MT834
Unit I

Components of the Internet
Course team
Developer: Nancy J Yeager, National Center for Supercomputing Applications, USA
Designer: Chris Baker, OUHK
Coordinator: Dr Reggie C Kwan, OUHK
Member: Dr Andrew Lui, OUHK

External Course Assessor
Professor Mingshu Li, Institute of Software at Chinese Academy of Sciences

Production
ETPU Publishing Team

Acknowledgement
The OUHK gratefully acknowledges a grant of $100,000,000 from the Hong Kong Government designated for course development.

MT834 Web Server Technology was developed in part using resources provided by this generous grant.

All rights reserved.
No part of this material may be reproduced in any form by any means without permission in writing from the President, The Open University of Hong Kong.

The Open University of Hong Kong
30 Good Shepherd Street
Ho Man Tin, Kowloon
Hong Kong
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td><strong>What is the Web?</strong></td>
<td>4</td>
</tr>
<tr>
<td>HTML</td>
<td>4</td>
</tr>
<tr>
<td>URLs</td>
<td>6</td>
</tr>
<tr>
<td><strong>The Internet</strong></td>
<td>13</td>
</tr>
<tr>
<td>The design and structure of the Internet</td>
<td>14</td>
</tr>
<tr>
<td><strong>The Web server</strong></td>
<td>19</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>22</td>
</tr>
<tr>
<td><strong>Solutions to self-tests</strong></td>
<td>23</td>
</tr>
</tbody>
</table>
Overview

Welcome to Unit 1 of MT834 Web Server Technology.

Please be reminded that you should have read the MT834 Course Guide by now. It’s also a good idea to browse the course website through the Open Learning Environment (OLE). The course website offers interesting information and activities associated with each unit. You will find this course provides theoretical concepts as well as hands-on experimentation. As a graduate student, you are also encouraged to visit the University’s award winning Electronic Library. If you have done all these things, you are ready to get started with this first unit.

I hope you are as excited as I am by the new era of the information age. Without a doubt, the Web — together with the Internet — plays a most important role in allowing information to be accessed easily and instantly. To understand Web servers, you need to have a good understanding of Internet topology as well as basic networking skills. (If any of the concepts covered in this unit need further clarification, please use the Electronic Library to find more information.)

This unit examines the components of the Internet:

- the physical network;
- Transmission Control Protocol/Internet Protocol (TCP/IP) network protocol;
- Domain Name Service (DNS);
- telnet;
- ftp and HyperText Transfer Protocol (HTTP) protocols;
- Web client and server software; and
- HyperText Markup Language (HTML) documents.

To learn about these concepts, you will read the Web Server Technology text and conduct a series of experiments on your local computer. You will install a Web browser, create your own HTML document, and then install and configure an Apache Web Server to serve your HTML document.

This first unit of MT834 Web Server Technology is expected to take you four weeks (or about 30 hours of study time) to complete. Please plan your time carefully. As you work through the unit, you will need to refer to online readings and activities on the MT834 Web Server Technology course website. You will also have to refer to the course textbook:


Please now begin Unit 1 by reading the unit’s learning objectives.
Objectives

By the end of Unit 1, you should be able to:

1. Identify the function and position of each Internet component in the network hierarchy.

2. Install and configure a Web client and Web server software on your own computer.

3. Create a simple HTML document. Install and serve the HTML document via your Apache Web server.

4. Use and understand the structure of a network name and how the Domain Name Service maps network names to network addresses.

5. Understand the basic functioning and characteristics of a TCP/IP as a network protocol.
Introduction

You have all probably experienced the point-and-click phenomenon that links us together through a web of information: the World Wide Web. The Web is a technology revolution which promises to change the way we communicate and conduct business. But what exactly is the Web made up of? Is the Web the same as the Internet? Is the Web a boundless ‘sea’ of information that we ‘surf’? Is the Web the software on our personal computer that brings the information world to us? Or is the Web a physical network similar to the telephone system? Perhaps the Web is all of these things. In this unit we answer the question ‘What is the Web?’
What is the Web?

