Contents

Unit 4. Application Layer -2

Unit 4
The Application Layer 2

Structure of the Unit

4.1 Unit Outcomes

After the successful completion of this unit, the student will be able to:

- Explain the working of the Mail Transfer Protocol.
- Describe the Domain Name Service

4.2 Introduction

Electronic Mail or E-mail as the name says, is a method of communication that uses electronic devices to exchange messages across computer networks and is one also of the widely used applications on the Internet today. An E-mail system includes individual messages and a message delivery system.

The E-mail system was first programmed by Raymond Tomlinson, an American programmer for exchanging messages between two computers on the Advanced Research Projects Agency (ARPANET) system. The modern E-mail system allows users to exchange images, audio and video files, and hyperlinks along with the textual content in an email.

Today there are various mail protocols running at the application layer such as Simple Mail Transfer Protocol (SMTP) for sending mail, and Post Office Protocol Version 3 (POP3) or Internet Mail Access Protocol (IMAP) for receiving mail over the Internet.

The important services provided by the E-mail system are:

- Composition of a mail through an editor.
- Transferring mail with the help of various software elements called agents.
- Reporting errors if any related to mail delivery, rejection, or loss.
- Displaying or presenting a mail.
- Disposition of the mail after reading or saving the mail for later use.

An E-mail system provides the following benefits to a user:

1. It is a very fast and convenient method of communicating with individuals or a group, residing anywhere in the world.
2. E-mail system provides an asynchronous mode of communication and users can send and receive messages at their own convenience. Messages can be stored easily and used later in time.
3. Attachments such as documents, images, and videos can be sent and received along with text messages.
4. It is very cost-effective in comparison with traditional mail and fax.
5. Service is available anywhere and anytime.

Though the E-mail system is a very useful method of communication, it has some limitations also:

1. There are risks of spam and phishing attacks.
2. Increased use can lead to decreased face-to-face communication and loss of personal touch.
3. Server outages can disrupt email service.
4. Miscommunication can occur if not used effectively.

In this unit, the first section discusses the working of the Internet E-mail system. In the second section, SMTP: the prominent protocol used to send messages on the Internet, and the formats used to send them is studied. The next section discusses the two protocols used to receive these messages such as POP3 and IMAP. The next section deals with the MIME protocol that is used as an extension to SMTP to enhance its features. The last section deals with another popular protocol used at the application layer, the Domain Name System (DNS) which is used to convert human-understandable host names to router-understandable IP addresses.

4.3 Working of an E-mail System

Similar to a postal mailing system, the email system includes two entities: a sender, and a recipient. A sender is a person who is sending a mail and the recipient is the one who receives it. Also, similar to the way a letter travels through several post offices before reaching the recipient's mailbox, the email messages sent from email clients, are routed through multiple servers called Mail Transfer Agents (MTAs) before they reach the recipient's email server.

Before we discuss the working of an E-mail system and the various elements involved in it, let us look into the three important components used for exchanging messages on the Internet.

1. An email address: A unique identifier used to identify each user on the Internet written in the following format.
   name@domain.com
2. An email client: A software program to send and receive messages.
   E.g.: Outlook and web browser (email client software) and Gmail (web-based email client software)
3. An email server: A computer that is used to store and forward mail messages to the recipient. Mail servers of any user are shared by other users and maintained by the user's Internet Service Provider (ISP).

4.4 Simple Mail Transfer Protocol (SMTP)

This protocol forms an important application layer protocol of the Internet Electronic Mail and uses Transmission Control Protocol (TCP) which is a reliable data transfer protocol at the transport layer. It transfers the messages from the senders' mail servers to the recipients' mail servers. For sending a message using SMTP, the body of the message has to be encoded into simple 7-bit ASCII. So, to send large images and other multimedia data like audio and video files, the data has to be encoded to ASCII while sending and decoded back to binary while receiving. But this restriction is removed with the use of HTTP protocol.

Let us study the process of message transfer with the help of a scenario where Ram wants to send a message to Shyam.

1. Ram composes a message, provides Shyam's address and invokes his user agent or mail client.

2. The user agent then sends the message to his mail server where it is placed in a message queue.

3. The client side of SMTP also called a Mail Transfer Agent (MTA) initiates an SMTP connection (on port 25 or 587) to the SMTP server running on Shyam's mail server.

4. The SMTP then checks Shyam's email address and then uses the Domain Name System to translate this address to an IP address that is recognizable by the network devices.

5. The SMTP will then search for the mail exchange server which is associated with Shyam's domain name and forwards the mail to that server.

6. The mail is stored there until the user accesses it using the POP3 or IMAP protocols. The POP protocol downloads the email to Shyam's device and deletes it from the mail server, but the IMAP protocol stores the email within the email client, allowing the recipient to access it from any connected device.

7. Shyam invokes his user agent whenever he wishes to read the message.

   Fig. 4.1 illustrates how mail is transferred from a Sender to a Receiver

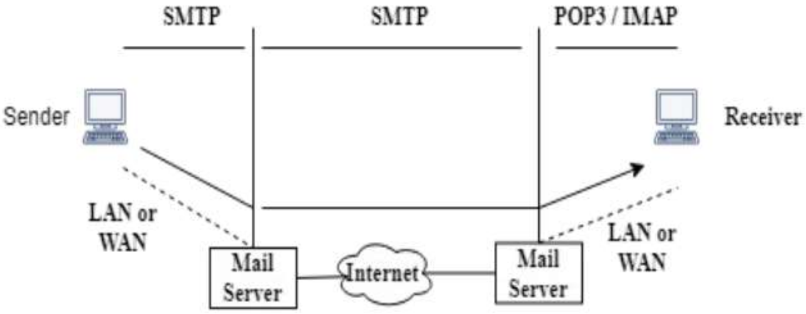


Fig. 4.1: Transfer of mail from Sender to Receiver

The following section discusses the process of mail transfer by considering the different software components called agents.

