World Wide Web

The Web was first conceived in 1989 by **Sir Tim Berners-Lee** at the European Particle Physics Laboratory (CERN) in Geneva. His original idea was to develop a database of information on physics research, but he found it difficult to fit the information into a traditional database. Instead, he decided to use a hypertext network of information. With hypertext, any document can contain a link to any other document.

WWW or the Web is a repository of information spread all over the world and linked together. WWW has unique combination of flexibility, portability and user-friendly features.

WWW is a distributed client-server service in which client using a **browser** can access a service using **server**. The service provided is distributed over many locations called **websites**.

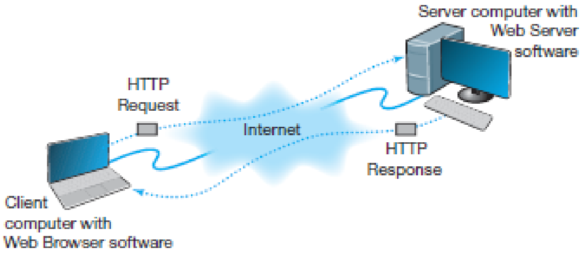


FIGURE 2.8 How the Web works

Working of WWW

The Web is a good example of two-tier client-server architecture (Figure 2.8). Each client computer needs an application layer software package called a **Web browser.**

There are many different browsers, such as Microsoft's Internet Explorer. Each server on the network that will act as a Web server needs an application layer software package called a **Web server.** There are many different Web servers, such as those produced by Microsoft and Apache.

To get a page from the Web, the user must type the **Uniform Resource Locator (URL)** for the page he or she wants (e.g., www.yahoo.com) or click on a link that provides the URL. **The URL** specifies the Internet address of the Web server and the directory and name of the specific page wanted. If no directory and page are specified, the Web server will provide whatever page has been defined as the site's home page.

For the requests from the Web browser to be understood by the Web server, they must use some standard **protocol** or language The standard protocol for communication between a Web browser and a Web server is **Hypertext Transfer Protocol (HTTP).**1 To get a page from a Web server, the Web browser issues a special packet called an **HTTP request** that contains the URL and other information about the Web page requested (see Figure 2.8). Once the server receives the HTTP request, it processes it and sends back an **HTTP response,** which will be the requested page or an error message.

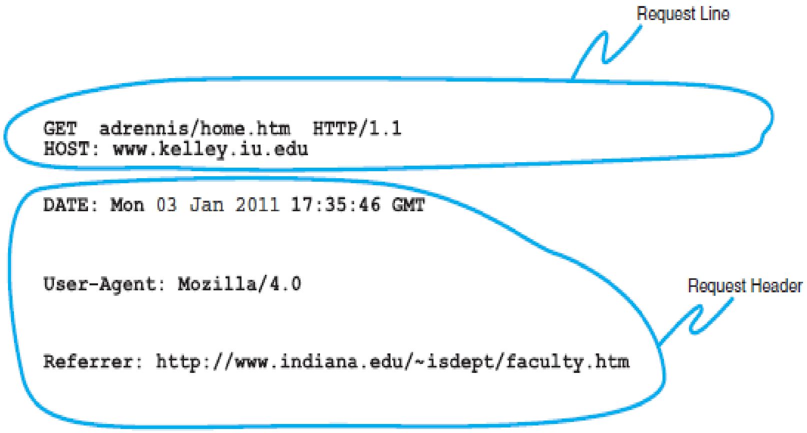
HTTP Request and Response

The HTTP response and HTTP request are simple text files that take the information provided by the application (e.g., the URL to get) and format it in a structured way so that the receiver of the message can clearly understand it. An HTTP request from a Web browser to a Web server has three parts. The first parts are required; the last is optional. The parts are:

i. The **request line,** which starts with a command (e.g., get), provides the Web page and ends with the HTTP version number that the browser understands; the version number ensures that the Web server does not attempt to use a more advanced or newer version of the HTTP standard that the browser does not understand.

ii. The **request header,** which contains a variety of optional information such as the Web browser being used (e.g., Internet Explorer) and the date.

iii. The **request body,** which contains information sent to the server, such as information that the user has typed into a form.