The Web is the most popular Internet application. This means the Web uses the Internet, but it is not the Internet. The Web was designed to work on top of the Internet and cannot function without it. The Internet, as we will read in the next section, is a set of physical networks (hardware) and networking software that deliver data from one point on the network to another point on the network. The Internet began in the 1960s and grew over time, while the Web was not developed until the 1990s.

The next reading will describe the components of the Web and the new technologies that were created to build the Web.

**Reading**

Chapter 1, Section 1.2, pages 3–5 in *Web Server Technology*.

In this reading we learned that the Web is one of many Internet services. But what makes the Web ‘a Web’, and what distinguishes it from other Internet services? The answer is found in the three technologies that were developed to create the Web:

1. the Web’s document language: the HyperText Markup Language (HTML);
2. the Web’s system for addressing and locating documents: Universal Resource Locators (URLs);
3. the communication language between the Web client and Web server: the HyperText Transfer Protocol (HTTP).

HTML and URLs will be explained here. HTTP will be explained in-depth in the next unit, *Unit 2*.

**HTML**

HTML is a markup language, a system for expressing text and formatting directives all in one document. A formatting directive is an instruction to the display software (in this case a Web browser) describing how the enclosed text should be displayed. Font size, paragraph breaks, tabs, and instructions to place a text segment into a table are all markup language formatting directives that can be found in HTML documents. An HTML formatting directive is expressed as an HTML tag. HTML tags begin with this specially designated bracket character: ‘<’, end with this specially designated bracket character: ‘>’, and contain a sequence of characters enclosed between the bracket characters which indicate the type of the HTML tag.
An HTML formatting directive to make the word ‘important’ appear in a bold font style would look like this in an HTML document:

This word is <b>important</b>.

When displayed by a Web browser the same text would look like this:

This word is <b>important</b>.

<b> is the HTML tag that instructs the Web browser display software to begin displaying the text in a bold font style. The </b> HTML tag instructs the Web browser display software to stop displaying the text in a bold font style.

A formatting directive to create a new paragraph in a text segment would look like this in an HTML document: This is a sentence

<p>Here’s another sentence.</p>

When displayed by a Web browser the text would look like this:

This is a sentence.

Here’s another sentence.

The <p> HTML tag instructs the Web browser display software to begin a new paragraph.

The easiest way to learn HTML formatting directives is through exercises such as these where you view the HTML tags in the source of a HTML document and then view how the Web browser interprets and displays the tags. We will trace through a few more of these examples later in the unit.

Each document creation system, such as Microsoft Word, Excel, or Lotus Notes, has its own conventions for marking up a document with formatting directives. HTML distinguishes itself from most of these other document systems in that it is also a hypertext system. Hypertext systems link documents and other resources together. Just as there are HTML tags for representing formatting directives, there are HTML tags called anchor tags for representing HTML links or ‘anchors’ to other Internet resources. When you point-and-click with your mouse in an HTML document and a new document or other multimedia resource pops up, you are using HTML anchors and HTML’s hypertext capabilities.

The text ‘Open University of Hong Kong’ with a hypertext anchor (link) to the OUHK’s home page would look like this in an HTML page:

<A HREF="http://www.ouhk.edu.hk"> The Open University of Hong Kong</A>

When displayed by your Web browser, the anchor is highlighted and underlined. It would look like this:

The Open University of Hong Kong
URLs

HTML hypertext anchors contain Universal Resource Locators (URLs) which tell the Web browser where the resource is located on the Internet. A URL is an address of an Internet resource. URLs express a resource’s unique location, just as street addresses express a unique location of a place in our physical world. From your travels on the Web you are probably familiar with the form of a URL as the protocol name, ‘http’, followed by ‘://’, followed by a website name, ‘www.ouhk.edu.edu’. In the above anchor you should recognize the form of the URL for the Open University of Hong Kong home page ‘<http://www.ouhk.edu.hk>’.