The four components of an E-mail system are as given below:

1. The User Agent (UA) – This is a program that helps a user in composing, receiving, and sending replies to messages.
2. The Message Transfer Agent (MTA) – It is basically a mail server. It is one of the essential elements in the mail delivery process. It helps in transferring messages to the recipient's mailboxes using SMTP.

   Also, in case of errors in sending or mail not having reached the destination, the MTA sends auto-response messages.
3. A Mail Box – This is a storage area where the delivered mail is stored. Each user will have a mailbox and sole access to it. The user can delete the emails if required.
4. A Spool File – This file contains the files that are to be sent or is known as the mailing list. The User-agent will save the messages in this file and the MTA will select them for delivery. Also, they hold the mail until the client retrieves it.

Fig. 4.2 shows the agent equivalent of the E-mail system shown in Fig. 4.1

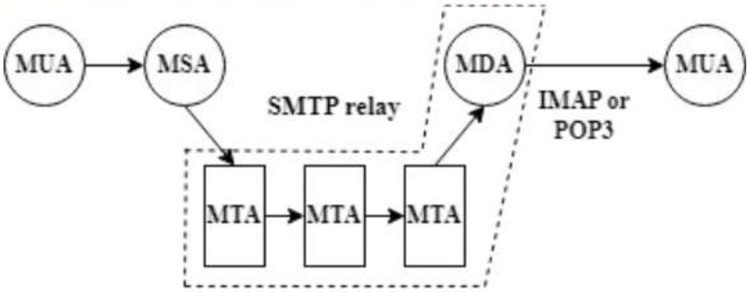


Fig. 4.2: Role of MTA in mail transfer

The mail user agent (MUA) also known as an email client is an application program that handles the mail-related issues for the user. It receives the email from the user and forwards it to the message submission agent (MSA). The MSA will then forward it to the MTA. The MTA will transfer the mail to the recipient's mailbox if it is local, else the mail will be forwarded to other MTAs until it reaches the Message delivery agent (MDA). The MDA transfers the mail to the recipient's mailbox. The sending of the email is carried out using SMTP protocol (or extended SMTP) which is only a push protocol, but for the receiving of the email (MDA to MUA), POP3 or IMAP4 protocol is used, which are discussed in the next section.

4.4.1 Mail Message Format

An Email message consists of three parts:

1. The Envelope
2. The Header
3. The Body

1. The Envelope – This part is used by the MTA for routing the message as it contains all the information such as the destination address, priority, and the security level needed.
2. The Header – This part contains several lines of ASCII text each giving a field name with a colon and the following information to be filled by the user.

   The main header fields are given below:

   1. To: The mail address of the primary recipient(s).
   2. Cc: This field specifies the address to whom a copy has to be sent, a secondary recipient (a carbon copy)
   3. BCC: This field specifies the address of a third party to whom a copy has to be sent without the primary and the secondary recipients knowing about it (Blind carbon copy).
   4. From: Field specifying the name of the sender
   5. Received: Gives the time the message was received and the sender data and other information used to find the route of the message
   6. Return Path: This information helps in finding the route back to the sender.
3. The Body: This part of the mail contains the actual message in the form of text, audio or video files, and images. Signatures or text that is automatically generated.

The email message contains other headers to specify the date and time of the message, a reply to field indicating the addresses to whom the message should be replied, a unique message ID, In-Reply-to field indicating the message ID of a message to which that would be a reply, some keywords, and a subject header indicating a short summary of the message for displaying it online.

4.5 Mail Access Protocols

These are pull-type protocols that are used to transfer mail from the recipient's mail server to the recipient's user agent. There are many popular mail access protocols like POP3, IMAP, and HTTP.

4.5.1 Post Office Protocol - Version 3 (POP3)

It is a very simple protocol but with limited features. When a mail recipient wants to access his mailbox, the recipient's user agent opens a TCP connection to the mail server on port 110. After this connection phase, the POP3 protocol works in three phases: authorization, transaction, and update.

1. Authorization – In this first phase, the username and password is sent by the user agent for the authentication of the user.
2. Transaction – In this phase, the messages are extracted by the user agent who also can mark the messages for deletion or remove the deletion marks.

3. Update - The third phase begins when the client issues the quit command and the POP3 session ends. During this phase, the message with the delete mark is deleted.

The POP3 transactions take place through commands given by the user agent to a server and the server replying to the command.
The user agent sends two commands while sending the username and password during the authorization phase as user <username> and pass <password>.
There are two responses by the server: +OK followed by some data when there is no error and -ERR when there is an error with a command.

The POP3 commands for the three phases can be illustrated with examples as given below.
1. Commands used in the authorization phase:
Here the POP3 server is invoked through telnet using port 110 and the name of the mail server is mymailserver.

```
telnet mymailserver 110
+OK POP3 server ready
user Mary
+OK
 pass cycle
+OK user successfully logged on
```

NOTE: If any command is misspelled, the server will respond with an -ERR message.