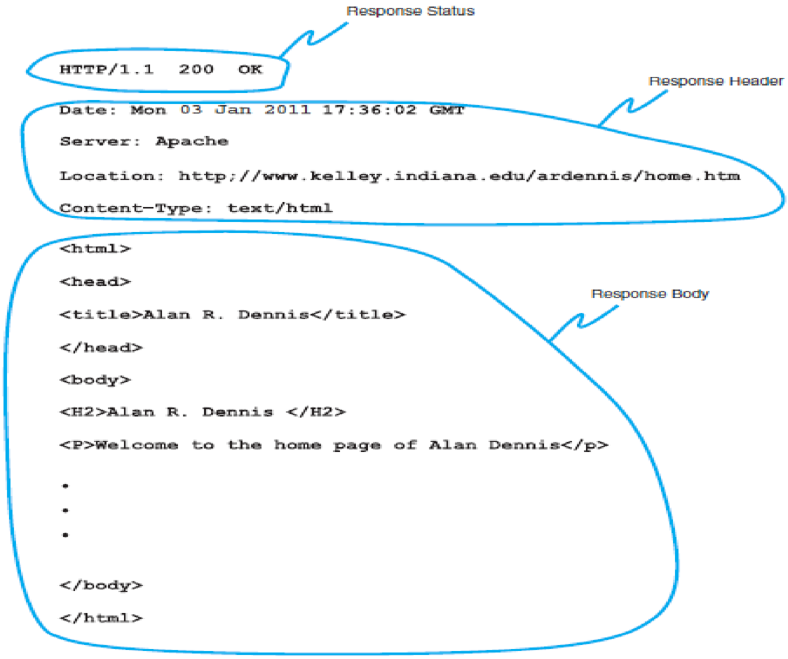
```
GET  adrennis/home.htm  HTTP/1.1
HOST: www.kelley.iu.edu

DATE: Mon 03 Jan 2011 17:35:46 GMT

User-Agent: Mozilla/4.0

Referrer: http://www.indiana.edu/~isdept/faculty.htm
```

FIGURE 2.9 An example of a request from a Web browser to a Web server using the HTTP (Hypertext Transfer Protocol) standard

Figure 2.9 shows an example of an HTTP request for a page on our Web server, formatted using version 1.1 of the HTTP standard. This request has only the request line and the request header, because no request body is needed for this request. This request includes the date and time of the request. The “Referrer” field means that the user obtained the URL for this Web page by clicking on a link on another page, which in this case is a list of faculty at Indiana University

The format of an HTTP response from the server to the browser is very similar to the HTTP request. It, too, has three parts, with the first required and the last two optional:

i. The **response status,** which contains the HTTP version number the server has used, a status code (e.g., *200* means “okay”; *404* means “not found”), and a reason phrase (a text description of the status code).
ii. The **response header,** which contains a variety of optional information, such as the Web server being used (e.g., Apache), the date, and the exact URL of the page in the response.
iii. The **response body,** which is the Web page itself.



```
HTTP/1.1  200  OK
Date: Mon 03 Jan 2011 17:36:02 GMT
Server: Apache
Location: http://www.kelley.indiana.edu/ardennis/home.htm
Content-Type: text/html
<html>
<head>
<title>Alan R. Dennis</title>
</head>
<body>
<H2>Alan R. Dennis </H2>
<P>Welcome to the home page of Alan Dennis</p>
.
.
.

</body>
</html>
```

FIGURE 2.10 An example of a response from a Web server to a Web browser using the HTTP standard

Figure 2.10 shows an example of a response from our Web server to the request in Figure 2.9. This example has all **three parts**. The response status reports “OK,” which means the requested URL was found and is included in the response body. The response header provides the date, the type of Web server software used, the actual URL included in the response body, and the type of file. In most cases, the actual URL and the requested URL is the same, but not always. The response body in this example shows a Web page in **Hypertext Markup Language (HTML).** The response body can be in any format, such as text, Microsoft Word, Adobe PDF, or a host of other formats, but the most commonly used format is HTML