The next reading describes how anchors and URLs make the Web ‘a web of information’. The important points to focus on the reading are:

- the description of the multimedia nature of Web documents and their digital format or digital encoding; and
- the concepts of MIME and hypertext.

**Reading**

Chapter 1, Section 1.3, pages 5–11 in *Web Server Technology*.

We have learned that a hypertext document contains highlighted areas which, when selected, automatically transfer the client to a different location, the URL, which may be another HTML document or a multimedia document such as an image, movie, or audio sound clip. The Web is a multimedia hypertext system. The activity below contains a series of experiments that will demonstrate how HTML and hypertext links work.
Activity 1.1

Usually, you will have to refer to the course website to complete activities in MT834 Web server technology. The activities in this course require you to access sites, to look at HTML files online, and to download software. These activities therefore need to be done online via the course website.

However, for these first few activities, I have also included printed notes to help you as you install some browsers and get you used to working online.

For this activity you will want to have two Web browsers installed on your home computer so that you can contrast how each Web browser handles common browser functions. If you have not already done so, you will need to download and set-up two Web browsers from:

- Microsoft Internet Explorer Web browser: <http://www.microsoft.com/downloads>

The download may take minutes to hours depending on your modem speed and whether you have broadband service. You can see that after half an hour, there is still over an hour left using my office computer. On the other hand, it took only minutes at home to download the 22 MB file.

Follow the instructions for downloading and installing a new version of each Web browser for your operating system. If you have a reasonably current version of one of these Web browsers you do not need to download and reinstall your existing Web browser software. In this exercise we will see that every Web browser is required to perform a set of standard tasks such as displaying HTML text. Web browsers may name their functions slightly differently or place their function buttons in a slightly different location on the graphical user interface but every Web browser must perform some standard functions. Let’s compare and contrast how Navigator and IExplorer handle these common functions.
1. View or alter the configuration settings of the Web browser
   - For IExplorer: View — Internet Options
   - For Netscape: Edit — Preferences

You will see common configuration settings for the browser such as font size and language preference. We will be altering some of these configuration settings in future units. We can simply choose the font we like:
2 **Viewing the properties of an image.** Point each Web browser at the URL for the Open University of Hong Kong’s home page: ‘<http://www.ouhk.edu.hk>’. Position your mouse over an image on the Web page.

- For IExplorer: Right click on mouse — Properties
- For Netscape: Right click on mouse — View Info

Both browsers display the media type of the image (e.g., a GIF or JPEG image and the URL for that image). Netscape Navigator shows this information for all images on the Web page and also provides an extensive description of other multimedia characteristics such as image dimension, size, and colour.

Let’s say you open the OUHK home page using IExplorer and you want to know more about the picture taken from across Good Shepherd Street. If you right click on the picture and then select properties, you will see:

![Properties](image.png)

You see the file name, file type, size, and so on.

3 **Viewing the source of the HTML document**

Point each Web browser at the URL for the Open University of Hong Kong’s home page: ‘<http://www.ouhk.edu.hk>’

- For IExplorer: View — Source
- For Netscape: View — Page Source

You should be able to recognize some HTML tags as well as HTML hypertext anchor tags. You will also see a variety of HTML tags indicating some interactive technologies such as Javascript or Java. Ignore these complex HTML tags and constructs for now; we
will cover these concepts in later units. The number and types of HTML tags is quite large. For now you should concentrate on learning the HTML basics shown in Figure 1, the HTML tags which should occur in every HTML document:

```html
<HTML>
<HEAD>
<TITLE> page title </TITLE>
</HEAD>
<BODY> body of the document </BODY>
</HTML>
```

**Figure 1.1** Tags occurring in every document

Can you find the HTML, HEAD, TITLE, and BODY tags in the HTML source of the Open University of Hong Kong’s home page, ‘<http://www.ouhk.edu.hk>’?

The tags labelled ‘HTML’ ‘/HTML’ denote the beginning and end of the HTML document. The BODY tag indicates that the enclosed text is the body of the HTML document, while the HEAD tag indicates the enclosed text contains descriptive information about the document. The title of the HTML document, displayed by the browser at the top of the document, is enclosed with the ‘TITLE’ and ‘/TITLE’ tags.