2. Commands used in the transaction phase:
Here, it is assumed that the user has two messages in his or her mailbox and the user agent has been configured to download and delete the messages by the user.
The list command is used to list the size of the three messages, the retr command is used to retrieve the messages, the dele command is used to delete the message, and finally, the quit command allows the server to move into the update phase, remove the messages from the mailbox.
Here ‘C’ indicates a client and ‘S’ indicates a server.

```
C: list
S: 1 200
S: 2 400
S: .
C: retr 1
S: ….
S ….
S: .
C: dele 1
C: retr 2
S: ….
```

S: ….
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

A POP3 server keeps track of which user messages have been deleted by maintaining some information, but it does not carry the state information across the sessions. This makes the POP3 implementation simpler.

4.5.2 Internet Mail Access Protocol (IMAP)

This protocol has more features compared to the POP3 protocol and hence is more complex to implement both on the client and server sides. With POP3, the user can only download the messages and then move them across folders or delete them only locally, but cannot perform these operations remotely. On the other hand, using the IMAP protocol, the user can create folders, move messages between folders, and read them. It has commands which help the user to search for messages based on certain criteria. Also, this protocol maintains the user state information across the IMAP sessions, such as the names of the folders and the different messages associated with these folders. Another important feature of IMAP is that it allows the user to extract only some component or part of a message when the bandwidth between the client and the mail server is low.

4.5.3 Multipurpose Internet Mail Extension (MIME)

It is a protocol that extends the limited capabilities of its parent protocol SMTP and is therefore an application layer protocol. It is basically used to describe the type of content in a message. MIME does not work independently but works in collaboration with SMTP and allows users to exchange audio, video, images, and non-ASCII types of textual data (text in languages other than English). The MIME standard was proposed by Bell Communications in 1991. Today, most of the mail messages on the Internet are transferred via SMTP in MIME format. Though MIME was designed for SMTP, the content type defined using MIME format can be used by the Hypertext Transfer Protocol (HTTP) for the World Wide Web (WWW).

Fig. 4.3 shows the functional block diagram showing the use of MIME.

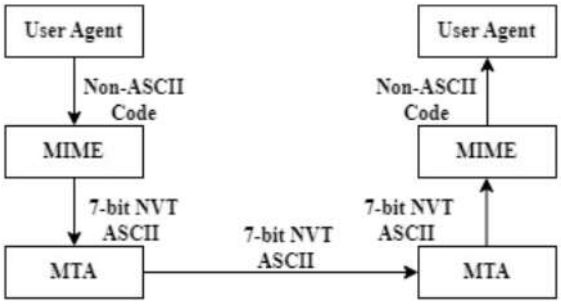


Fig. 4.3: Use of MIME Protocol at the Application Layer

The MIME component at the sender receives non-ASCII data from the user agent, converts it into 7-bit NVT ASCII, and passes it to the MTA or the sender mail server. This MTA routes the message to the receiver side mail server or the MTA. The MIME component at the receiver converts the 7-bit ASCII data to the original non-ASCII code to the user agent.

The other features of MIME are given below:

- Multiple attachments can be sent in a single message.
- MIME supports different content types and multi-part messages.
- It supports compound documents.

4.5.3.1 MIME Header

An e-mail has a header section to which a MIME header is added in order to define the transformation of the non-ASCII data to ASCII.

The five headers added to the original E-mail header are given below:

1. Version of MIME – This header defines the version of the MIME protocol used. The current version is 1.1
2. Type of content – This header defines the type of data used in the body of the message such as audio, video, image, etc. It also defines the sub-type of the content such as JPEG or GIF format in the case of image data along with some other parameters.
3. Encoding of content transfer – This header defines the encoding schemes in order to convert the data into bits (0s and 1s) to be transferred on the medium. Encoding schemes can be 7-bit, 8-bit, binary, etc.
4. Identification or ID of the content – In the case of a multi-message environment, the message ID is used to identify a whole message.
5. Description of the content – This header describes the content of the body of the message.

4.6 Domain Name Service (DNS) – The Internet's Directory Service

Host names on the Internet are easier to identify and remember by humans if they are in a mnemonic form or a more readable format such as www.yahoo.com or www.myschool.edu. But this format consisting of alphanumeric characters with variable lengths is difficult to process by the routers. So hosts are given IP addresses for identification.

An IP address uses a four-byte dotted decimal notation and has a fixed hierarchical structure (will be discussed in detail in Unit 7). This address gives more specific information about the location of the host on the internet if we scan it from left to right.

4.6.1 Services Provided by DNS

As humans prefer mnemonic hostnames and routers prefer hierarchical, fixed-length, structured IP addresses, there should be a directory service that will help in translating hostnames to IP addresses. This job is done by the Domain name system.

A DNS consists of a distributed database in the form of a group of DNS servers structured in a hierarchy. It is an application layer protocol used by hosts to query the database. It uses port 53 and runs over UDP at the transport layer. HTTP, SMTP, FTP, and other application layer protocols use DNS to translate or map a user-given hostname to an IP address.

To understand the translation process, let us take an example of an HTTP client requesting the URL www.myschool.edu/page1.html. The steps taken by the translation process are given below:

1. The client side of the DNS application runs on the user's machine.
2. The above-requested hostname is extracted by the browser of the client side of the DNS application and sends a query to a DNS server.
3. The reply message by the server contains the IP address of the hostname.
4. The browser uses the IP address to initiate a TCP connection to the HTTP server at port 80.