Methods in Request line of request message

Method	**Action**
GET	Request a document from the server
HEAD	Requests information about a document but not the document itself
POST	Sends some information from the client to the server
PUT	Sends a document from server to the client
TRACE	Echoes the incoming request
CONNECT	Reserved
DELETE	Remove the web page
OPTIONS	Enquires about available options

Status codes and status phrase in status line of response message

Status Code	**Status Phrase**	**Description**
Informational		
100	Continue	The initial part of the request received, continue.
101	Switching	The server is complying to switch protocols
Success		
200	OK	The request is successful.
201	Created	A new URL is created.
202	Accepted	The request is accepted, but it is not immediately acted upon
204	No Content	There is no content in the body.

Redirection		
301	Moved Permanently	The requested URL is no longer used by the server.
302	Moved Temporarily	The requested URL has moved temporarily.
304	Not Modified	The document has not modified.
Client Error		
400	Bad Request	There is a syntax error in the request.
401	Unauthorized	The request lacks proper authorization.
403	Forbidden	Service is denied.
404	Not found	The document is not found.
405	Method not allowed	The method is not supported in the URL.
406	Not acceptable	The format requested is not acceptable.
Server Error		
500	Internal server error	There is an error, such as crash, at the server site.
501	Not implemented	The action request cannot be performed.
503	Service unavailable	The service is temporarily unavailable.

HTTPS

HTTP is fundamentally an insecure protocol. To address the need for secure web networking, alternatives are available such as HTTPS. HTTPS refer to the combination of HTTP and Secure Socket Layer (SSL) to implement secure communication between a Web browser and a Web server.

HTTPS capability is built into all modern Web browsers. The major difference seen by a user of a web browser is that URL addresses begin with **https://** rather that **http://. HTTP** connection uses port 80 and **HTTPS** uses port 443.

When HTTPS is used, the following elements of the communications are encrypted.

1. URL of the requested document
2. Contents of the document.
3. Contents of browser forms (filled in by browser user)
4. Cookies sent from browser to server and from server to browser.
5. Contents of HTTP header.

Electronic Mail

Electronic mail (or **email**) was one of the earliest applications on the Internet and is still the most heavily used application today. With email, users create and send messages to one user, several users, or all users on a **distribution list.** Several standards have been developed to ensure compatibility between different email software packages. Any software package that conforms to a certain standard can send messages that are formatted using its rules. The most commonly used standard is SMTP (Simple Mail Transfer Protocol).

Working of Email

The **Simple Mail Transfer Protocol (SMTP)** is the most commonly used email standard simply because it is the email standard used on the Internet. Email works similarly to how the Web works, but it is a bit more complex. SMTP email is usually implemented as a two-tier thick client-server application, but not always.

- **Two-Tier Email Architecture** With two-tier thick client-server architecture, each client computer runs an application layer software package called a **mail user agent,** which is usually more commonly called an email client. There are many common email client software packages such as Eudora and Outlook, Thunderbird, Amazon WorkMail, Spike. The user creates the email message using one of these email clients, which formats the message into an SMTP packet that includes information such as the sender's address and the destination address. The user agent then sends the SMTP packet to a mail server that runs a special application layer software package called a **mail transfer agent,** which is more commonly called mail server software (see Figure 2.11).

This email server reads the SMTP packet to find the destination address and then sends the packet on its way through the network—often over the Internet—from mail server to mail server, until it reaches the mail server specified in the destination address (see Figure 2.11). The mail transfer agent on the destination server then stores the message in the receiver's mailbox on that server. The message sits in the mailbox assigned to the user who is to receive the message until he or she checks for new mail.

The SMTP standard covers message transmission between mail servers (i.e., mail server to mail server) and between the originating email client and its mail server. A different standard is used to communicate between the receiver's email client and his or her mail server. **Two commonly used standards for communication between email client and mail server are Post Office Protocol (POP) and Internet Message Access Protocol (IMAP).** IMAP, the most noticeable difference is that before a user can read a mail message with a POP (version 3) email client, the email message must be copied to the client computer's hard disk and deleted from the mail server. With IMAP, email messages can remain stored on the mail server after they are read. IMAP therefore offers considerable benefits to users, who read their email from many different computers (e.g., home, office, computer labs).