The easiest way to learn about HTML is by using a learn-by-example approach. As you browse documents on the Web you can learn more about HTML by:

- viewing the HTML document source;
- saving the HTML document source or part of the HTML document into a file with your text editor; and then
- seeing how the Web browser interprets and displays the HTML source code.

The following activity allows you to practice this learn-by-example approach to HTML.

**Activity 1.2**

1. Save the HTML document source (text and HTML tags) shown in Figure 1 of Activity 1.1 to a file called, for example, ‘mydoc.htm’ on your local computer.

2. View ‘mydoc.htm’ with your Web browser to see how the Web browser interprets the HTML formatting directives and displays the text.
• For Netscape: File — Open Page — Choose File to (select ‘mydoc.htm’)
• For IExplorer: File — Open — Browse to (select ‘mydoc.htm’)

Where is the HTML title displayed? Where is the HTML body displayed?

Using this same testing technique as you work through the next reading on HTML tags is helpful. This supplementary online reading gives you a brief tutorial on how to construct an HTML document containing HTML tags.

**Online reading 1.1**

Read through this document: <http://www.arachnoid.com/lutusp/html_tutor.html>. Try the last exercise of creating HTML documents to test out the HTML tags and witness how your Web browser displays these HTML tags.

When you have completed the reading you should be able to construct an HTML document containing HTML tags for these HTML formatting directives:

1. bold text
2. italicized text
3. underlined text
4. new paragraph
5. hypertext link
6. inline image
7. background graphic
8. font colour
9. background colour

Here are some further useful online references to consolidate your study of HTML. If you did not understand the concepts covered in the previous two readings from the Web Server Technology text you should consult these references. Reading these documents is not required. They are for your reference only.
Online reading 1.2

Please complete the following optional readings:

<http://www.hwg.org/resources/?cid=14>
<http://www.htmlhelp.org/reference/html40>
<http://www.htmlhelp.org/glossary/>
<http://www.htmlhelp.org/links/tutorials.htm/>

You should now understand the basics of HTML, URLs, hypertext, and the multimedia nature of Web documents. Test your knowledge by completing the following self-test. Feedback is provided at the back of the unit.

Self-test 1.1

1 What three new technologies were created to build the Web? What do each of these technologies do?

2 What is hypertext? What are some of the characteristics of hypertext?

3 What is a URL? Give an example of a URL.

4 How do you represent a hypertext link in HTML?

5 How do you represent an inline image in HTML?

6 What HTML tags should always be included in an HTML document?
The Internet

The Internet began as an effort to create a communication network that would survive a powerful wartime bomb. The idea behind the new network was that even if a section of it was destroyed, messages could still be delivered by redirecting them over sections that were still intact. Networks of the time were not designed to work this way; messages on the network had fixed routes. Every time a message travelled from point A to point B it travelled across the same route, through a centralized node E, for example. If node E was destroyed, messages could not be delivered from point A to point B. It was necessary, if node E was destroyed, for the new ‘blastproof’ network to be able to deliver messages from point A to point B by rerouting the message through another network node. To fulfil the requirements of a wartime network, the network needed a new design, a design that could handle failure and reroute traffic around trouble spots. Therefore, a new type of network was designed and built: one that we now call the Internet. The following online reading outlines the history of the Internet and its unique design.

Read the first two pages up to the paragraph that begins with ‘ARPANET itself formally expired in 1989’ to understand how the Internet developed to meet the requirements of a wartime network. You may read the rest of the article if you are interested, but it is not required. You do not have to remember historical facts and organizations in the article such as RAND, MIT, UCLA, ARAPANET and Pentagon.