The other important services provided by the DNS are listed below:

1. **Host aliasing** – Sometimes a hostname is very complicated and so for easy usage, it is given simple alias names for it. The real hostname is called a canonical hostname and communication can happen only with this name. So, DNS can be used to obtain the canonical name for the alias names.
2. **Mail server aliasing** - Sometimes the hostname of a mail server can be very complicated but mail addresses with an account with that server will be simple alias names for easy usage. So DNS can be used to obtain the canonical hostname for the alias names.
3. **Load distribution** – When a Web server becomes very busy with many requests, the responses get delayed due to a heavy load on it. To overcome this, a Web server is replicated and runs on many hosts with different IP addresses and alias host names. But all these web servers with a set of IP addresses and alias hostnames will be associated with a single canonical hostname.
   Whenever a client makes a DNS query for a hostname with multiple addresses, the server sends all the IP addresses. With every different reply, the server keeps rotating the addresses so that the load gets distributed among different servers and only the first IP address is not chosen always.

4.6.2 Structure and Working of DNS

The simplest structure of DNS can be a single DNS server containing all the hostname to IP address mappings in it. In this design, all the clients send their queries to this central server and the server responds to the clients directly. But there are many limitations with this structure given as follows:

1. If the server crashes, the whole system fails.
2. A single system cannot handle the traffic volume.
3. Frequent updation of a single database is needed to add new hosts.

A Distributed Hierarchical Database

In the hierarchical structure, the mappings are distributed across many servers. Here there are three classes of DNS servers:

- Root DNS servers
- Top-level domain (TLD) DNS servers
- Authoritative DNS servers

Root DNS servers: There are 13 root DNS servers on the internet and each of these is replicated for reliability and security purposes. There are around 247 root servers as of 2011.

Top-level domain (TLD) servers: These servers maintain the top-level domains such as com, org, edu, net, and gov and also country top-level domains such as uk, fr, ca, and jp.

Authoritative DNS servers: Most organizations that have their hosts accessible publicly store their hostnames to IP addresses or DNS records in their own authoritative DNS servers. Alternatively, they can pay and store them in an authoritative server maintained by a service provider.

Fig. 4.4 shows a portion of the hierarchy of DNS servers.

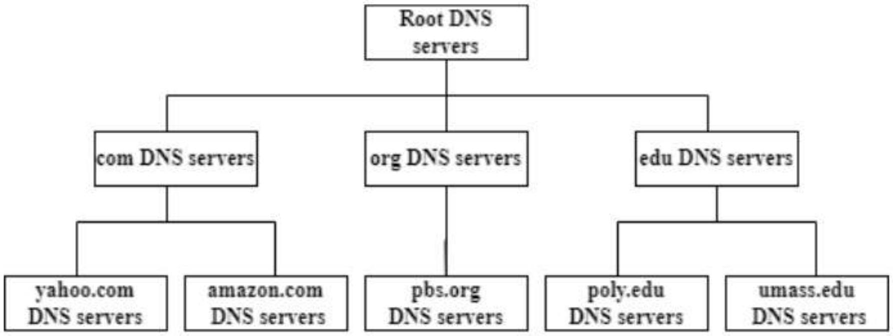


Fig. 4.4: Portion of the hierarchy of DNS servers

Local DNS Server:

This is another important type of DNS server that is not usually included in the hierarchy but is always used in the process of DNS operation. Local DNS are servers maintained by the ISPs of organizations, companies, educational institutions, residential connections, etc.

Whenever a host gets connected to his ISP, the ISP provides it with the IP addresses of one or more of its local DNS servers. The host then makes a DNS query to the local DNS server, which acts as a proxy and forwards the query to the server hierarchy (Fig. 4.4).

List of events showing the interaction between the three classes of servers for determining the IP address for the hostname xyz.cs.umass.edu using Iterative DNS query

1. First the local host wxy.poly.edu sends a DNS query message with the hostname xyz.cs.umass.edu whose IP address is needed to its local DNS server dns.poly.edu
2. The local DNS server forwards the query message to the root DNS server.
3. The root server returns the IP address of the top-level **edu** domain TLD server to the local DNS server.
4. With the IP address, the local DNS server contacts one of the TLD servers.
5. The TLD server returns the IP address of the authoritative server for umass.edu to the local DNS server.
6. With the IP address, the Local DNS server contacts one of the authoritative servers.
7. The authoritative server returns the IP address for the hostname xyz.cs.umass.edu

Fig. 4.5 shows the interaction of the various DNS servers using Iterative DNS Query.

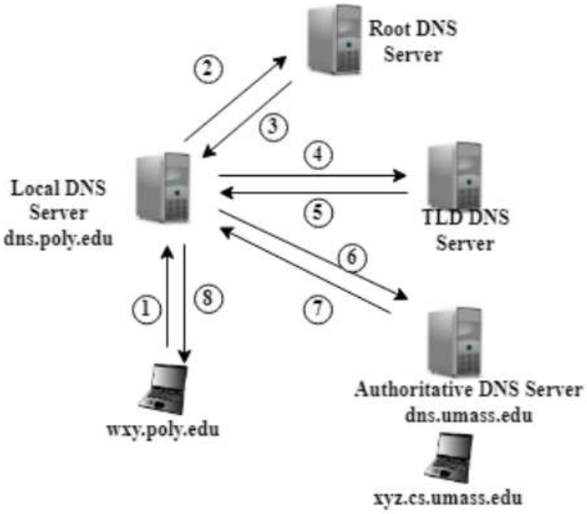

Root DNS Server
②
③
④
⑤
Local DNS Server dns.poly.edu
TLD DNS Server
⑥
⑦
①
⑧
wxy.poly.edu
Authoritative DNS Server dns.umass.edu
xyz.cs.umass.edu

Fig.4.5: Interaction of various DNS servers

Fig. 4.6 shows the interaction of the various DNS servers using Recursive DNS Query. Here, one DNS server communicates with many other DNS servers to fetch the IP address. So here, the communication for the query is between the client and the local DNS server.

