of the two processes as the client and another process as the server.

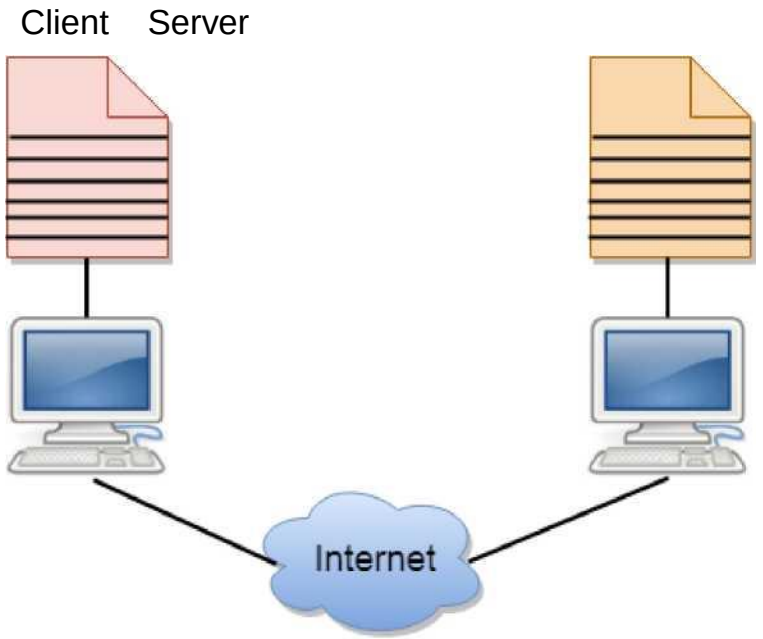
. With P2P file sharing, the peer which is downloading the file is known as a client, and the peer which is uploading the file is known as a server. However, we have observed in some applications such as P2P file sharing; a process can be both as a client and server. Therefore, we can say that a process can both download and upload the files.

APPLICATION LAYER - OSI MODEL

Client And Server Model

. A client and server networking model is a model in which computers such as servers provide the network services to the other computers such as clients to perform a user based tasks. This model is known as client-server networking model.

. The application programs using the client-server model should follow the given below strategies:



APPLICATION LAYER - OSI MODEL

. An application program is known as a client program, running on the local machine that requests for a service from an application program known as a server program, running on the remote machine.

. A client program runs only when it requests for a service from the server while the server program runs all time as it does not know when its service is required.

. A server provides a service for many clients not just for a single client. Therefore, we can say that client-server follows the many-to-one relationship. Many clients can use the service of one server.

. Services are required frequently, and many users have a specific client-server application program. For example, the client-server application program allows the user to access the files, send e-mail, and so on. If the services are more customized, then we should have one generic application program that allows the user to access the services available on the remote computer.

APPLICATION LAYER - OSI MODEL

Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

Server

A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises. The server waits for the incoming requests from the clients. When the request arrives at the server, then it responds to the request.

APPLICATION LAYER - OSI MODEL

Advantages of Client-server networks:

. **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.

. **Security:** These networks are more secure as all the shared resources are centrally administered.

. **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.

. **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

APPLICATION LAYER - OSI MODEL

Disadvantages of Client-Server network:

. **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.

. It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.

. A client/server network is very decisive. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.

. Sometimes the resources exist in the server but may not exist in the client. For example, If the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

APPLICATION LAYER - OSI MODEL

Virtual Terminal

In open systems, a virtual terminal (VT) is an application service that:

1. Allows host terminals on a multi-user network to interact with other hosts regardless of terminal type and characteristics,
2. Allows remote log-on by local area network managers for the purpose of management,
3. Allows users to access information from another host processor for transaction processing,
4. Serves as a backup facility.

They are also called Virtual Console, are emulated text terminals, using the keyboard and monitor of a personal computer or workstation. The word "text" is key since virtual consoles are not GUI terminals and they do not run inside a graphical interface. Virtual consoles are found on all GNU/Linux systems, even

APPLICATION LAYER - OSI MODEL

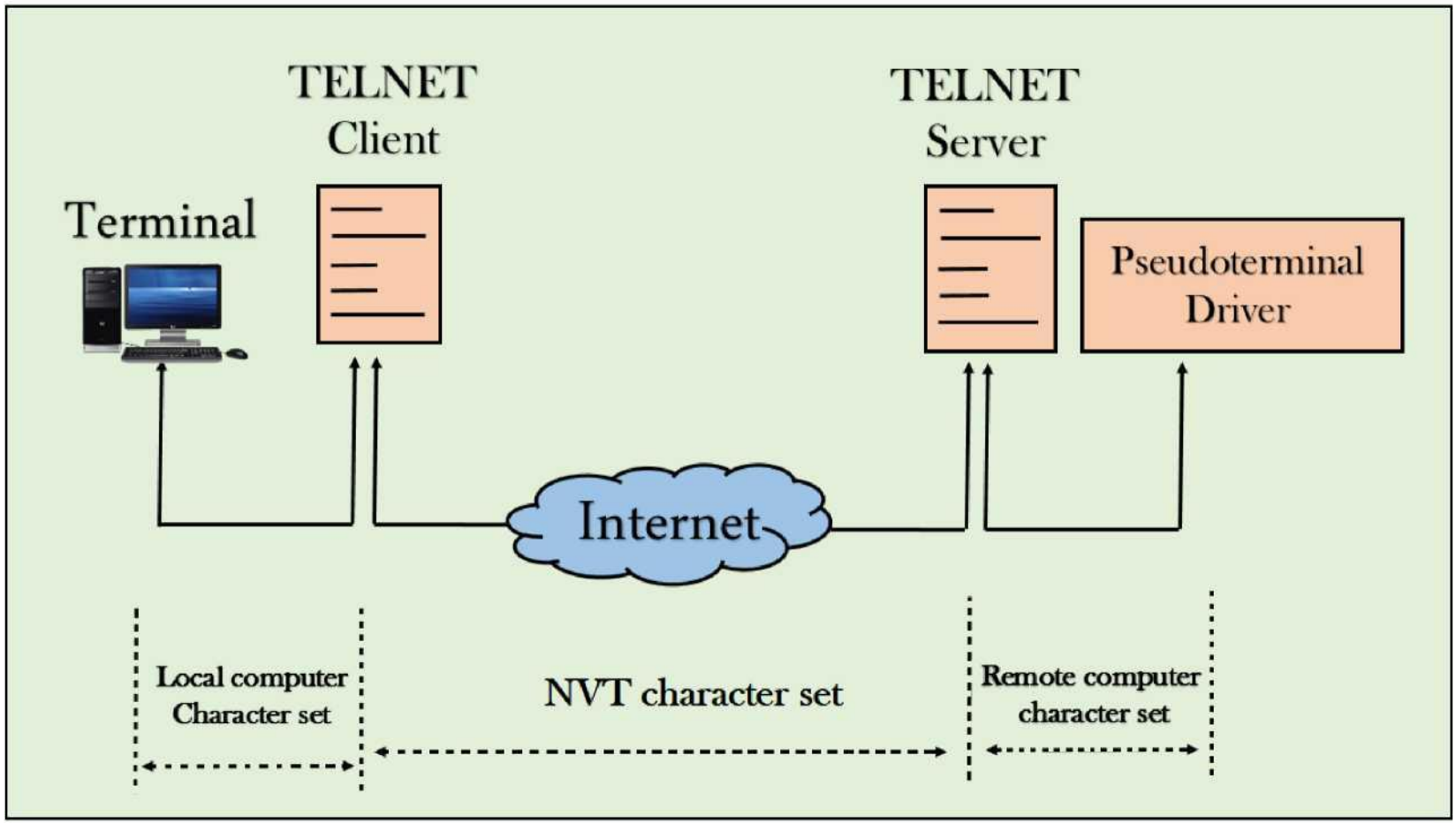
on systems which don't have a desktop environment or graphical system installed. They are primarily used to access and interact with servers.

PuTTY is an example of a virtual terminal.

ITU-T defines a virtual terminal protocol based on the OSI application layer protocols. However, the virtual terminal protocol is not widely used on the Internet.

APPLICATION LAYER - OSI MODEL

NETWORK VIRTUAL TERMINAL (NVT)



- The network virtual terminal is an interface that defines how data and commands are sent across the network.

APPLICATION LAYER - OSI MODEL

- 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.

MAIL SERVICES

Email is a store-and-forward method of sending, storing, and retrieving electronic messages across a network. Email messages are stored in databases on mail servers. ISPs often maintain mail servers that support many different customer accounts.

Email clients communicate with mail servers to send and receive email. Mail servers communicate with other mail servers to transport messages from one domain to another. An email client does not communicate directly with another email client when sending email. Instead, both clients rely on the mail server to transport messages. This is true even when both users are in the same domain.

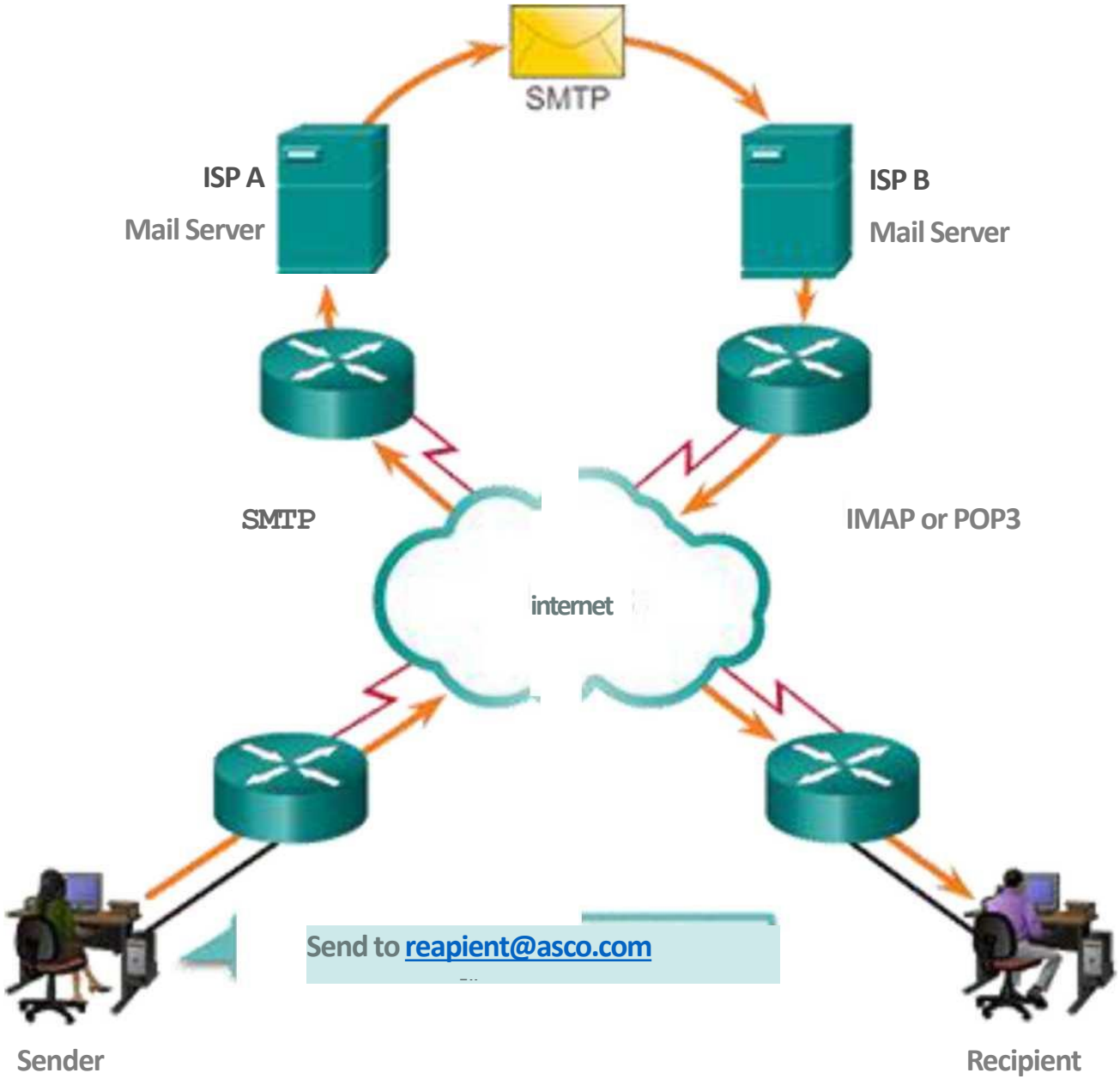
APPLICATION LAYER - OSI MODEL

Email clients send messages to the email server configured in the application settings. When the server receives the message, it checks to see if the recipient domain is located on its local database. If it is not, it sends a DNS request to determine the IP address of the mail server for the destination domain. The email is then forwarded to the appropriate server.

Email supports three separate protocols for operation: Simple Mail Transfer Protocol (**SMTP**), Post Office Protocol (**POP**), and Internet Message Access Protocol (**IMAP**). **The application layer process** that sends mail, uses SMTP. This is the case if sending from a client to a server, as well as when sending from one server to another.

A client retrieves email, however, using one of two application layer protocols: POP or IMAP.

APPLICATION LAYER - OSI MODEL



Simple Mail Transfer Protocol

Simple Mail Transfer Protocol (SMTP) transfers mail reliably and efficiently. For SMTP applications to work properly, the mail message must be formatted properly and SMTP processes must be running on both the client and server.

SMTP message formats require a message header and a message body. While the message body can contain any amount of text, the message header must have a properly formatted recipient email address and a sender address. Any other header information is optional.