Online reading 1.3

The history of the Internet
<http://www.forthnet.gr/forthnet/isoc/short.history.of.internet/>

From Reading 1.3 you should have learned that the Internet is a very large collection of cooperating computer networks. At the very lowest level these networks consist of a set of physical network hardware and low-level communication software. This physical network layer is the lowest layer where a network connection takes place: two physical nodes, computers on the network, ‘connect’ to exchange a communication message in some form. A network connection can have a variety of physical forms as shown in Figure 1.2.
Figure 1.2 Some physical connections to the Internet

All of the network connections shown in Figure 1.2 are a part of the Internet. Each network has its own low-level communication software (such as Ethernet, FDDI, X.25, IBM token ring, or ATM) so that the specialized network hardware can ‘talk’ to one another. The Internet network software operates over the top of these communication layers. The magic of the Internet is how these very different computer networks cooperate to form one ‘Inter-network’ — the Internet.

The design and structure of the Internet

The design features of the Internet and how the Internet works are important concepts to understand. The Internet is a decentralized resource — there are no top-level centralized authorities that can fail and stop the operation of the entire network.

The most important design aspect of the Internet is its ability to recover from failure in one section of the network. Messages are broken into discrete units called ‘packets’ which contain the source of the message (node A) and the requested destination of the message (node B). If the packet fails in traversing the network at node E, the packet is simply resent, again, and again, across another network path or ‘route’ until it reaches its requested destination, node B.

Communication protocols

It is relatively easy to add a new network to the Internet because the Internet’s communication protocols are built on standards that are open and publicly available. Diverse types of networks are unified to create the Internet through a higher level network communication protocol called TCP/IP as seen in Figure 1.3, page 14 in Web Server Technology. Owners of new physical network types only need a version of TCP/IP that works over their network to join the Internet. For example, the owner of a physical network composed of Ethernet network cards (hardware) and low-level Ethernet communication software can join the Internet by reading the TCP/IP communication standard, and writing a TCP/IP software module to work on top of his or her Ethernet network.
At the topmost level in Figure 1.3 (p. 14 in *Web Server Technology*) is a set of Internet services which are built on top of TCP/IP. The next reading introduces these Internet services and reviews in detail how the Internet works.

**Reading**

Chapter 1, Section 1.4, pages 11–14 in *Web Server Technology*.

After reading this section you should be able to describe the roles of TCP and IP in TCP/IP, how IP addresses work, Internet services such as DNS, FTP and telnet, Internet domains and domain names.

We have learned that the Internet Protocol, IP, defines a system for assigning unique addresses to resources on the network so that it can find other computers and Web services. One host computer usually has one IP address. TCP/IP must use the IP system of assigning unique addresses to Internet resources so that it can route packets across the Internet. Moreover, we have learned that the IP addressing system defines a hierarchy of networks and we know that a host computer’s IP address reflects its position in the network hierarchy. Let’s apply this concept to an example to see how IP addresses and routing work together. Consider this IP address:

202.40.157.163

Starting from left to right in interpreting this address, we move from a larger, less specific area of the network (network 202) to a more specific individual host on a smaller network (163).

Imagine this Internet address belongs to a host on the Open University of Hong Kong’s local area network. A simplistic way of interpreting this address is:

- 202 is a network that covers all of Asia;
- 40 is a network for the city of Hong Kong;
- 157 is the network containing all the computers for the Open University of Hong Kong;
- 163 is the individual host identifier of the computer on the Open University of Hong Kong’s network.

In terms of routing packets, the IP layer on the Asia network (202) only needs to know how to send packets to the city of Hong Kong network, 40. It does not need to know anything about network 157 or host 163. The city of Hong Kong network, 40, only needs to know how to route packets to the 157 network, and the 157 network only needs to know how to route packets to host number 163. The way that the networks are organized as a hierarchy limits the amount of knowledge that any one routing node must have about the entire system of networks.
There is a special IP address that we will be using in future exercises: 127.0.0.1

Network 127 is a specially designated network that is not ‘owned’ by any official organization. Individual computer hosts assume ownership of the 127 network address to manage their network resources. 127.0.0.1 is the ‘loopback address’, a special address that computer hosts use to direct TCP/IP traffic to back to themselves. The loopback address is useful for debugging and testing Internet services and we will use it in setting up our own Internet services.