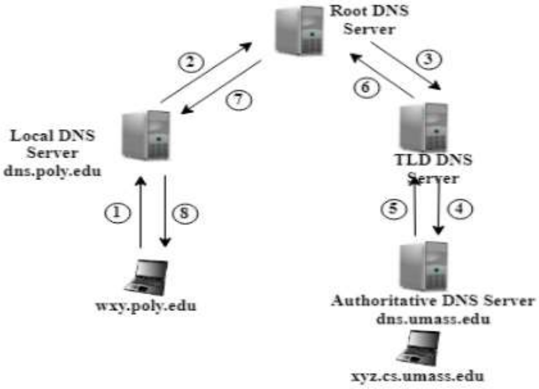

Root DNS Server
②
⑦
③
⑥
Local DNS Server dns.poly.edu
TLD DNS Server
①
⑧
⑤
④
wxy.poly.edu
Authoritative DNS Server dns.umass.edu
xyz.cs.umass.edu

Fig.4.6: Interaction using Recursive DNS Query

DNS Caching:

Fig. 4.6 shows that for obtaining the IP address of one hostname, there are four query messages and four reply messages exchanged. To reduce this query traffic, DNS caching is used.

In the chain of query messages, the Local DNS servers can cache or store the Hostname to IP address mapping in their local memory for some time before discarding it. If there is a query for the same hostname again within this cached time, it can return the IP address. A local DNS server can also cache the TLD server's IP address so that it can bypass the root DNS servers in the chain of queries.

4.6.3 DNS Records and Messages

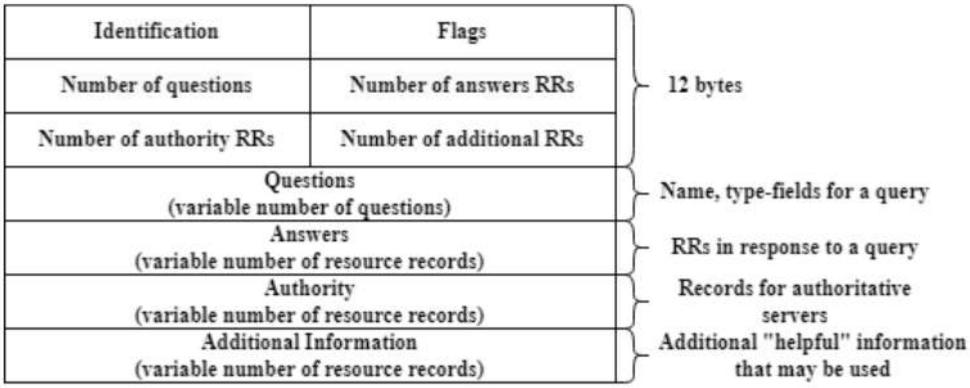
For the DNS to function, it needs a large database of records holding the hostname to IP address mappings. This database store also known as resource records is implemented through a large set of DNS servers. For every DNS query, there is a DNS reply message consisting of one or more resource records.

A resource record has four values: (Name, Value, Type, TTL)

1. Name is the hostname
2. Value is the IP address
3. TTL gives the time-to-live value of the resource record before which it will be removed from the cache.
4. Type gives the type of the hostname. There are four Type values: A, NS, CNAME, and MX
    - Type 'A' for a standard hostname-to-IP address mapping,
      E.g. (abc.xyz.slr.com, 125.57.83.128, A)
    - Type 'NS' if it is a domain name with the Value giving the hostname of the authoritative DNS server.
      E.g. (slr.com, dns.slr.com, NS)
    - Type 'CNAME' where the Value gives the canonical hostname for an alias hostname.
      E.g. (slr.com, xyz.abc.slr.com, CNAME)
    - Type 'MX' where the Value is the canonical hostname of a mail server that has an alias hostname Name.
      E.g (slr.com, mail.abc.slr.com, MX)

DNS Messages

There are two kinds of DNS messages: DNS query and DNS reply message. Both these messages have the same format as shown in Fig. 4.7



Identification	Flags
Number of questions	Number of answers RRs
Number of authority RRs	Number of additional RRs
Questions (variable number of questions)	
Answers (variable number of resource records)	
Authority (variable number of resource records)	
Additional Information (variable number of resource records)	

12 bytes
Name, type-fields for a query
RRs in response to a query
Records for authoritative servers
Additional "helpful" information that may be used

Fig. 4.7: DNS Message Format

The first 12 bytes of the header consist of the following fields.

1. Identification Field: It is a 16-bit unique identifier for every query. This ID is copied in the reply message for a query for the client to match it.
2. Flags: There are many flags in the flag field.
    - 1-bit query/reply flag - This is used to indicate whether it is a query or a reply. 0 indicates query and 1 indicates it is a reply message.
    - 1-bit authoritative flag – This is set when a DNS reply is from an authoritative server.
    - 1-bit recursion desired flag – This flag is set instructing that the DNS server should use recursion when the desired record is not found in it.

- 1-bit recursion available flag – This flag is set in a DNS reply when the DNS server has the recursion support.

3. Questions: This field gives all the query information.
4. Answers: This field holds the resource records from the reply by the DNS server.
5. Authority: This field holds the records of other authoritative servers.
6. Additional: This field holds other useful information such as a canonical hostname of a mail server in a DNS reply for a DNS query of type MX.

There is one command available on Unix and Windows platforms called **nslookup.** This helps us to send a DNS query message to a DNS server from the host on which we are working. The nslookup command can also be employed remotely from our own host by visiting the websites that allow us to do so.