When a client sends email, the client SMTP process connects with a server SMTP process on well-known port 25. After the connection is made, the client attempts to send the email to the server across the connection. When the server receives the message, it either places the message in a local account, if the recipient is local, or forwards the message using the same SMTP connection process to another mail server for delivery.

The destination email server may not be online or may be busy when email messages are sent. Therefore, SMTP spools messages to be sent at a later time. Periodically, the server checks the queue for messages and attempts to send them again. If the message is still not delivered after a predetermined expiration time, it is returned to the sender as undeliverable.

Post Office Protocol

Post Office Protocol (POP) enables a workstation to retrieve mail from a mail server. With POP, mail is downloaded from the server to the client and then deleted on the server.

The server starts the POP service by passively listening on TCP port 110 for client connection requests. When a client wants to make use of the service, it sends a request to establish a TCP connection with the server. When the connection is established, the POP server sends a greeting. The client and POP server then exchange commands and responses until the connection is closed or aborted.

APPLICATION LAYER - OSI MODEL

Because email messages are downloaded to the client and removed from the server, there is not a centralized location where email messages are kept. Because POP does not store messages, it is undesirable for a small business that needs a centralized backup solution.

POP3 is desirable for an ISP, because it alleviates their responsibility for managing large amounts of storage for their email servers.

Internet Message Access Protocol

Internet Message Access Protocol (IMAP) is another protocol that describes a method to retrieve email messages. However, unlike POP, when the user connects to an IMAP-capable server, copies of the messages are downloaded to the client application. The original messages are kept on the server until manually deleted. Users view copies of the messages in their email client software.

Users can create a file hierarchy on the server to organize and store mail. That file structure is duplicated on the email client as well. When a user

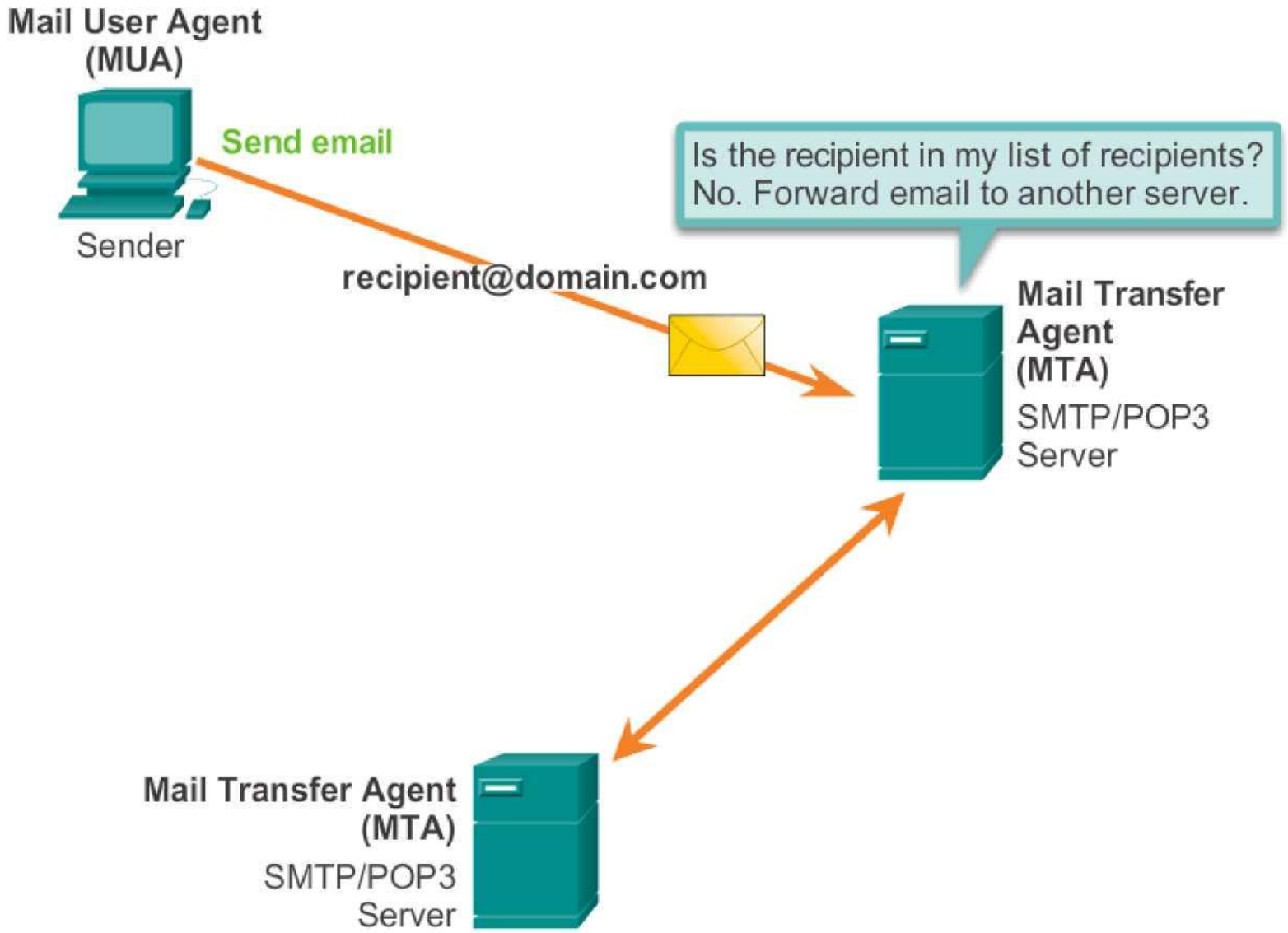
APPLICATION LAYER - OSI MODEL

decides to delete a message, the server synchronizes that action and deletes the message from the server.

For small- to medium-sized businesses, there are many advantages to using IMAP. IMAP can provide long-term storage of email messages on mail servers and allows for centralized backup. It also enables employees to access email messages from multiple locations, using different devices or client software. The mailbox folder structure that a user expects to see is available for viewing regardless of how the user accesses the mailbox.

For an ISP, IMAP may not be the protocol of choice. It can be expensive to purchase and maintain the disk space to support the large number of stored emails. Additionally, if customers expect their mailboxes to be backed up routinely, that can further increase the costs to the ISP.

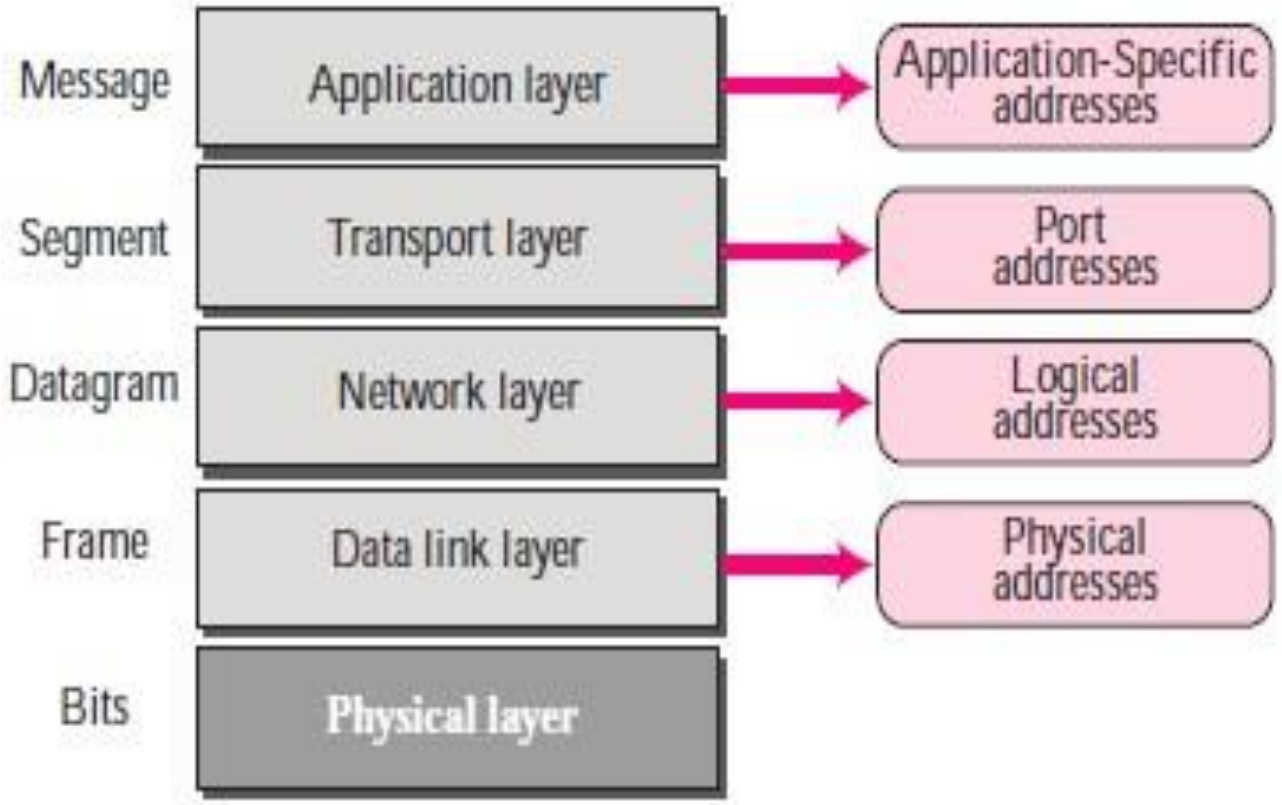
APPLICATION LAYER - OSI MODEL



APPLICATION LAYER - OSI MODEL

ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: physical address, logical address, port address, and application-specific address. Each address is related to a one layer in the TCP/IP architecture, as shown in the following Figure.

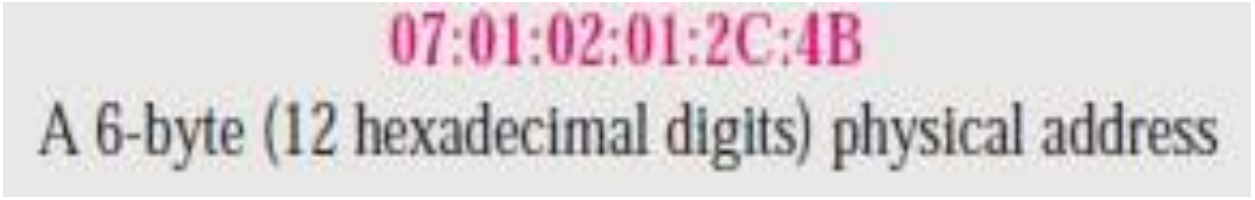


APPLICATION LAYER - OSI MODEL

Physical Addresses

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC).

Most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below.



A 6-byte (12 hexadecimal digits) physical address

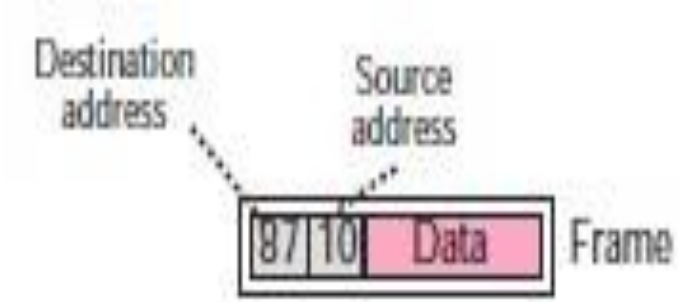
Example (1)

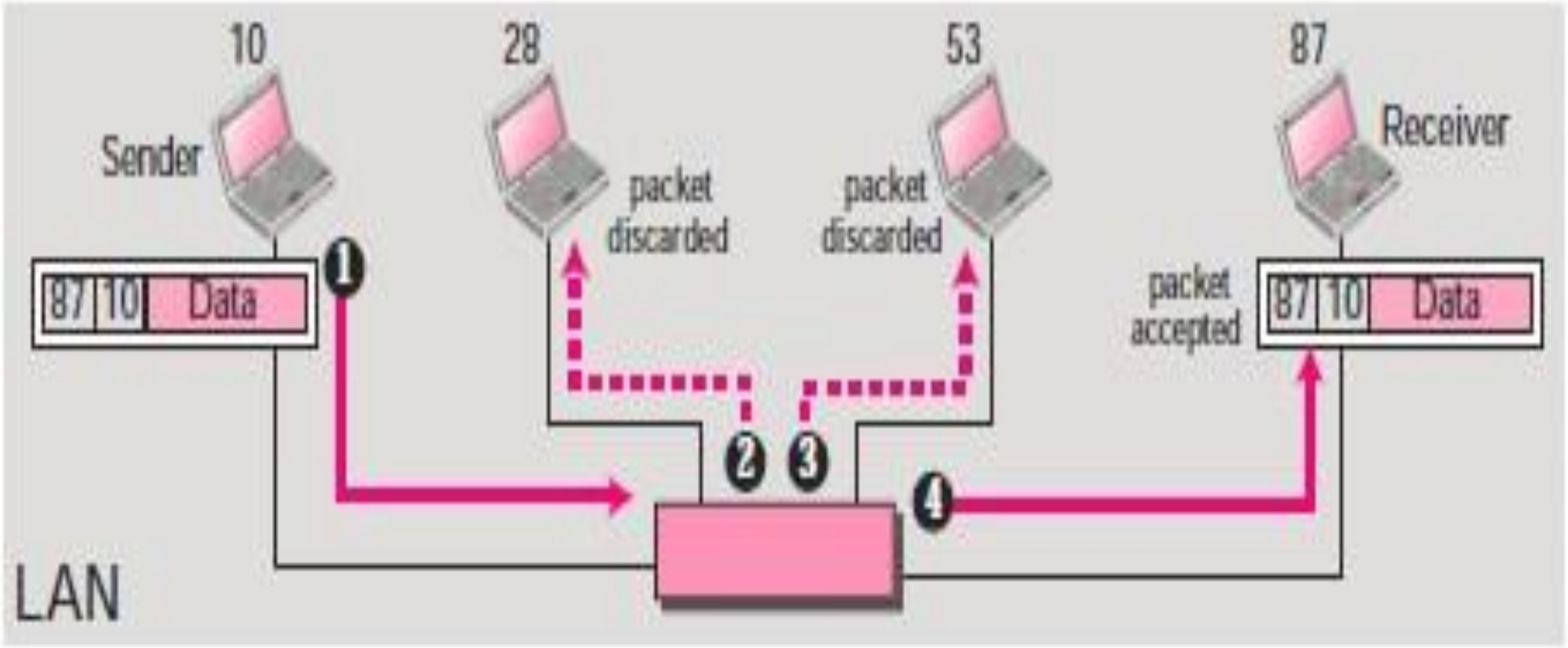
In Figure below a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (a LAN). At the

data link layer, this frame contains physical (link) addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. The trailer usually contains extra bits needed for error detection. The data link layer at the sender receives data from an upper layer. It encapsulates the data in a frame, adding a header and a trailer. The header, among other pieces of information, carries the receiver and the sender physical (link) addresses.