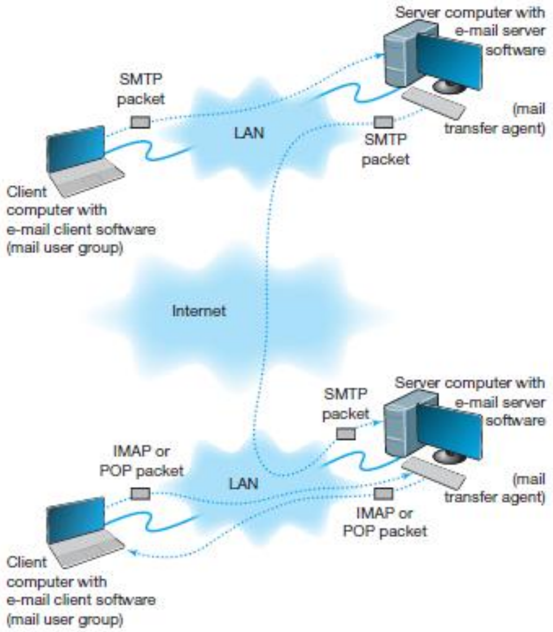
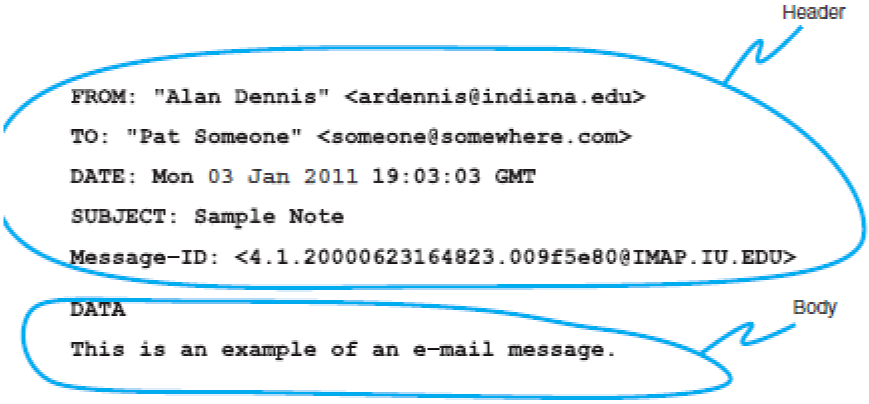


FIGURE 2.11 How SMTP (Simple Mail Transfer Protocol) email works. IMAP = Internet Message Access Protocol; LAN = local area network

SMTP Packet

SMTP defines how message transfer agents operate and how they format messages sent to other message transfer agents. An SMTP packet has two parts:

i. The **header,** which lists source and destination email addresses (possibly in text form [e.g., “Pat Smith”]) as well as the address itself (e.g., psmith@ somewhere.com), date, subject, and so on.
ii. The **body,** which is the word *DATA,* followed by the message itself.



```
FROM: "Alan Dennis" <ardennis@indiana.edu>
TO: "Pat Someone" <someone@somewhere.com>
DATE: Mon 03 Jan 2011 19:03:03 GMT
SUBJECT: Sample Note
Message-ID: <4.1.20000623164823.009f5e80@IMAP.IU.EDU>
DATA
This is an example of an e-mail message.
```

Header
Body

FIGURE 2.13 An example of an email message using the SMTP (Simple Mail Transfer Protocol) standard

Figure 2.13 shows a simple email message formatted using SMTP. The header of an SMTP message has a series of fields that provide specific information, such as the sender’s email address, the receiver’s address, date, and so on. The information in quotes on the *from* and *to* lines is ignored by SMTP; only the information in the angle brackets is used in email addresses. The *message ID* field is used to provide a unique identification code so that the message can be tracked. The message body contains the actual text of the message itself.