Inserting Records into the DNS Database

Responses to DNS queries are possible only if the needed records are inserted in the DNS database. Let us study the process of inserting the records with a specific example:

Consider a startup company called 'Sunshine Networks' intending to create its own website in order to connect the company to the external world.

Steps to insert the DNS record for the website are given below:

- Register the domain name of the site sunshinenetworks.com at the registrar and also provide the names and IP addresses of the primary and secondary authoritative DNS servers.
- The registrar verifies the uniqueness of the domain name and enters the domain name into the DNS database,
- The names and IP addresses of the servers are entered into the TLD com servers in the following way.

  If the names are dns1.sunshinenetworks.com and dns2.sunshinenetworks.com, and the IP addresses are 222.222.222.1 and 222.222.222.2. , then the Type 'NS' and Type 'A' are also entered in the TLD.

  The registrar would enter the below resource records in the DNS system for the primary authoritative server

  (sunshinenetworks.com, dns1.sunshinenetworks.com, NS)

  (dns1.sunshinenetworks.com, 222.222.222.1, A)
- Also, the Type A resource record for the web server www.sunshinenetworks.com and the Type MX record for the mail server mail. sunshinenetworks.com is entered into the company's authoritative server.
- After the above steps, the clients will be able to send emails to the employees of the company.

NOTE: Today, there is an option known as UPDATE available with the DNS protocol to allow data to be added and deleted dynamically using DNS messages.

4.7 Peer-to-Peer Applications

One of the important requirements of a Client-server architecture is that the server should always be

ON and running (Section 3.3.1) to provide services to its clients. However, in a peer-to-peer architecture (P2P), peers can be connected to each other only during communication. Also, the server is owned and controlled by a service provider, whereas the peer machines such as laptops and desktops can be controlled by the users themselves.

There are two important P2P applications: File Distribution and Distributed Hash Table (DHT)

In this section, we will study the File Distribution application.

4.7.1 P2P File Distribution

To understand the benefits of P2P architecture in comparison with a client-server architecture, let us consider distributing a large file to a large number of hosts. In a client-server architecture, the server has to distribute the file to all the clients, which puts a large load on the server and consumes a large bandwidth. However, with a P2P file distribution system, one client can redistribute a complete file or a part of a file to another client after receiving it from the server, thus reducing the load on the server.

One of the important protocols used for file distribution is BitTorrent, developed by Bram Cohen, an American Computer Programmer in 2001. Today, there are many BitTorrent clients confirming to BitTorrent protocol. Before we discuss the BitTorrent protocol, let us compare P2P and Client-server architectures in the context of scalability and understand the benefits of P2P architectures.

4.7.1.1 Scalability of P2P Architectures

In order to understand the benefits of P2P architecture, let us consider a file distribution application for a fixed set of peers and then compare both architectures by a common parameter: the file distribution time.

Fig. 4.8 shows a simple quantitative model where the server and the peers are connected to the Internet with access links. In the figure, u_s is the server's access link upload rate, u_1 to u_N is the client's access links upload rate, and d_1 to d_N is the access links download rate.

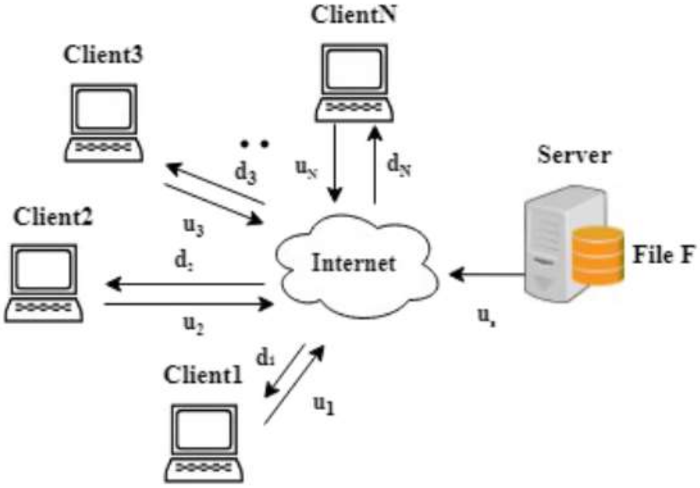


Fig. 4.8: An Illustration of File Distribution Problem

Let F be the size of a file (in bits) to be copied and N be the number of clients. The distribution time is defined as the time required to get a copy of the file by all the N clients. For simplifying the calculation of the distribution time, it is assumed that the bandwidth at the core of the network is very high but not at the access networks. Also, all the clients and the server are using only the file distribution application and are not busy with any other application when the distributed time is being calculated. So the upload and download access bandwidth is fully used by the file distribution application.

For the client-server application, two observations can be done:

1. With the server upload rate of u_s and file size of F, the file distribution time will be the time required by the server to copy the file to every client and is given by NF/ u_s
2. The minimum distribution time can be calculated as F/d_{min}, where d_{min} is the file download time by the peer with the lowest download rate.

With the above observations, it can be concluded that the lower bound on the minimum distribution time for the client-server architecture can be given as,

$$D_{cs} \geq max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\} \quad \text{--- (1)}$$

From equation (1), we can see that as the number of peers increases, the distribution time also increases linearly.

For the P2P application, three observations can be done:

1. The server has to send all the bits of the file at least once to a peer and later it can be redistributed by that peer to others. So the minimum distribution time is at least F/u_s.
2. Similar to the Client-server architecture, the minimum distribution time is at least F/d_{min} , where d_{min} is the file download time by the peer with the lowest download rate.
3. The total upload capacity of the system is equal to $u_{total} = u_s + (u_1 + u_2 + u_3 + \ldots. + u_n)$ and the system must upload at least F bits to each of the N peers, a total of NF bits, which is never greater than u_{total}. So, the minimum distribution time will be at least = NF/ u_{total}.