Note that in most data link protocols, the destination address 87 in this case, comes before the source address (10 in this case). The frame is propagated through the LAN. Each station with a physical address other than 87 drops the frame because the destination address in the frame does not match its own physical address. The intended destination computer, however, finds a match between the destination address in the frame and its own physical address. The frame is checked, the header and trailer are dropped, and the data part is decapsulated and delivered to the upper layer.

APPLICATION LAYER - OSI MODEL



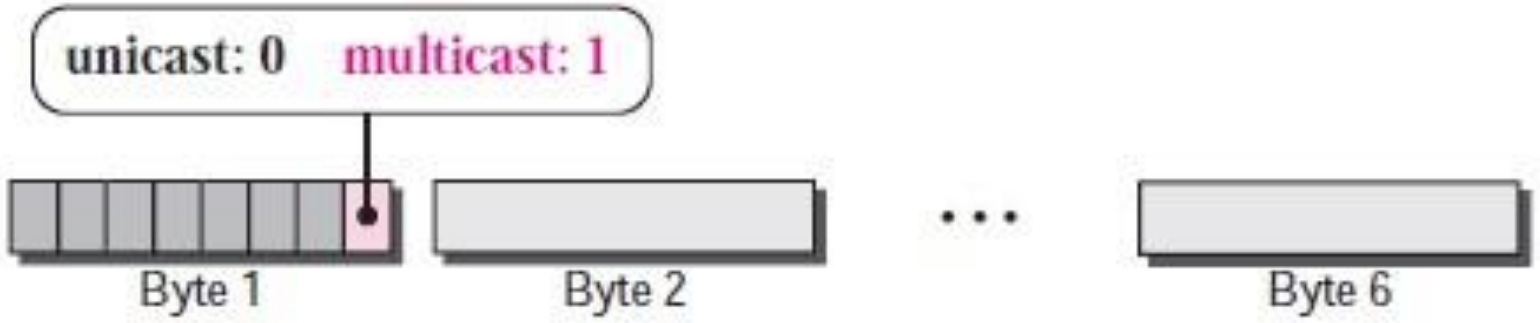


APPLICATION LAYER - OSI MODEL

Unicast, Multicast, and Broadcast Physical Addresses

Physical addresses can be either unicast (one single recipient), multicast (a group of recipients), or broadcast (to be received by all systems in the network). Some networks support all three addresses.

A source address is always a unicast address—the frame comes from only one station. The destination address, however, can be unicast, multicast, or broadcast. The least significant bit of the first byte defines the type of address.



Q: Define the type of the following destination addresses:

1. 4A:30:10:21:10:1A
2. 47:20:1B:2E:08:EE
3. FF:FF:FF:FF:FF:FF

APPLICATION LAYER - OSI MODEL

Logical Addresses

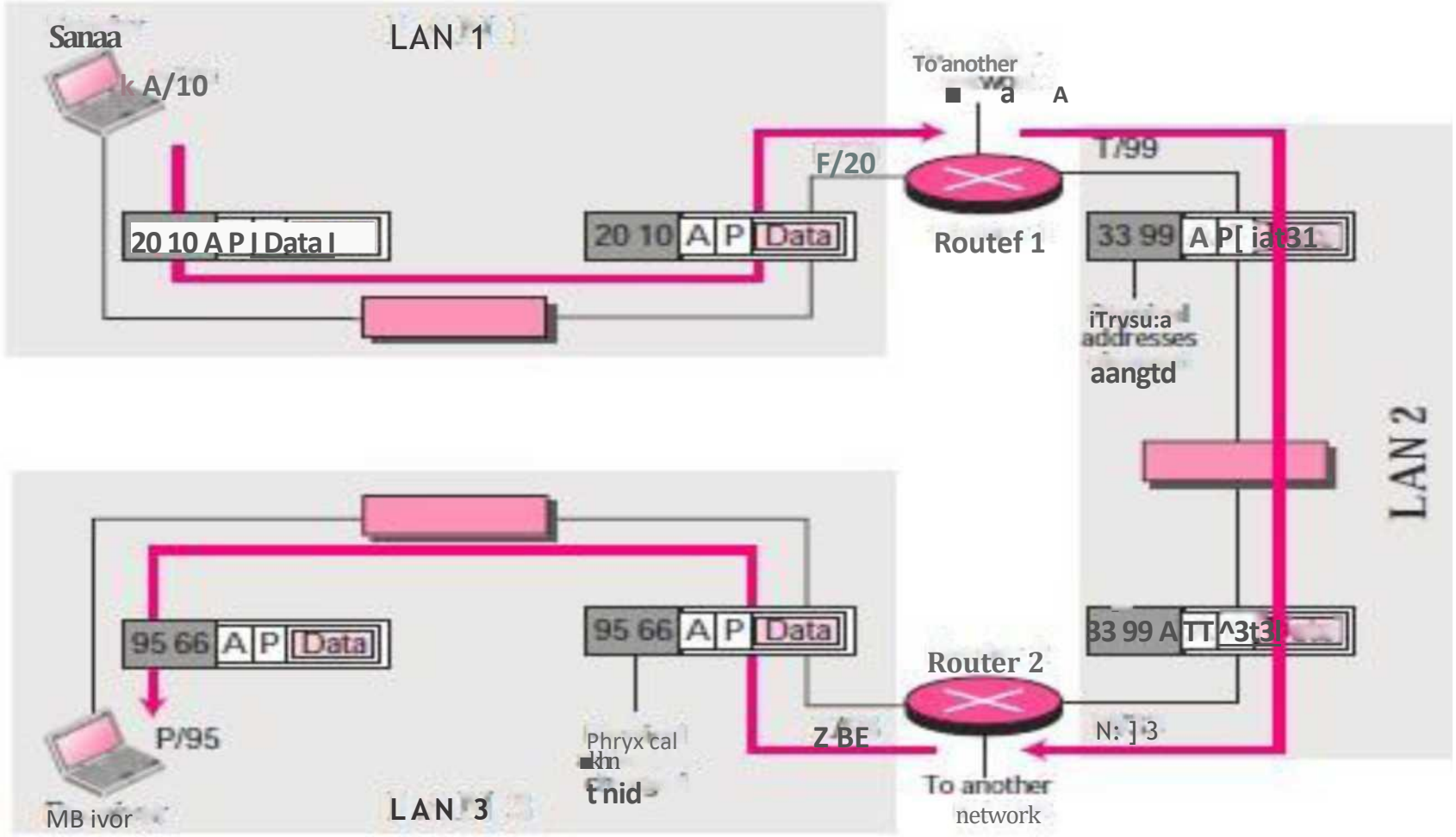
Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

APPLICATION LAYER - OSI MODEL

Example (2)

The Figure below shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection. Although it may be obvious that each router must have a separate physical address for each connection, it may. The computer with logical address **A** and physical address **10** needs to send a packet to the computer with logical address **P** and physical address **95**. The sender encapsulates its data in a packet at the network layer and adds two logical addresses (A and P). Note that in most protocols, the logical source address comes before the logical destination address (contrary to the order of physical addresses). The network layer, however, needs to find the physical address of the next hop before the packet can be delivered. The network layer consults its routing table and finds the logical address of the next hop (router 1) to be F.

APPLICATION LAYER - OSI MODEL



Another protocol, Address Resolution Protocol (ARP) finds the physical address of router 1 that corresponds to its logical address (20). Now the network layer passes this address to the data link layer, which in turn,

APPLICATION LAYER - OSI MODEL

encapsulates the packet with physical destination address 20 and physical source address 10. The router decapsulates the packet from the frame to read the logical destination address P. Since the logical destination address does not match the router's logical address, the router knows that the packet needs to be forwarded. The router consults its routing table and ARP to find the physical destination address of the next hop (router 2), creates a new frame, encapsulates the packet, and sends it to router 2.

Note the physical addresses in the frame. The source physical address changes from 10 to 99. The destination physical address changes from 20 (router 1 physical address) to 33 (router 2 physical address). The logical source and destination addresses must remain the same; otherwise the packet will be lost. At router 2 we have a similar scenario. The physical addresses are changed, and a new frame is sent to the destination computer. When the frame reaches the destination, the packet is decapsulated. The destination logical address P matches the logical address of the computer. The data are decapsulated from the packet and delivered to the upper layer. Note that although physical

addresses will change from hop to hop, logical addresses remain the same from the source to destination.

The physical addresses will change from hop to hop, but the logical addresses remain the same.

Unicast, Multicast, and Broadcast Addresses

The logical addresses can be either unicast (one single recipient), multicast (a group of recipients), or broadcast (all systems in the network).

Port Addresses

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. Computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating

APPLICATION LAYER - OSI MODEL

with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes.

In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

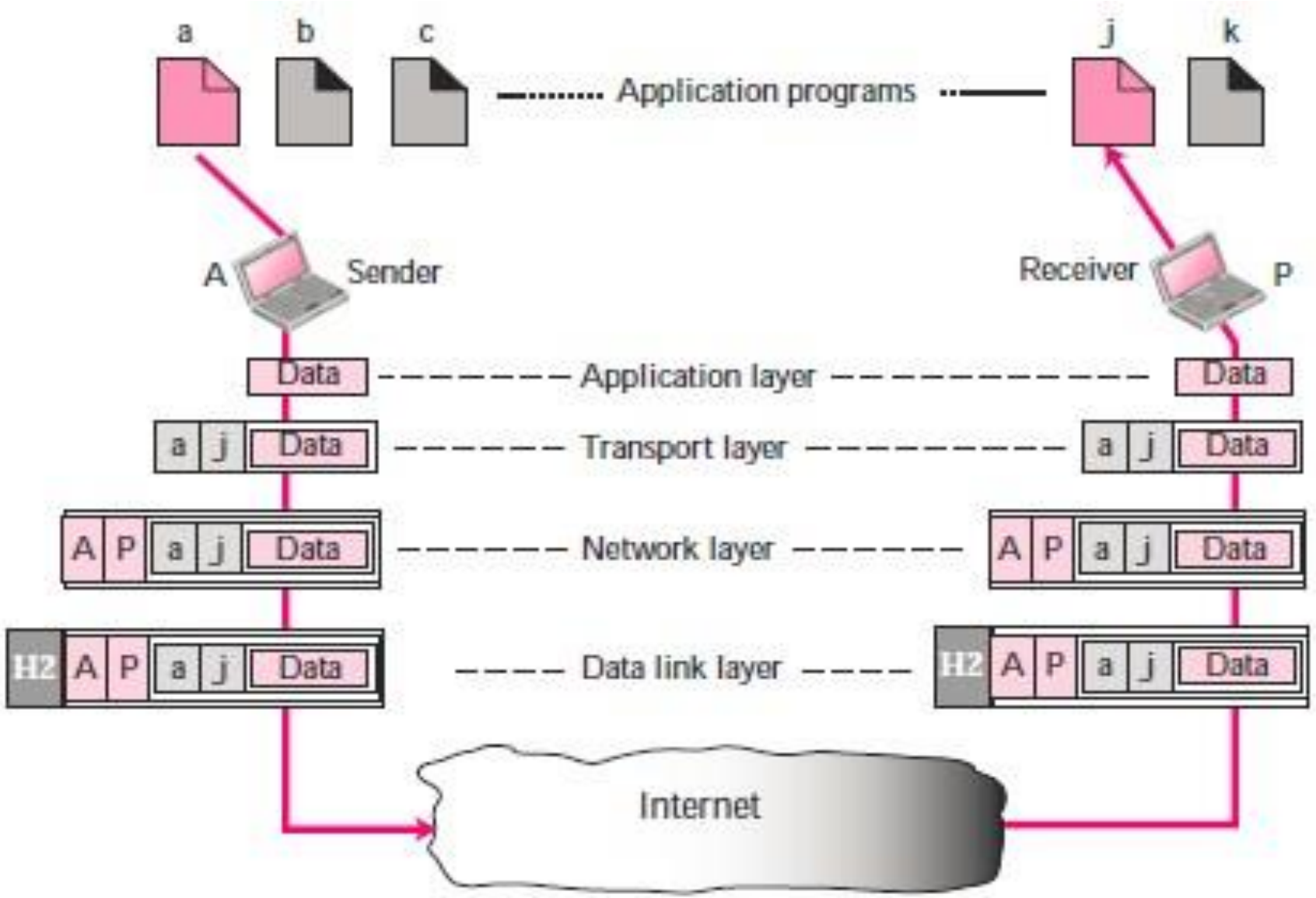
A port address is a 16-bit address represented by one decimal number as shown.