Internet addresses — like 127.0.0.1 or 202.40.157.163 — are difficult to memorize and discuss, so a system of network names, also called domain names was developed. Each Internet address has one or more network names. For example, the loopback address, 127.0.0.1, is given the network name ‘localhost’. In the reading you learned that the Domain Name Service (DNS) was created to map Internet addresses to domain names. These domain names are the basis for the Web’s URL since a URL contains domain names. You would have heard the phrase ‘dot com’. ‘.com’ is a network name that refers to the large portion of the Internet that belongs to the commercial (com, for short) domain. Let’s conduct a few experiments to see how DNS maps Internet addresses to network names.

Activity 1.3

In this activity you will experiment with the nslookup command on your local computer. When nslookup is given a network name, it calls the Domain Name Service (DNS) which returns the IP address for that network name.

The experiment results, shown below, were conducted on a computer on a local area network configured with a single, statically assigned IP address and DNS service. ‘Statically assigned’ means the computer on the network always has the same IP address and always uses the same DNS server. These experiments may or may not work on your home computer depending on how the Internet address and DNS service is assigned by your Internet service provider (ISP). Some ISPs dynamically assign IP addresses to hosts. With a dynamically assigned IP address your IP address will be different for each Internet session, and it will be difficult to configure your computer to use the DNS nslookup command correctly. If this is the case, just read through and understand the results shown below.

Open an MS DOS Command.exe window and type:

    nslookup

If you see a response similar to this then your DNS is set up correctly and you can conduct these experiments on your home machine.

    Address: 207.46.130.45
If, however, you see an error response such as ‘DNS request timed out’, then your ISP manages your DNS and IP address, and your computer is likely to have a dynamically assigned IP address. (On Windows NT, you can consult your TCP/IP configuration at: My Computer — Control Panel — Network — Protocols — TCP/IP Properties). If this is the case, follow along with this exercise but do not try it on your local computer.

You can also try this through the Web: <http://www.kloth.net/services/nslookup.php>

You may also want to try this experiment later if you have access to a laboratory computer on a local area network.

1 The most common DNS resolution process is the case where one network name maps to one Internet address. For example:

```
slookup www.sierraclub.org
Name: www.sierraclub.org
Address: 207.90.163.3
```

2 A single network name can have multiple IP addresses. For example:

```
slookup microsoft.com
Name: microsoft.com
Address: 207.46.130.45, 207.46.230.218, 207.46.230.219, 207.46.230.229
```

A single network name has multiple IP address when the computer is multi-homed, i.e. has more than one network interface.

3 A single network IP address can have multiple network names as we see here:

```
slookup learn.ouhk.edu.hk
Name: du02.ouhk.edu.hk
Address: 202.40.157.162

nslookup study.ouhk.edu.hk
Name: study.ouhk.edu.hk
Address: 202.40.157.162
```

In this case there is one computer with one network interface which performs several Internet services, each with it’s own name, or ‘alias’.
In Activity 1.3, we witnessed that 'learn.ouhk.edu.hk' is a network name with a corresponding Internet address. In the previous section we witnessed that the network name 'www.ouhk.edu.hk' was used in the URL 'http://www.ouhk.edu.hk'. This serves to demonstrate that the Web uses the Internet's Domain Name address system in its URL address system. The Web depends on Internet services as its foundation; the Web cannot exist without the Internet!

At this point you should understand TCP/IP, IP addresses and domain names and have a basic understanding of Internet services. Answer these questions to test your knowledge, then check your answers with those at the end of the unit.