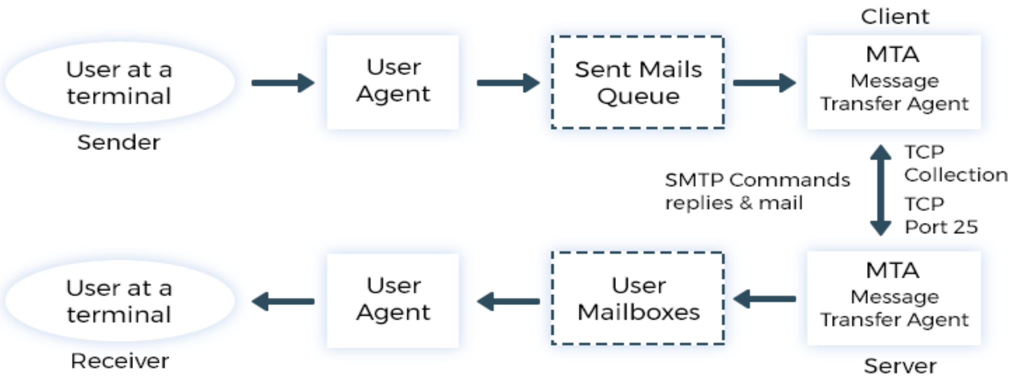


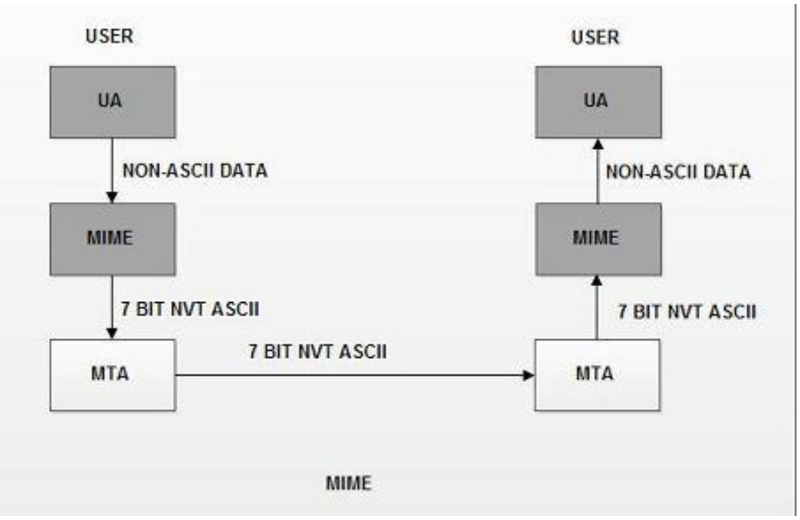
Fig: SMTP System

Multipurpose Internet Mail Extension

Electronic mail has simple structure it can send message only in (Network Virtual Terminal) NVT 7-bit ASCII format. SMTP is a simple standard that permits only the transfer of text messages. It was developed in the early days of computing, when no one had even thought about using email to transfer non text files such as graphics or word processing documents.

Several standards for non-text files have been developed that can operate together with SMTP, such as **Multipurpose Internet Mail Extension (MIME)**.

MIME is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA (Message Transfer Agent) to be sent through the Internet. The message at the receiving site is transformed back to the original data. I.e. MIME is a set of software functions that transforms non ASCII data to ASCII data and vice versa



Other Applications

There are literally thousands of applications that run on the Internet and on other networks. Most application software that we develop today, whether for sale or for private internal use, runs on a network. In this section, we will briefly discuss only three commonly used applications: **Telnet, Instant Messaging (IM), and Video conferencing**.

Telnet: Telnet is a protocol that enables users to log in to servers (or other clients). It requires an application layer program on the client computer and an application layer program on the server or host computer. Once Telnet makes the connection from the client to the remote host, the client becomes virtual terminal allowing to communicate with the remote host. In most of the cases while using telnet protocol there must be the use of account name and password of an authorized user to log in. occasionally user can log in as guest or public without having an account.

Although Telnet was developed in the very early days of the Internet (actually, the very first application that tested the connectivity on ARPANET was Telnet), it is still widely used today. Because it was developed so long ago, Telnet assumes a host-based architecture.

Even Telnet requires a logging name and password; it is vulnerable to hacking because it sends all data including password in plaintext a hacker can easily obtain logging name and password. Because of this security issue, the use of Telnet has been replaced by another protocol called Secure Shell (SSH). However Telnet is often used by Network Administrator for diagnostic & debugging purposes.