APPLICATION LAYER - OSI MODEL

Example (3)

The following Figure shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses **a, b,** and **c**. The receiving computer is running two processes at this time with port addresses **j** and **k**. Process **a** in the sending computer needs to communicate with process **j** in the receiving computer. Note that although both computers are using the same application, FTP, for example, the port addresses are different because one is a client program and the other is a server program.

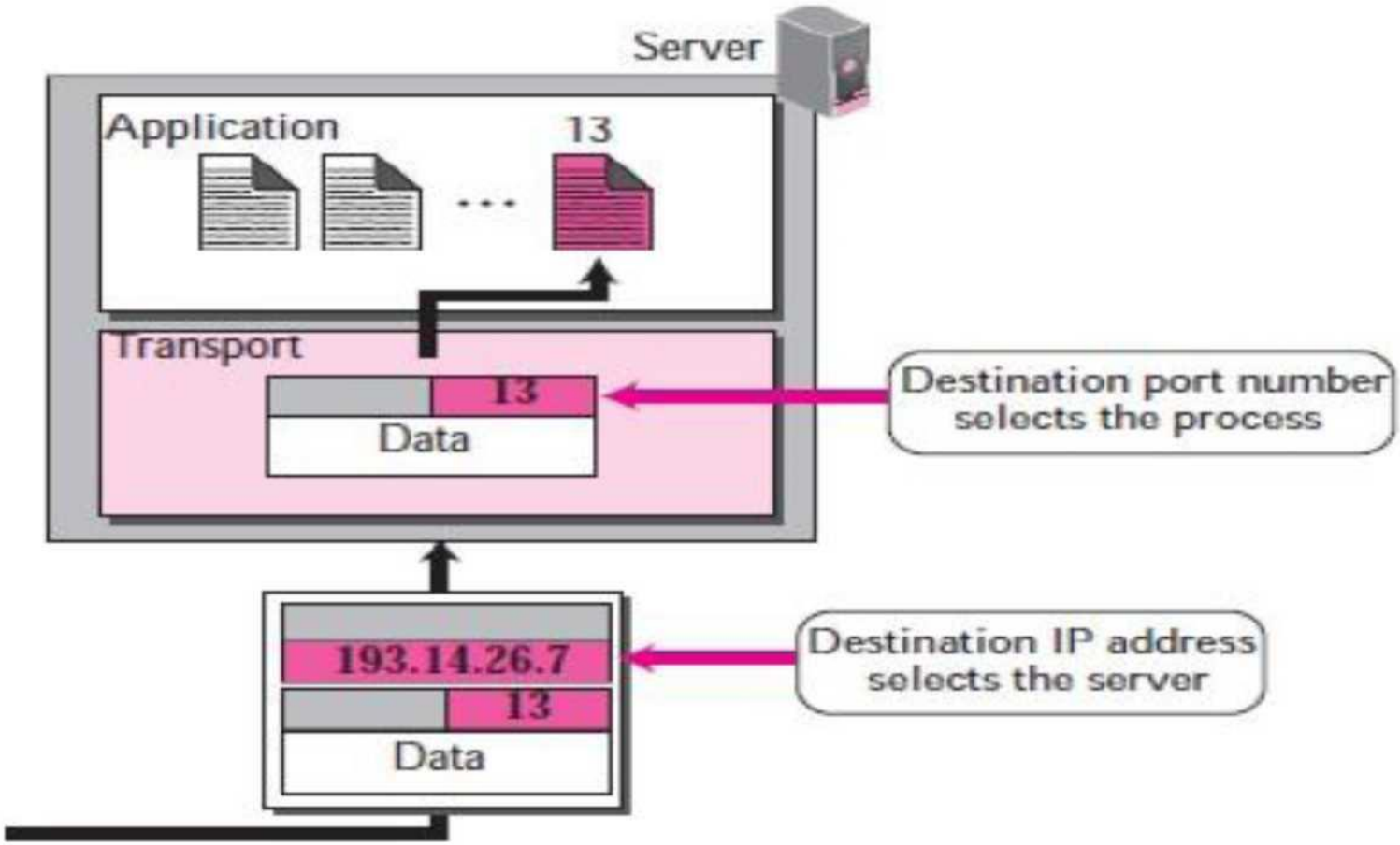
APPLICATION LAYER - OSI MODEL



APPLICATION LAYER - OSI MODEL

To show that data from process **a** need to be delivered to process **j**, and not **k**, the transport layer encapsulates data from the application layer in a packet and adds two port addresses (**a** and **j**), source and destination. The packet from the transport layer is then encapsulated in another packet at the network layer with logical source and destination addresses (**A** and **P**). Finally, this packet is encapsulated in a frame with the physical source and destination addresses of the next hop. We have not shown the physical addresses because they change from hop to hop inside the cloud designated as the Internet. Note that although physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination.

APPLICATION LAYER - OSI MODEL



- In the TCP/IP protocol suite, the port numbers are integers between 0 and 65,535.

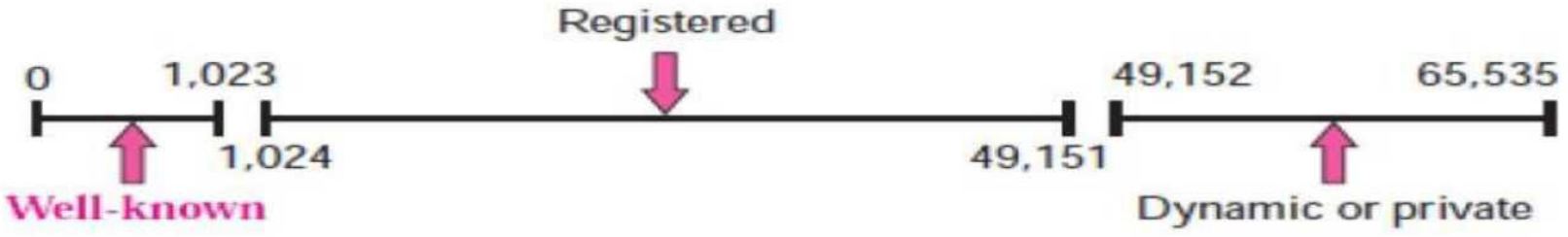
APPLICATION LAYER - OSI MODEL

- The client program defines itself with a port number, called the **ephemeral port number** (chosen randomly). The word ephemeral means *short lived.*
- The server process must also define itself with a port number (called well-known port numbers). This port number, however, cannot be chosen randomly.

APPLICATION LAYER - OSI MODEL

ICANN Ranges (Internet Corporation for Assigned Names and Numbers)

ICANN has divided the port numbers into three ranges: well-known, registered, and dynamic (or private)



- **Well-known ports**: The ports ranging from 0 to 1,023 are assigned and controlled by ICANN..
- **Registered ports**: The ports ranging from 1,024 to 49,151 are not assigned or controlled by ICANN. They can only be registered with ICANN to prevent duplication.

APPLICATION LAYER - OSI MODEL

•**Dynamic ports**: The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used as temporary or private port numbers. The original recommendation was that the ephemeral port numbers for clients be chosen from this range. However, most systems do not follow this recommendation.

Application-Specific Addresses

Some applications have user-friendly addresses that are designed for that specific application. Examples include the e-mail address (for example, co_sci@yahoo.com) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web. These addresses, however, get changed to the corresponding port and logical addresses by the sending computer.

DIRECTORY SERVICES

A Directory Service is nothing but a software system that responds to requests for information about entities, e.g. people in an organization.
X.500 and Network Information Service (NIS) are examples of directory services.

Need of Directory Service

Enterprise Computing Environments have a need to store information in a centralized data store so that it can be added to, deleted, modified, and queried by users and applications. The information stored could be user accounts, email addresses, digital certificates, component object names, network names, printers, groups and so on. There is a need to access this information both from within the enterprise and from the Internet. The amount of information stored

APPLICATION LAYER - OSI MODEL

varies greatly with the customer. This data store has come to be known as a *Directory Service.*

Directory services not only allow you to locate and access these resources, but also let you manage the relationships among them.

For our own use, we all maintain personal address directory where we store addresses, telephone nos. and other information in a format that is most suitable for us. But when we talk about maintaining a global directory service on Internet or in any organization, *The Directory Service must be:*

- *Flexible* enough to store a range of information types
- *Secure* when accessing from both the Internet and intranet
- *Scalable* from a small business to the largest enterprise
- *Extensible* as business needs change
- *Accessible* via an open, standards-based protocol

APPLICATION LAYER - OSI MODEL

Using an open protocol enables the information in the Directory Service to be accessible from clients from different vendors. Directory Services from different vendors communicating using an open protocol can exchange information with each other to create aggregated directories.

X.500 Directory

X.500 is a standard for a Directory Service by the International Telecommunications

Union (ITU). X.500, the OSI directory standard, defines a comprehensive Directory Service, including an information model, namespace, functional model, and authentication framework. X.500 also defines the Directory Access Protocol (DAP) used by clients to access the directory. DAP is a full OSI protocol that contains extensive functionality, much of which is not used by most applications.

Topic Objective & recap of previous Topic

- **APPLICATION PROTOCOLS (CO3)**

Topic Objective

- We will understand the Various protocols in Application layer
- How various services are provided

Recap of Previous Topic

- Application Layer
- Functions of the Application layer

APPLICATION PROTOCOLS (CO3)

DNS

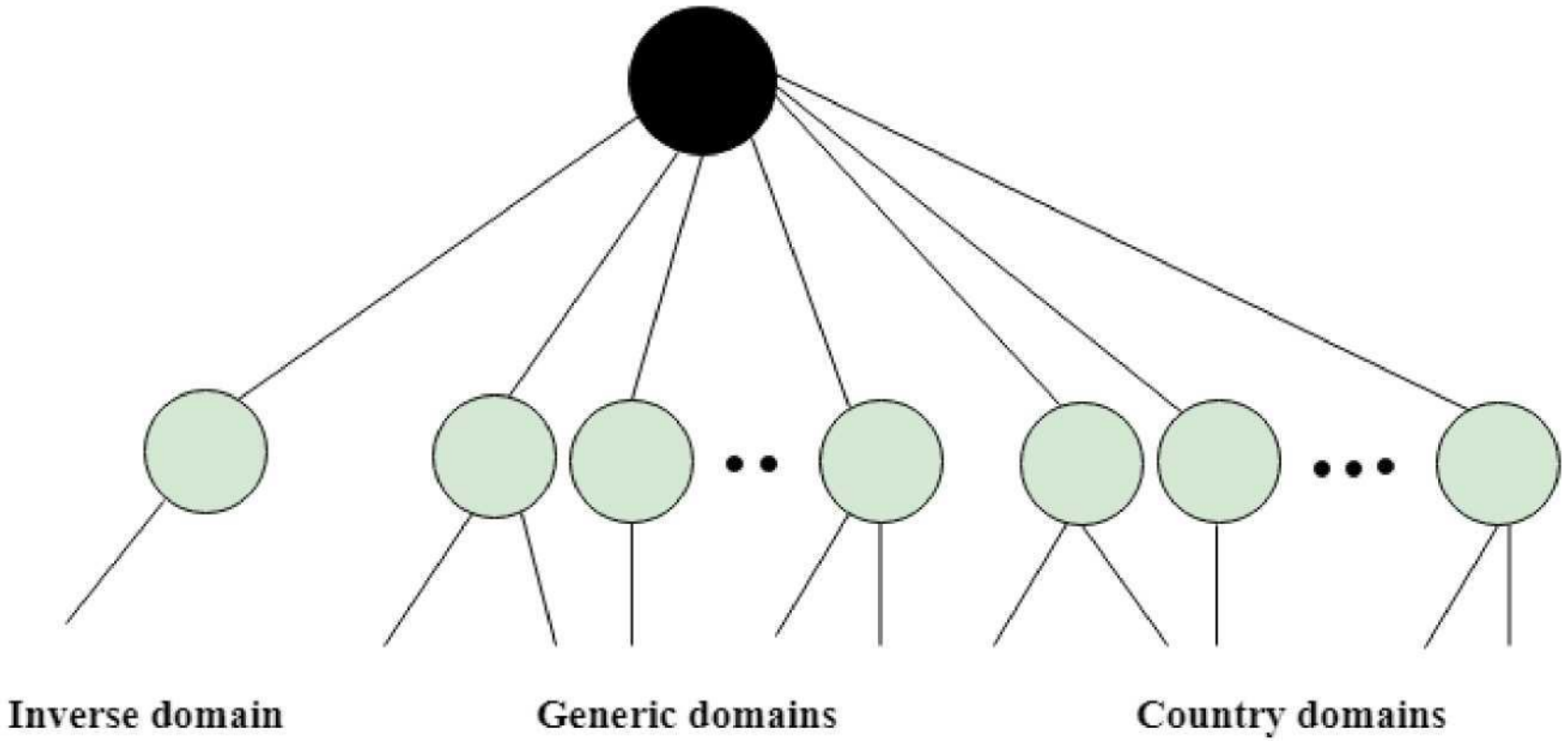
An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

- DNS stands for Domain Name System.
- DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.
- DNS is required for the functioning of the internet.
- Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.
- DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.
- For example, suppose the FTP site at EduSoft had an IP address of 132.147.165.50, most people would reach this site by specifying

APPLICATION PROTOCOLS

ftp.EduSoft.com. Therefore, the domain name is more reliable than IP address.

DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.