With the above observations, it can be concluded that the lower bound on the minimum distribution time for the P2P architecture can be given as,

$$D_{P2P} \geq max\left(\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{NF}{\left(u_s + \sum_{i=1}^{N} u_i\right)}\right) -- (2)$$

As each peer can redistribute the file to the other peers, the redistribution time does not increase linearly with the increase in the number of peers as shown in equation (2).
Fig. 4.9 shows the distribution time for both the architectures.

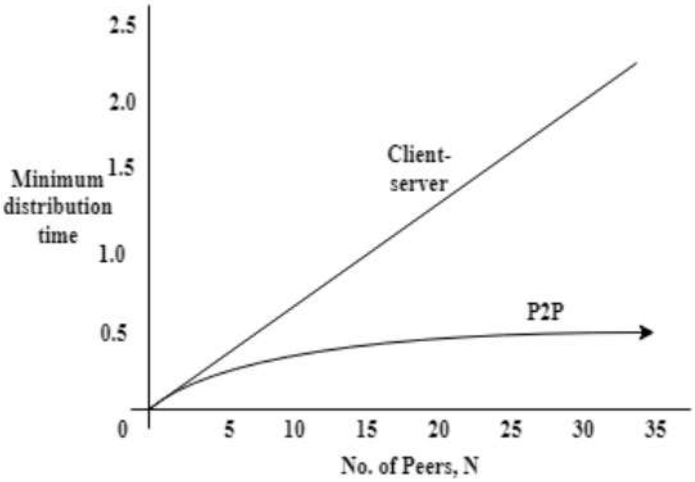


Fig. 4.9: Distribution time for P2P and client-server architectures

4.7.1.2 BitTorrent

This is a P2P protocol used for file distribution. A set of peers using this protocol for a specific file distribution is known as **Torrent.** Initially, when a peer joins a Torrent, it has no chunks of a file. It downloads and accumulates the file chunks over time and then redistributes them to the other peers. The peers in a torrent download equal sized chunks of a file from one another. Also, the peers upload equal sized chunks to other peers. A peer may leave the torrent either after downloading the whole file or only some part of it and also can join the Torrent anytime later.

Operation of BitTorrent

A BitTorrent infrastructure consists of a set of peers and a node called a **tracker.** The number of peers may range from less than tens to thousands of them. When the peers first join the torrent, they register with the tracker and periodically inform it about their existence so that it keeps track of all the participating peers.

To understand the operation of BitTorrent let us consider an example where a participant named Mary joins a torrent and registers herself. The tracker then randomly selects a set of participants from all the peers and sends their IP addresses to Mary. She then tries to establish a connection with them. Initially, only a few connections are established and as time passes more connections are established. The connected participants are called neighbors.

Fig. 4.10 shows Mary connected to three neighbors but subsequently get connected to all the peers in the torrent. The peers in a torrent at any instance of time have only a subset of chunks of file with them, so Mary has three chunks of a file.

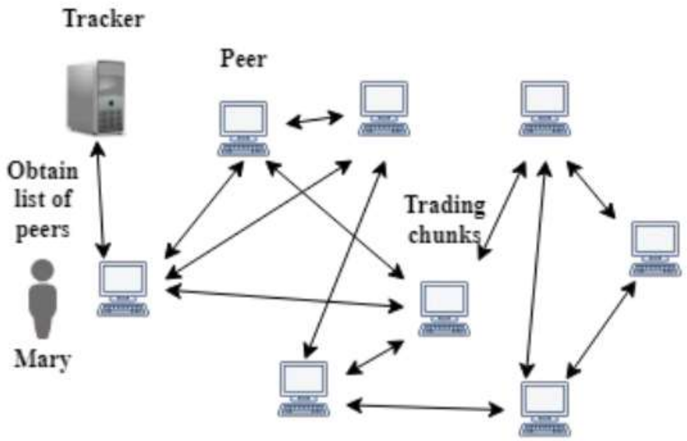


Fig. 4.10: File Distribution with BitTorrent

Periodically, Mary keeps asking the list of file chunks they have to her neighbors so that she can request the file chunks she needs. After collecting this information, Mary can decide upon two important processes: One, to which neighbor she can request the file chunks first, and two to which neighbor she can send the requested chunks.

To choose the neighbors for requesting file chunks, Mary uses a technique called "rarest first". With this technique, Mary chooses the file chunks that have fewer repeated copies of them among her neighbors. This allows the rarest copies of the file to be redistributed faster and make an equal number of copies of each file chunk in the torrent.

To choose the neighbors for sending file chunks, Mary uses another trading algorithm. In this method, she gives priority to a small set of neighbors who have been sending data to her at the highest rate and sends the file chunks. This rate is calculated every 10 seconds and modifies the set of neighbors. The peers who are in the set are called **unchoked** neighbors. Also, after every 30 seconds, Mary chooses one more neighbor named Rosy at random and sends a file chunk. In BitTorrent, this neighbor is called as **optimistically unchoked.** Mary now becomes a new neighbor and on the top priority to Rosy to whom she starts sending file chunks. But after 30 seconds if Mary finds a new neighbor, she will choose a new trading partner and initiates trading with that partner. So the peers with a better rate of data transfer find each other, and also due to a random selection new peers find an opportunity to receive and send chunks. But the other peers outside this set receive any chunks from Mary. This mechanism of trading is called tit-for-tat. There are other variants of BitTorrent proposed and many live streaming applications are inspired and implemented by this protocol.