Instant Messaging: One of the fastest growing Internet applications has been **instant messaging (IM).** With IM, we can exchange real-time typed messages or chat with your friends. Some IM software also enables us to verbally talk with friends in the same way we use the telephone or to use cameras to exchange real-time video in the same way we might use a videoconferencing system. Several types of IM currently exist, including Google Talk, Instant Messenger, MS Teams etc.

Instant messaging works in much the same way as the Web. The client computer needs an IM client software package, which communicates with an IM server software package that runs on a server. When the user connects to the Internet, the IM client software package sends an IM request packet to the IM server informing it that the user is now online. The IM client software package continues to communicate with the IM server to monitor what other users have connected to the IM server. When one of your friends connects to the IM server, the IM server sends an IM packet to your client computer so that you now know that your friend is connected to the Internet. The server also sends a packet to your friend’s client computer so that he or she knows that you are on the Internet.

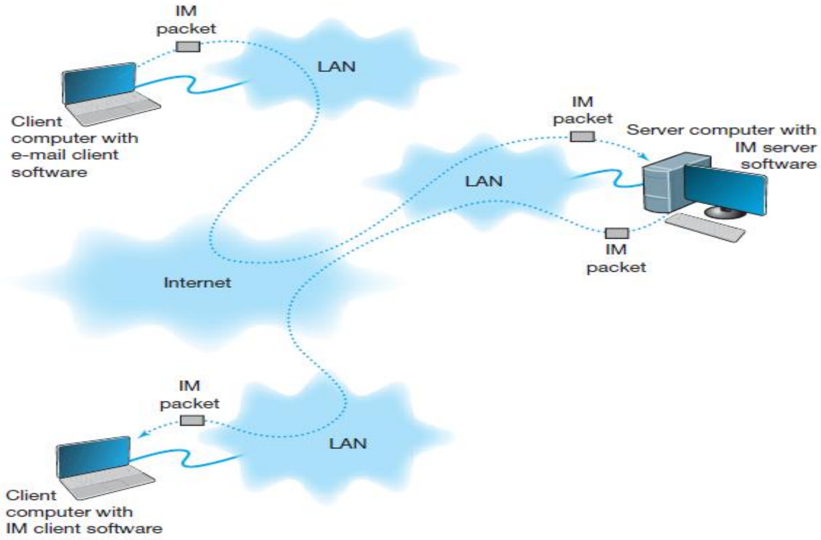


Fig: How Instant Messaging Works

Videoconferencing

Videoconferencing provides real-time transmission of video and audio signals between two or more users regardless of their location. Initially video conferencing software only enabled users to make video calls or hold group video conferences. However now, with advancement in technology, video conferencing software acquires plenty of helpful tools and features for remote communication and learning. Today video conferencing is a tool for integrated video collaboration and unified communications platforms that offer screen sharing, slideshow; recording, instant messaging, project management tools, telephony integration and many others. The fastest growing form of videoconferencing is **desktop videoconferencing.** Small cameras installed on top of each computer permit meetings to take place from individual offices. Special application software such as Zoom, Microsoft Team, webex, Skype, is installed on the client computer and transmits the images across a network to application software on a videoconferencing server. The server then sends the signals to the other client computers that want to participate in the videoconference.

The transmission of video requires a lot of network capacity. Most videoconferencing uses data compression to reduce the amount of data transmitted. Surprisingly, the most common complaint is not the quality of the video image but the quality of the voice transmissions. Special care needs to be taken in the design and placement of microphones and speakers to ensure quality sound and minimal feedback.

The common use standards for video conferencing are:

i. **H.320**: it is designed for room-to-room videoconferencing over high-speed telephone lines.
ii. **H.323**: It is a family of standards designed for desktop videoconferencing and just simple audio conferencing over the Internet.
iii. **MPEG-:** It is designed for faster connections, such as a LAN or specially designed, privately operated WAN.

Webcasting is a special type of one-directional videoconferencing in which content is sent from the server to the user. The developer creates content that is downloaded as needed by the users.