APPLICATION PROTOCOLS

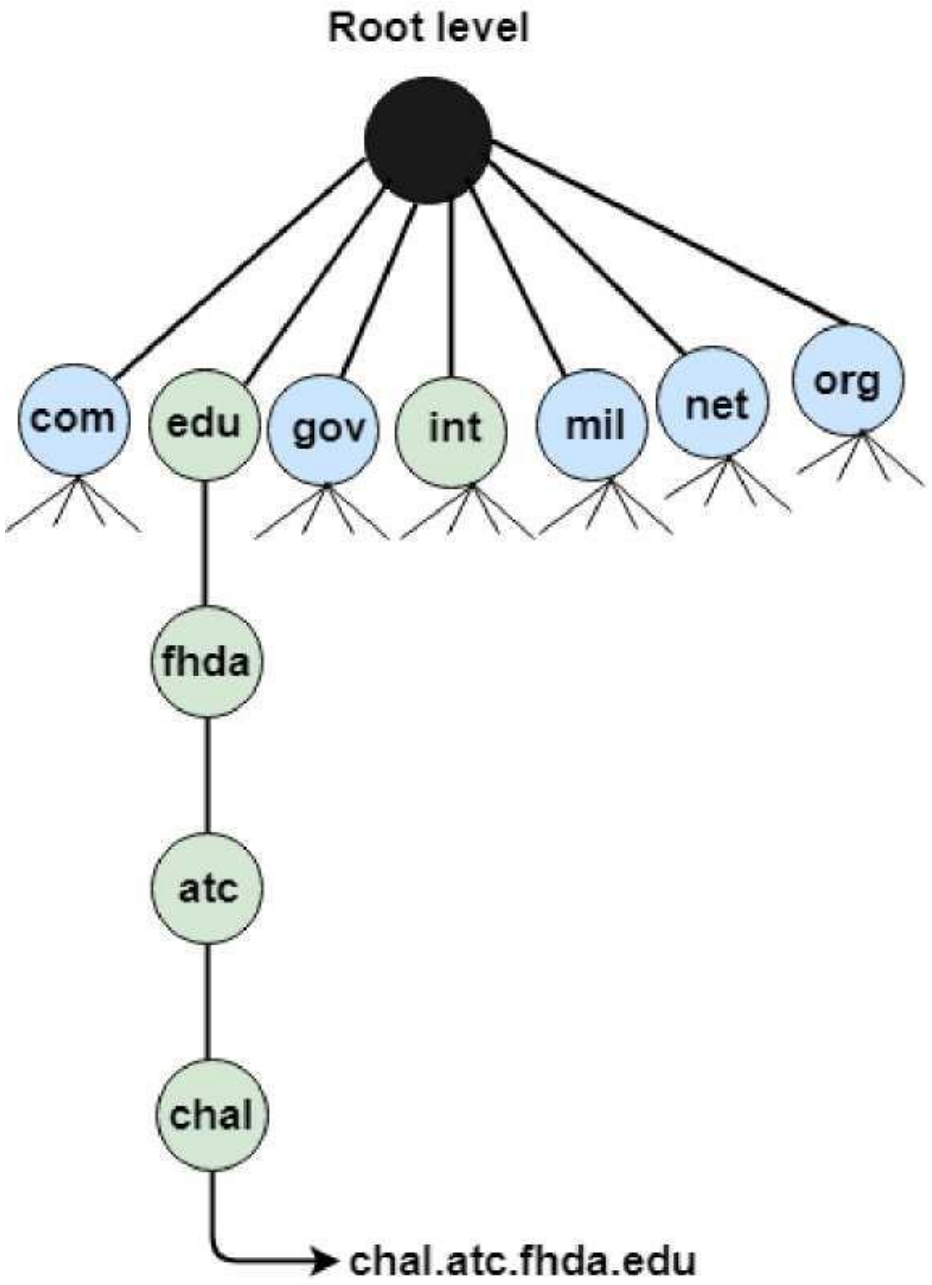
Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial Organizations
coop	Cooperative business Organizations
edu	Educational institutions
gov	Government institutions
info	Information service providers
int	International Organizations
mil	Military groups
museum	Museum & other nonprofit organizations
name	Personal names
net	Network Support centers
org	Nonprofit Organizations
pro	Professional individual Organizations

APPLICATION PROTOCOLS

Generic Domains

- It defines the registered hosts according to their generic behavior.
- Each node in a tree defines the domain name, which is an index to the DNS database.
- It uses three-character labels, and these labels describe the organization type.

APPLICATION PROTOCOLS



APPLICATION PROTOCOLS

Country Domain

The format of country domain is same as a generic domain, but it uses two-character country abbreviations (e.g., us for the United States) in place of three character organizational abbreviations.

Inverse Domain

The inverse domain is used for mapping an address to a name. When the server has received a request from the client, and the server contains the files of only authorized clients. To determine whether the client is on the authorized list or not, it sends a query to the DNS server and ask for mapping an address to the name.

APPLICATION PROTOCOLS

Working of DNS

- DNS is a client/server network communication protocol. DNS clients send requests to the. server while DNS servers send responses to the client.
- Client requests contain a name which is converted into an IP address known as a forward DNS lookups while requests containing an IP address which is converted into a name known as reverse DNS lookups.
- DNS implements a distributed database to store the name of all the hosts available on the internet.
- If a client like a web browser sends a request containing a hostname, then a piece of software such as **DNS resolver** sends a request to the DNS server to obtain the IP address of a hostname. If DNS server does not contain the IP address associated with a hostname, then it forwards the request to another DNS server. If IP address has arrived at the resolver, which in turn completes the request over the internet protocol.

APPLICATION PROTOCOLS

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.

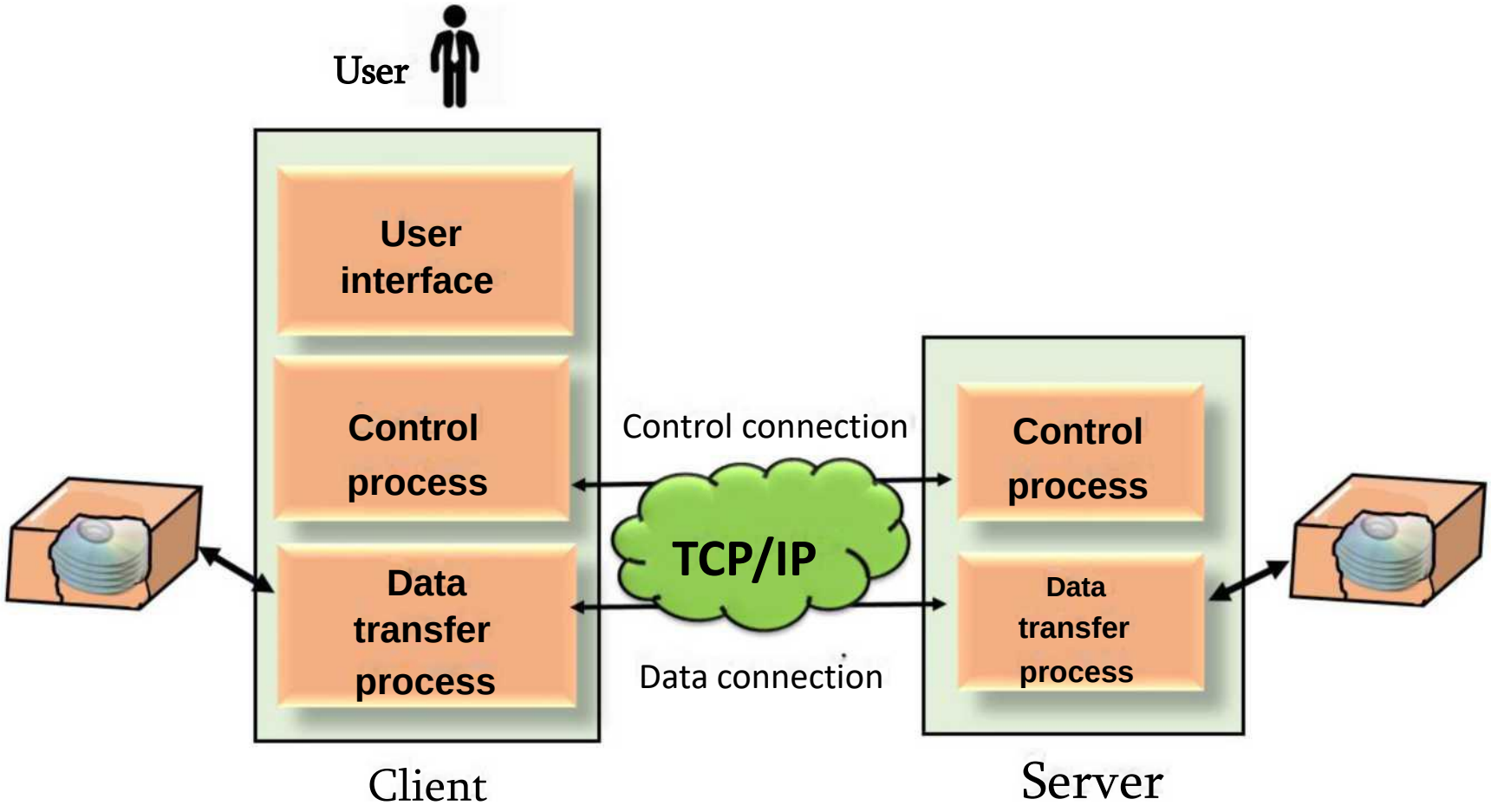
APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

Mechanism of FTP

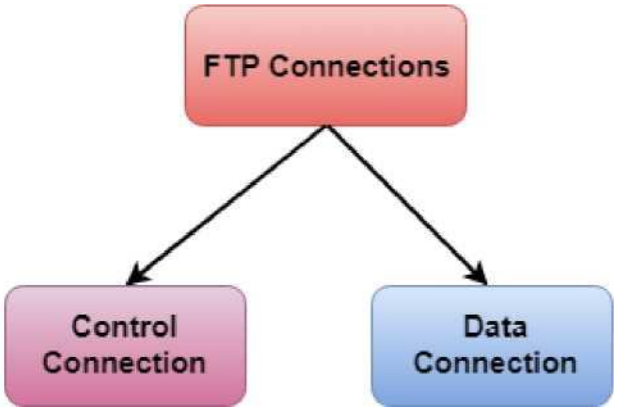


The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The

APPLICATION PROTOCOLS

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

APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

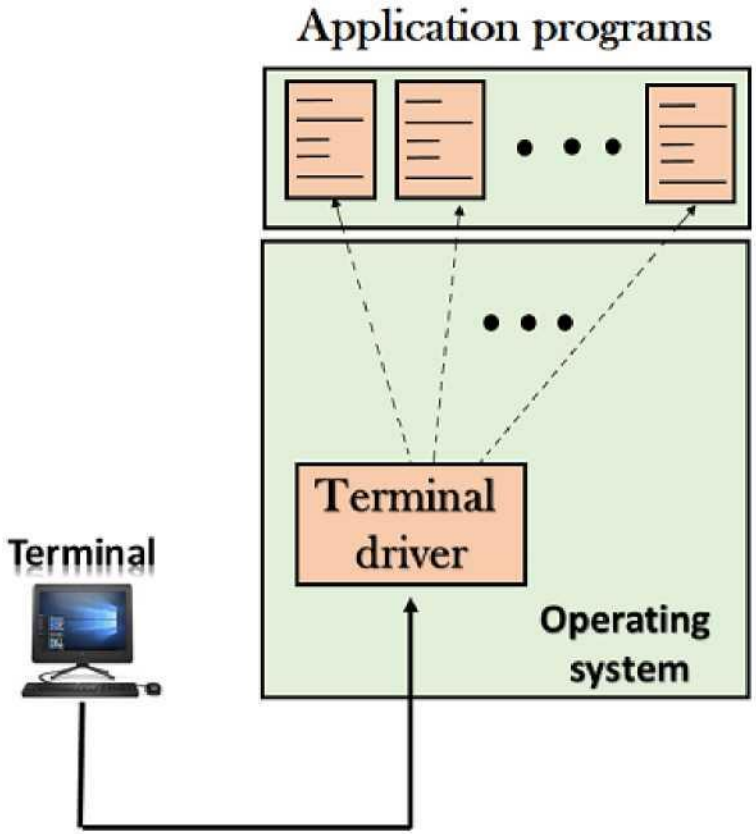
TELNET

- 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.