4.8 Self-Assessment Questions

Q1. Briefly discuss how an E-mail system operates. (5 marks, L2)

Q2. Explain the working of SMTP with a neat diagram. (10 marks, L2)

Q3. List the Mail Access Protocols (2 marks, L1)

Q4. Briefly explain the working of IMAP protocol (8 marks, L2)

Q5. With the help of a neat diagram, explain the working of POP3 protocol (8 marks, L3)

Q6. Discuss the structure and working of DNS (10 marks, L3)

Q7. Briefly explain the difference between a client-server and peer-to-peer application (6 marks, L3)

Q8. Explain BitTorrent Protocol with an example (10 marks, L3)

4.9 Self-Assessment Activities

A1. Discuss SMTP AUTH, PIPELINING, and CHUNKING used in the context of SMTP.

A2. Can we optimize the SMTP process for effective delivery and avoiding spam when sending large number emails.

A3. Find out the difference between the process of deleting a message in IMAP and POP3 protocols.

A4. Discuss how the IMAP handles a multipart messages.

A5. How does BitTorrent prioritize the requests of a file chunk from a peer? Discuss if the method helps in efficient downloading of that chunk.

4.10 Multiple-Choice Questions

Q1. The application layer protocol for sending and receiving emails is, [1 mark, L1]

A. HTTP
B. FTP
C. SMTP
D. None of the above

Q2. The application layer protocol which helps in pushing the email messages to the server is, [1 mark, L1]

A. SMTP
B. IMAP
C. HTTP
D. POP2

Q3. The protocol used to transfer mails from the recipients mail server to the recipient user agent is, [1 mark, L1]

A. SMTP
B. POP3
C. FTP
D. None of the above

Q4. The port used for SMTP communication is, [1 mark, L1]

A. 25
B. 80
C. 586
D. None of the above

Q5. The user specifies the addresses in the ____ part of an email. [1 mark, L1]

A. Envelope
B. Body

C. Header

D. Attachment

Q6. In DNS, the resource record of type MX indicates, [1 mark, L1]

A. The message format

B. The message security

C. The specific mail server for a domain

D. None of the above

Q7. To stop the message transmission, POP3 uses ___ command, [1 mark, L1]

A. FINISH

B. QUIT

C. CLOSE

D. None of the above

Q8. MIME protocol is used to __ [1 mark, L1]

A. Attach multimedia to a mail message

B. Encrypt a message

C. Authenticate the user's identity

D. None of the above

Q9. MIME is used to encapsulate ____ type of content [1 mark, L1]

A. Only Text

B. Only Images

C. Only video files

D. Text, Images, Audio, Video, etc

Q10. The main function of BitTorrent is, [1 mark, L1]

A. Share files over a small network

B. Distribute large files over the Internet

C. Securing files over the Internet

D. None of the above

4.11 Keys to Multiple-Choice Questions

Q1. SMTP (C)

Q2. SMTP (A)

Q3. POP3 (B)

Q4. 25 (A)

Q5. Header (C)

Q6. The specific mail server for a domain (C)

Q7. QUIT (B)

Q8. Attach multimedia to a mail message (A)

Q9. Text, Images, Audio, Video, etc (D)

Q10. Distribute large files over the Internet (B)

4.12 Summary of the Unit

This unit covers one of the important applications used on the Internet which is E-mail. The discussion is divided into four important topics. The first topic covers the working of an E-mail system, which helps in the safe transmission of messages from the sender to the receiver. This system is implemented through various software components called agents deployed at the sender side, the receiver side, and the servers used by them. The software components help in the composition of the mail, sending and receiving process, saving, deleting, and organizing the mail messages and to control and manage the agents, protocols are used.

The second topic covers the primary protocol: SMTP used to push mails from the sender's system to the receiver's mail server. This topic also covers two more protocols: IMAP and POP3 used to pull the mail from the receiver's mail server to the receiver's system. It also covers another important protocol which is used to extend the functionality of SMTP to add multimedia components such as image, audio and video files to the textual content of an email message.

The third topic discusses about another application running at the application layer which works along with FTP, HTTP and SMTP protocols which is the DNS. This is also called as the Internet's directory service, whose basic functionality is to convert the domain names or website names or hostname provided by the users to IP addresses for those hosts for routing the information on the Internet. It acts like a phone book infrastructure on the Internet.

The fourth topic gives a brief introduction to Peer-to-Peer applications on the Internet. It covers one P2P application, that of a Large File Distribution among a set of peers and a protocol called BitTorrent which is used for this application.

4.13 Recommended Learning Resources

[1] James F Kurose and Keith W Ross, Computer Networking, A Top-Down Approach, Sixth Edition, Pearson,2017.

[2] Behrouz A Forouzan, Data and Communications and Networking, Fifth Edition, McGraw Hill, Indian Edition

[3] https://www.geeksforgeeks.org/introduction-to-electronic-mail/

[4] https://www.cloudflare.com/learning/email-security/what-is-email/

[5] https://www.techtarget.com/whatis/definition/e-mail-electronic-mail-or-email

[6] https://mailtrap.io/blog/mail-transfer-agent/

[7] https://www.oreilly.com/library/view/managing-imap/059600012X/ch01.html#:~:text=The%20Agents%20(MUA%2C%20MTA%2C,Mail%20User%20Agent%20(MUA).

[8] https://en.wikipedia.org/wiki/Email_agent_%28infrastructure%29

[9] https://www.geeksforgeeks.org/multipurpose-internet-mail-extension-mime-protocol/

[10] https://en.wikipedia.org/wiki/MIME