APPLICATION PROTOCOLS

There are two types of login:

1. Local Login



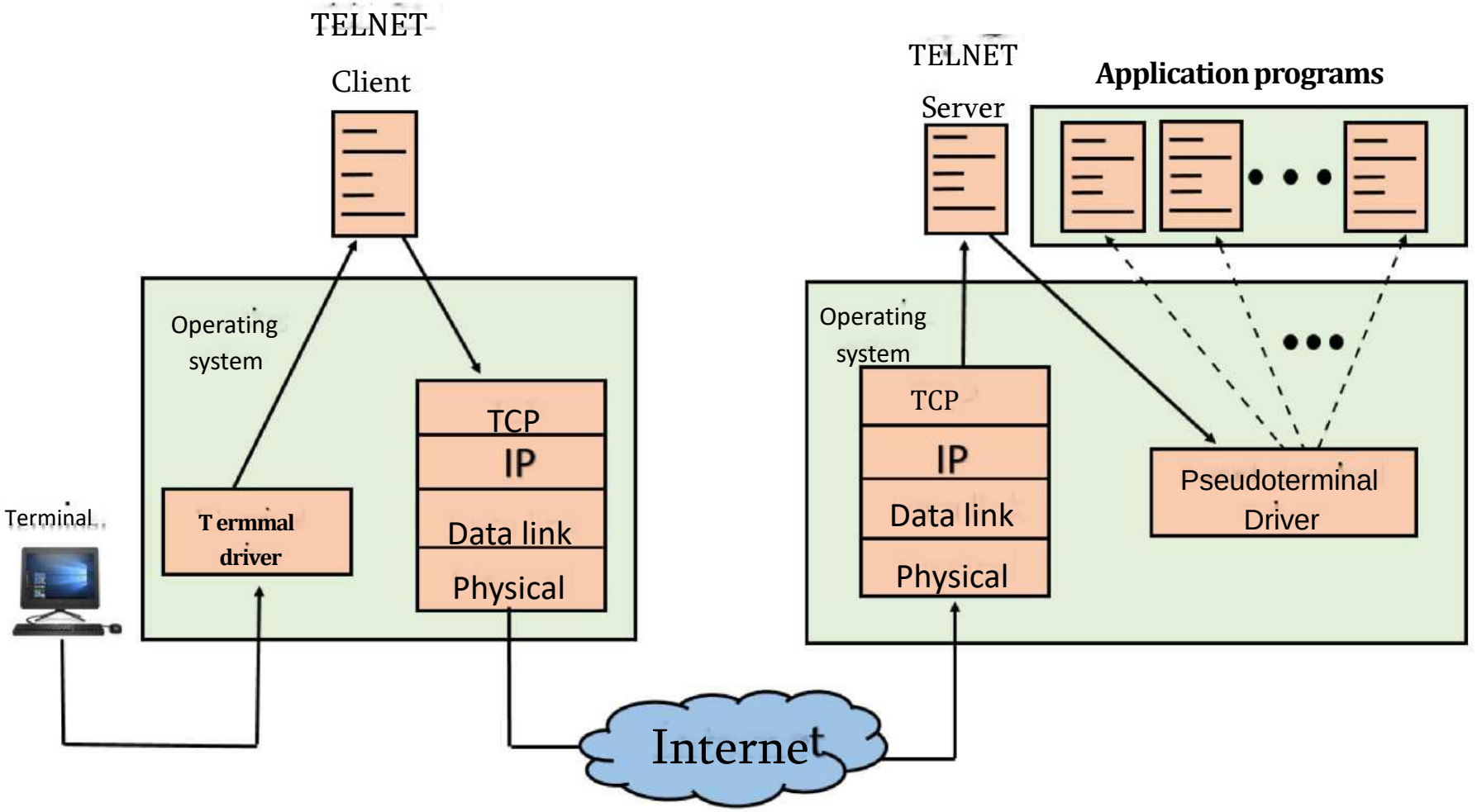
- When a user logs into a local computer, then it is known as local login,

APPLICATION PROTOCOLS

- 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.

APPLICATION PROTOCOLS

2.Remote login



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

APPLICATION PROTOCOLS

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

APPLICATION PROTOCOLS

the characters from the TELNET server. The operating system then passes these characters to the appropriate application program.

APPLICATION PROTOCOLS

SMTP

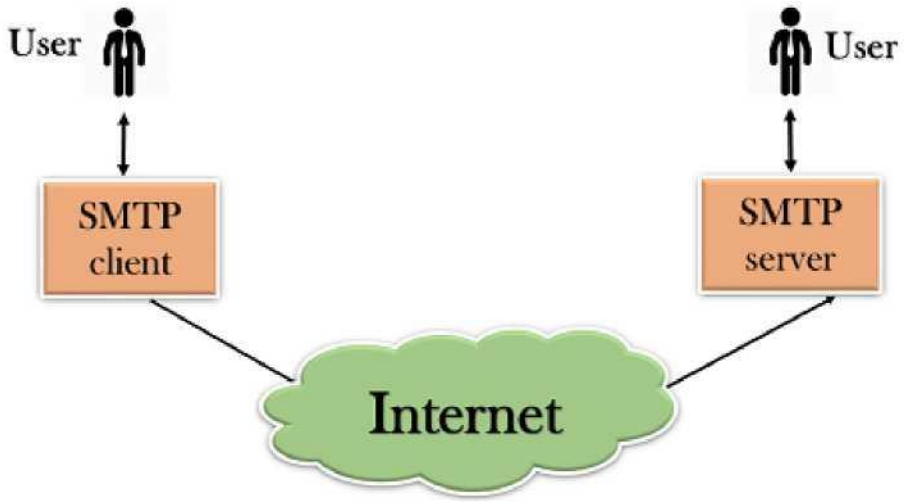
- SMTP stands for Simple Mail Transfer Protocol.
- SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called **Simple Mail Transfer Protocol.**
- It is a program used for sending messages to other computer users based on e-mail addresses.
- It provides a mail exchange between users on the same or different computers, and it also supports:
  - o It can send a single message to one or more recipients. o Sending message can include text, voice, video or graphics. o 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

APPLICATION PROTOCOLS

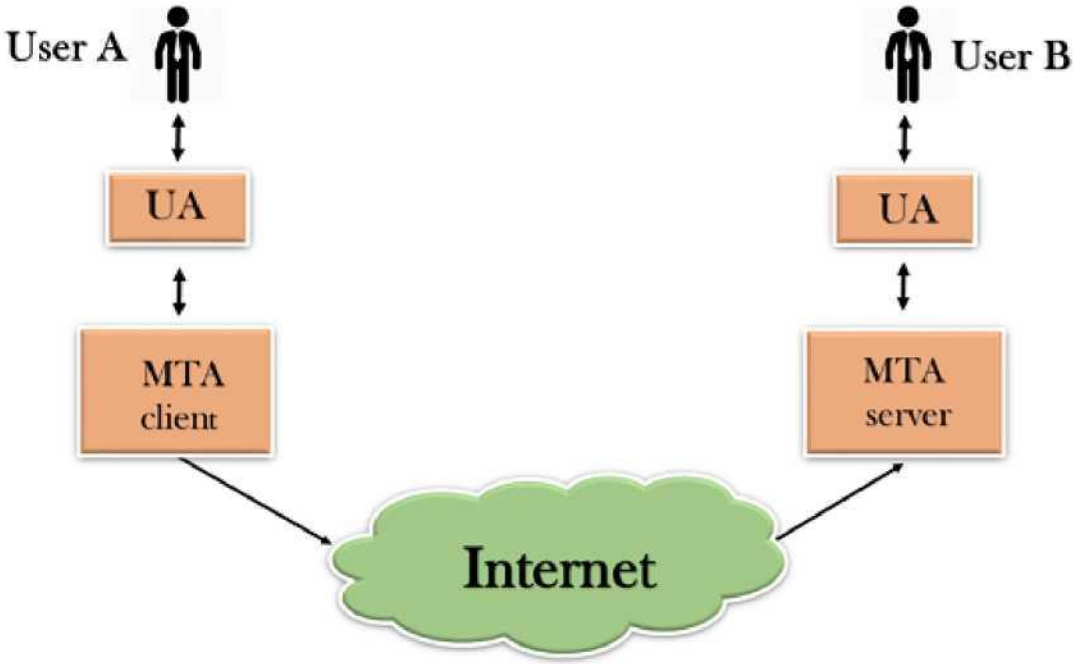
recipient address is wrong, then receiving server reply with an error message of some kind.

Components of SMTP



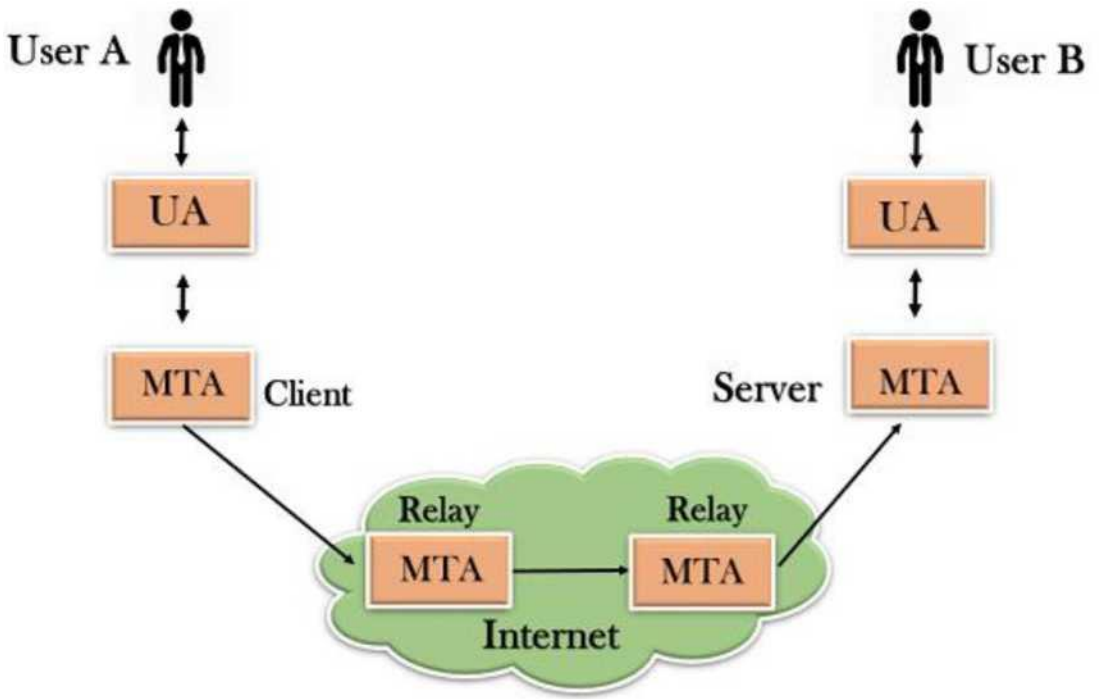
- First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.

APPLICATION PROTOCOLS



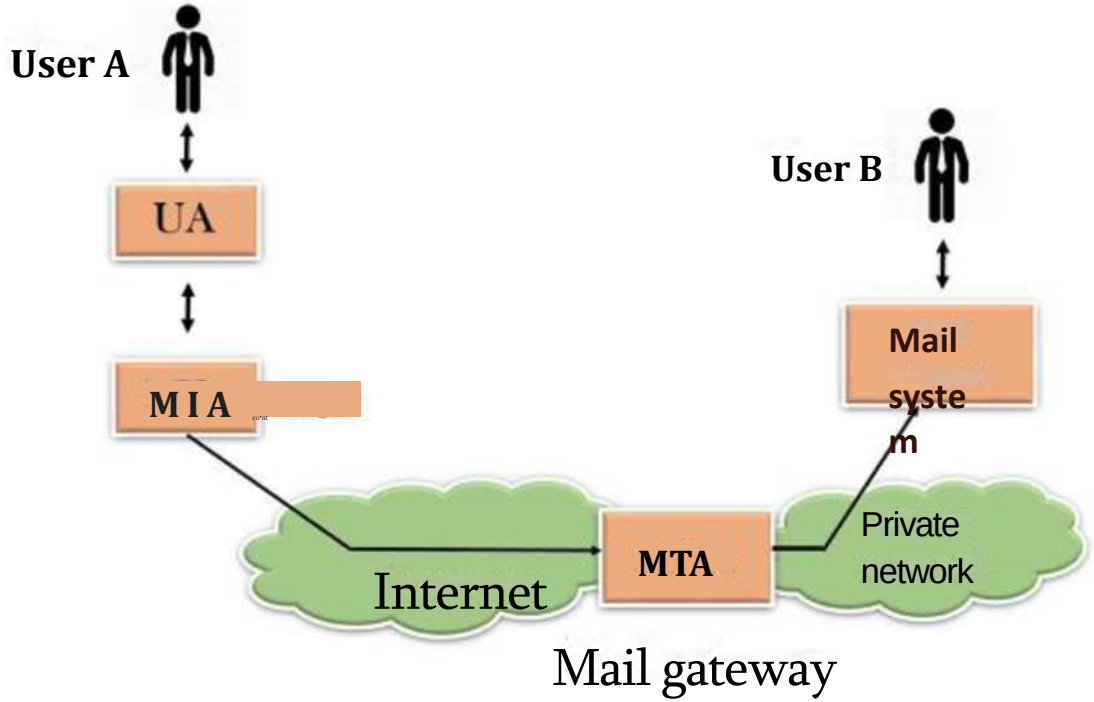
SMTP allows a more complex system by adding a relaying system. Instead of just having one MTA at sending side and one at receiving side, more MTAs can be added, acting either as a client or server to relay the email.

APPLICATION PROTOCOLS



. 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.

APPLICATION PROTOCOLS



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

APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

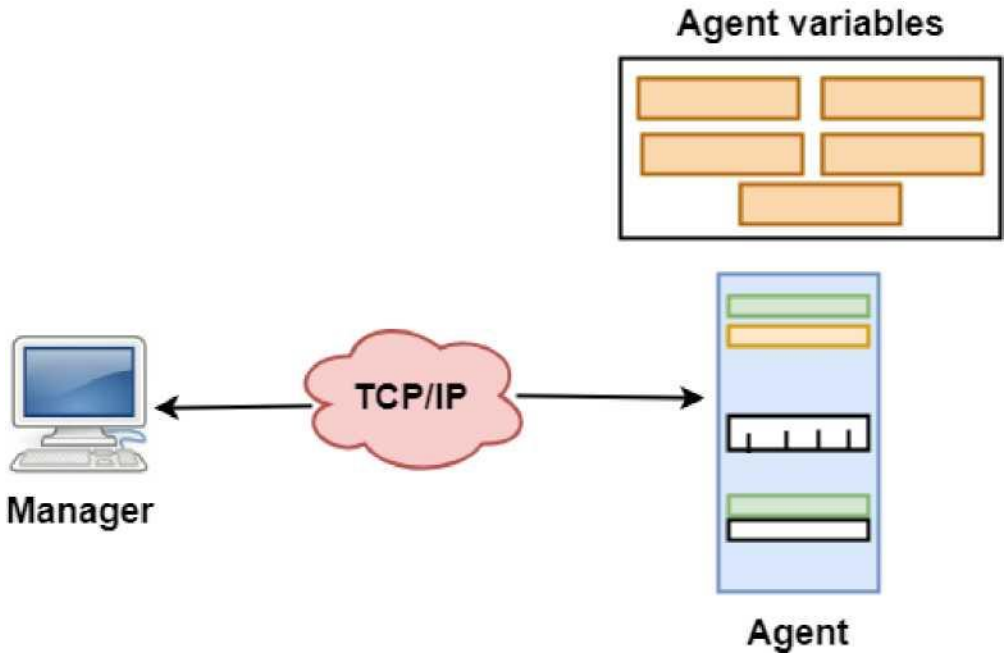
4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

APPLICATION PROTOCOLS

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



APPLICATION PROTOCOLS

. 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

APPLICATION PROTOCOLS

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.

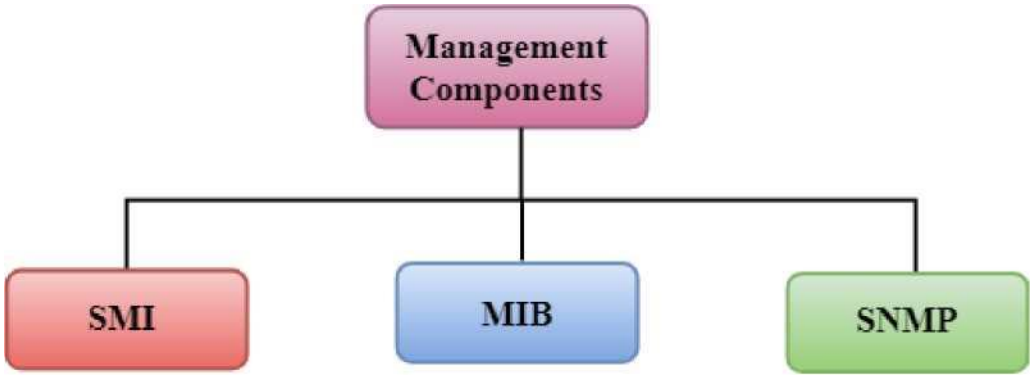
. An agent also contributes to the management process by warning the manager regarding an unusual condition.

APPLICATION PROTOCOLS

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).



APPLICATION PROTOCOLS

SMI

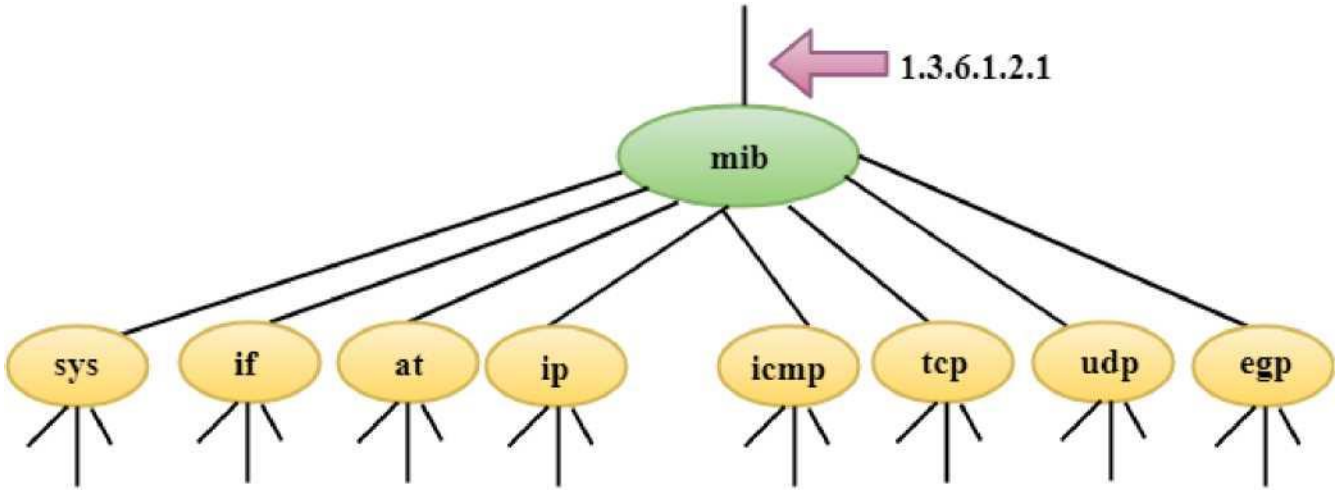
The SMI (Structure of management information) is a component used in network management. Its main function is to define the type of data that can be stored in an object and to show how to encode the data for the transmission over a network.

APPLICATION PROTOCOLS

MIB

. The MIB (Management information base) is a second component for the network management.

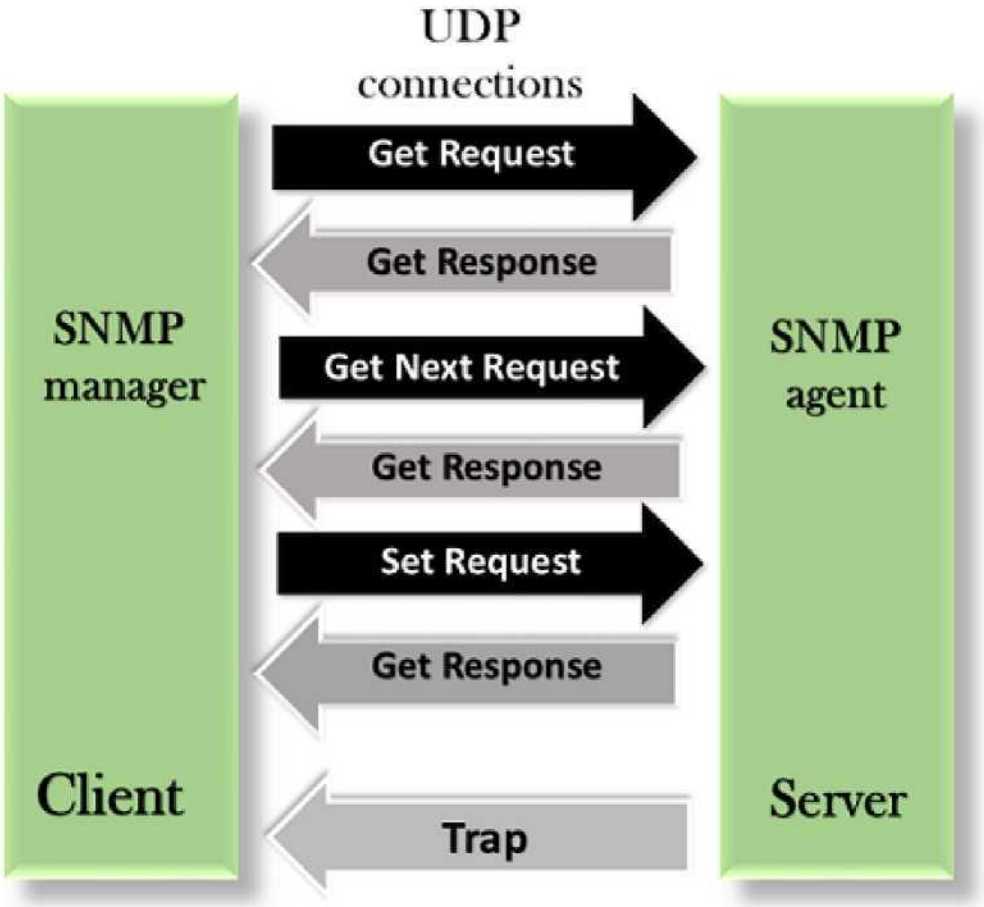
. Each agent has its own MIB, which is a collection of all the objects that the manager can manage. MIB is categorized into eight groups: system, interface, address translation, ip, icmp, tcp, udp, and egp. These groups are under the mib object.



APPLICATION PROTOCOLS

SNMP Messages

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



APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

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.

APPLICATION PROTOCOLS

Features of HTTP:

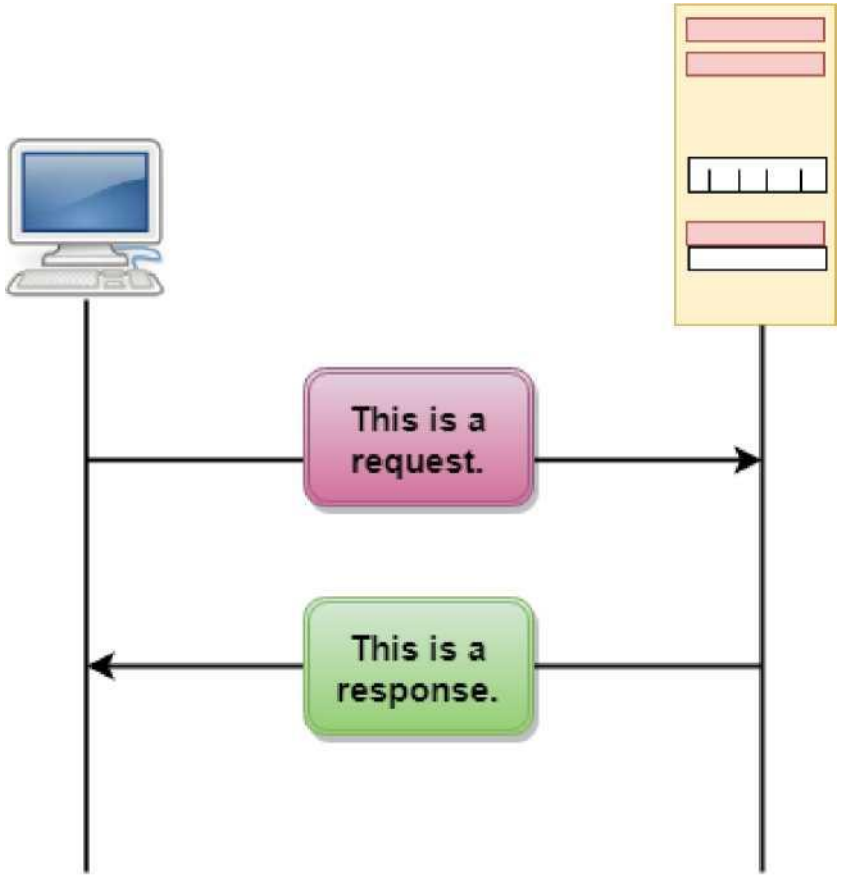
. **Connectionless protocol**: HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. 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.

APPLICATION PROTOCOLS

HTTP Transactions

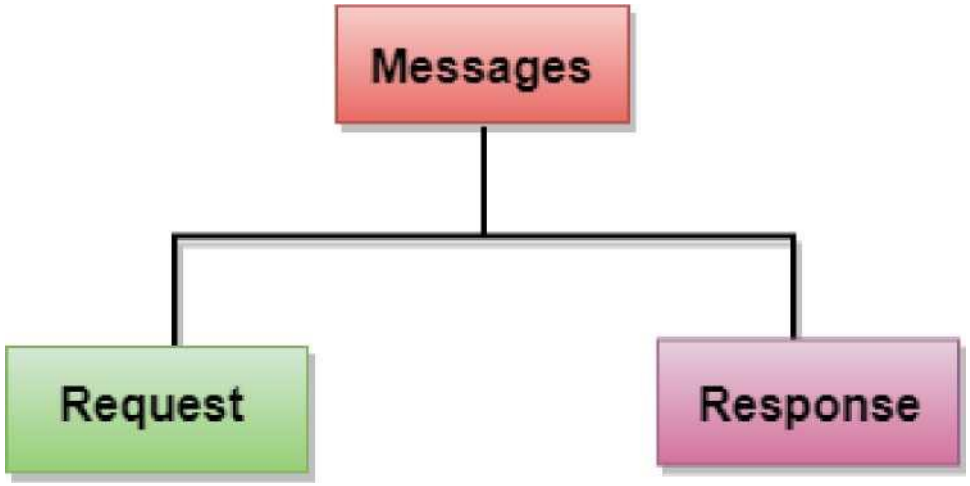


The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

APPLICATION PROTOCOLS

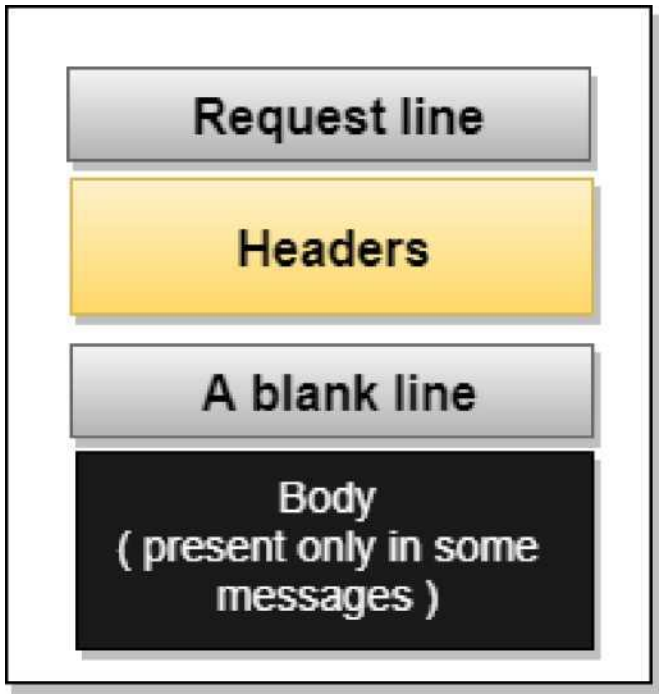
Messages

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



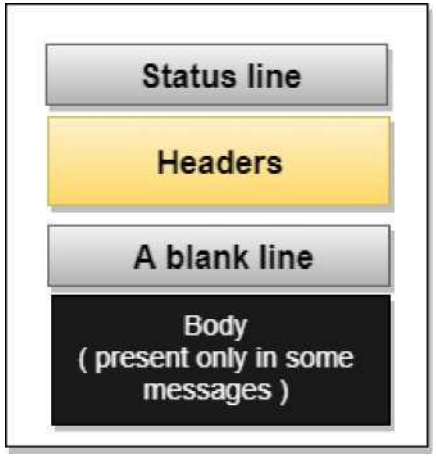
APPLICATION PROTOCOLS

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



APPLICATION PROTOCOLS

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



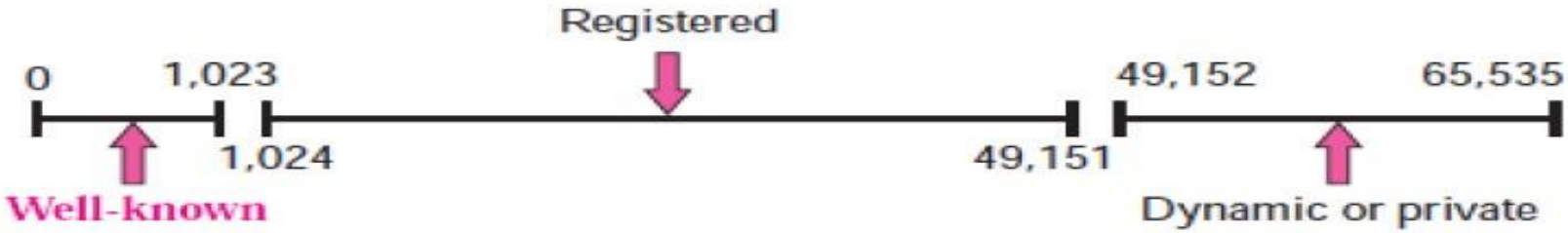
APPLICATION PROTOCOLS

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

Uniform Resource Locator

Method: The method is the protocol used to retrieve the document from a server. For example, HTTP.

APPLICATION PROTOCOLS

. **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.

APPLICATION PROTOCOLS

Other Application Layer Protocols

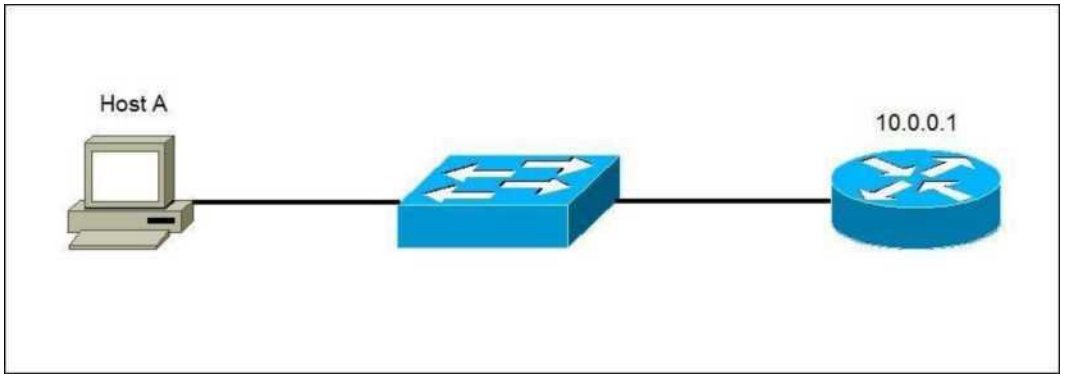
TFTP:

Trivial File Transfer Protocol (TFTP) is a network protocol used to transfer files between hosts in a TCP/IP network. It is the stripped-down, stock version of FTP and it doesn't have all of its functions; for example, you cannot list, delete, or rename files or directories on a remote server. In fact, TFTP can only be used to send and receive files between the two computers. TFTP doesn't support user authentication and all data is sent in clear text. So It's a technology for transferring files between network devices and is a simplified version of FTP

The only real advantage that TFTP has over FTP is that it uses less resources. It is not widely used today, but Cisco does still use it on its devices, for example to backup a router's IOS image.

APPLICATION PROTOCOLS

Consider the following example:



A user wants to transfer files from Host A to the router R1. R1 is a Cisco device and it has a TFTP server installed. The user will start an TFTP client program and initiate the data transfer.

TFTP uses a well-known UDP port 69.

Command

tftp [options...] [host [port]] [-c command]

APPLICATION PROTOCOLS

NFS:

It stands for network file system.It allows remote hosts to mount file systems over a network and interact with those file systems as though they are mounted locally. This enables system administrators to consolidate resources onto centralized servers on the network.

Command

service nfs start

APPLICATION PROTOCOLS

LPD:

It stands for Line Printer Daemon.It is designed for printer sharing.It is the part that receives and processes the request. A "daemon" is a server or agent.

Command

lpd [-d] [-l] [-D DebugOutputFile]

APPLICATION PROTOCOLS

X **window**:

It defines a protocol for the writing of graphical user interface-based client/server applications. The idea is to allow a program, called a client, to run on one computer. It is primarily used in networks of interconnected mainframes.

Command

Run xdm in runlevel 5

APPLICATION PROTOCOLS

DHCP:

It stands for Dynamic Host Configuration Protocol (DHCP).It gives IP addresses to hosts.There is a lot of information a DHCP server can provide to a host when the host is registering for an IP address with the DHCP server. Port number for DHCP is 67, 68.

Command

clear ip dhcp binding {address | * }

APPENDIX

Difference between Network and Internet

Computers and their systems square measure difficult in their approach, and it gets doubly robust once you need to comprehend 2 terms associated with this subject that act already utilized in the regular language, those mentioned adequately during this article square measure Network and net, they will appear totally different from one another, and so they will seem like one another.

The most distinction between them comes in their definition; a Network could be a association of 1 or additional computers placed in associate surroundings, and also the Internet is that the relationship of computers connecting them from everywhere the planet.

The basic distinction between network and net is that the Network consists of pcs that area unit physically connected and may be used as a private computer yet on share data with one another. Conversely, the Internet could be a technology that links these little and huge networks with one another and builds a additional in depth network.

APPENDIX

Let's see that the difference between network and internet:

S.NO	*NETWORK*	*INTERNET*
1.	Network is defined as the group of two or more computer systems.	Whereas internet is the interrelationship of a few networks.
2.	The coverage of network is limited in comparison of internet.	While it covers large geographical area.
3.	It provides the link between many computers and network-enabled devices.	While it provide connection among many networks.
4.	The types of network are: LAN, MAN, WAN, CAN and HAM.	Whereas the types of internet is world wide web.
5.	Through network, hundreds or a few thousands of computer system can linked simultaneously.	While through internet, millions of computer system can linked simultaneously.
6.	It requires less number of hardware devices.	While it requires various hardware devices.

APPENDIX

Terminal emulator

A terminal emulator, terminal application, or term, is a computer program that emulates a video terminal within some other display architecture. Though typically synonymous with a shell or text terminal, the term terminal covers all remote terminals, including graphical interfaces. A terminal emulator inside a graphical user interface is often called a terminal window.

A terminal window allows the user access to a text terminal and all its applications such as command-line interfaces (CLI) and text user interface (TUI) applications. These may be running either on the same machine or on a different one via telnet, ssh, or dial-up. On Unix-like operating systems, it is common to have one or more terminal windows connected to the local machine.

Terminals usually support a set of escape sequences for controlling color, cursor position, etc. Examples include the family of terminal control sequence standards known as ECMA-48, ANSI X3.64 or ISO/IEC 6429.

APPENDIX

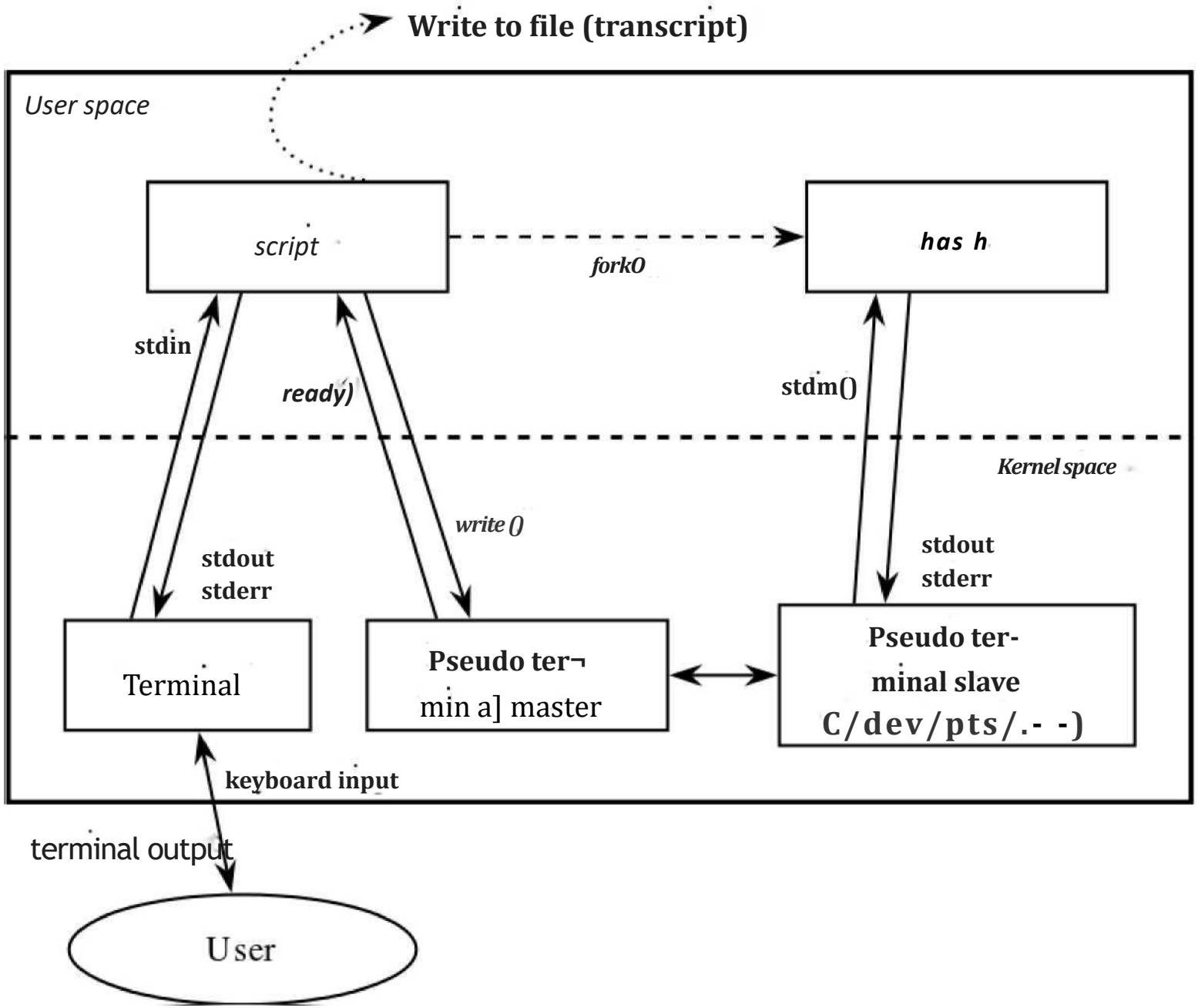
POSIX

The Portable Operating System Interface (POSIX) is a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. POSIX defines the application programming interface (API), along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems.

In some operating systems, including Unix, a pseudoterminal, pseudotty, or PTY is a pair of pseudo-devices, one of which, the slave, emulates a hardware text terminal device, the other of which, the master, provides the means by which a terminal emulator process controls the slave.

The PTY feature is part of POSIX and the Single Unix Specification in the form of a posix_openpt() function since 1998.

APPENDIX


Write to file (transcript)
User space
script
forkO
has h
stdin
ready)
stdm()
Kernel space
write 0
stdout
stderr
stdout
stderr
Terminal
Pseudo ter¬
min a] master
Pseudo ter-
minal slave
C/dev/pts/.- -)
keyboard input
terminal output
User

APPENDIX

In some operating systems, including Unix, a pseudoterminal, pseudotty, or PTY is a pair of pseudo-devices, one of which, the slave, emulates a hardware text terminal device, the other of which, the master, provides the means by which a terminal emulator process controls the slave.

The PTY feature is part of POSIX and the Single Unix Specification in the form of a posix_openpt() function since 1998.[1]

APPENDIX

PuTTY

PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection. It can also connect to a serial port. The name "PuTTY" has no official meaning.

PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix-like platforms, with work-in-progress ports to Classic Mac OS and macOS, and unofficial ports have been contributed to platforms such as Symbian, Windows Mobile and Windows Phone.

PuTTY was written and is maintained primarily by Simon Tatham, a British programmer.

APPENDIX

PuTTY supports many variations on the secure remote terminal, and provides user control over the SSH encryption key and protocol version, alternate ciphers such as AES, 3DES, RC4, Blowfish, DES, and Public-key authentication. PuTTY supports SSO through GSSAPI, including user provided GSSAPI DLLs. It also can emulate control sequences from xterm, VT220, VT102 or ECMA-48 terminal emulation, and allows local, remote, or dynamic port forwarding with SSH (including X11 forwarding). The network communication layer supports IPv6, and the SSH protocol supports the zlib@openssh.com delayed compression scheme. It can also be used with local serial port connections.

PuTTY comes bundled with command-line SCP and SFTP clients, called "pscp" and "psftp" respectively, and plink, a command-line connection tool, used for non-interactive sessions.