<table>
<thead>
<tr>
<th>Self-test 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  What is a protocol? Why are protocols important to the Internet?</td>
</tr>
<tr>
<td>2  Name three Internet services that use TCP/IP and describe what each service does.</td>
</tr>
<tr>
<td>3  What is the role of the IP layer in TCP/IP?</td>
</tr>
<tr>
<td>4  What is the role of the TCP layer in TCP/IP?</td>
</tr>
</tbody>
</table>
The Web server

We have investigated what makes the World Wide Web a ‘web’ and found that it is the hypertext links in HTML. But what makes the Web ‘world-wide’? As you sit at your home computer and access with ease resources that physically reside in a computer across the globe, you must conclude that it is the Internet and the Web browser and Web server software that makes the Web ‘world-wide’. The Web is a client-server architecture which means there are two pieces of software involved in the communication, and each piece has its own logic and role in the communication. The Web server software delivers information and the Web browser software receives the information and displays the result for you. In this section, we will read about Web client and Web server software and HTTP, the Web’s communication protocol.

Reading

Chapter 1, Section 1.5, page 14 to page 18 in Web Server Technology.

We have seen that the Internet is built on open, public, standards such as TCP/IP and that this openness permits a range of diverse networks to easily join the Internet. The Web follows the Internet tradition and is also built on open communication standards: the HyperText Transfer Protocol (HTTP). Any program that implements the HTTP protocol is a full-fledged participant in the World Wide Web. The standards and protocols are general enough so that Web browsers and Web servers are implemented on a wide variety of computers and in a wide variety of computing languages: C, C++, Java, Python, etc.

We will read much more about the internal workings of the HTTP protocol in the next unit, Unit 2.

Let’s now demonstrate how the HTTP communication protocol works on your local computer by setting up a Web server and serving an HTML page to your Web browser. In the next activity you will set up the Apache Web server which is a cross-platform Web server; it compiles and runs on a variety of Unix and Windows operating systems. You will set up your Web server to communicate through the loopback address, 127.0.0.1 since your ISP probably has dynamically assigned the IP address of your home computer. The loopback address is the only fixed IP address that your Web server can rely on to be available session after session. If your computer is on a local area network with a fixed IP address, you will use that IP address to configure your Web server and then the documents you placed on your Web server will be available to the entire Internet. HTML documents served by your localhost Web server can only be retrieved from Web browsers on the localhost through the loopback network. Other remote Web browsers on the Internet will not be able to communicate with your localhost Web server.
Activity 1.4

1. Download the Apache Web server from http://httpd.apache.org that works on your operating system. Follow the instructions to install and configure the Apache Web server. If you have a static IP address, you can enter it as the server name during the setup process. Otherwise, you can use 127.0.0.1.

2. Install your Apache Web server in any directory on your computer. We will refer to this top-level directory as: $(Apache_rootdir)

3. Read the Web server’s configuration file in $(Apache_rootdir)\conf\httpd.conf and look for the IP address you just typed. This is the address your Web server uses to communicate. You will also see ‘80’ as the network port. You will learn more about network ports in your reading for Unit 2. For now, accept that a network port can be added onto an IP address.

4. Copy the HTML document you created called ‘mydoc.htm’ into $(Apache_rootdir)\htdocs

5. Start your Apache server by selecting an appropriate item on the Apache program group. After starting the server, a small icon will appear on the lower right corner of the screen. This icon is used to stop the server.

6. Type this URL: ‘<http://127.0.0.1/mydoc.htm>’ into your Web browser and see ‘mydoc.htm’ being served to you over the loopback network to your Web browser.

7. Type this URL: ‘<http://localhost/mydoc.htm>’ into your Web browser and retrieve the document. This demonstrates that 127.0.0.1 is the Internet address that maps to the ‘localhost’ network name.

8. Create your own HTML documents, use the HTML document you created in Example 1.2, or download any HTML page you like from the Web (with your Web browser use File — Save As) and install the HTML pages on your localhost Web server. View the HTML pages in your Web browser at:

‘http://localhost/(your_document_name).htm’

as described above. Experimenting in this way is a great way to learn how to use new HTML tags!
From this demonstration you should understand how the Web client and Web server communicate over the network when exchanging an HTML document. Answer the following question to test your understanding of the Web’s system architecture.

**Self-test 1.3**

The Web is a client-server architecture. What does this mean and how does this differ from a peer-to-peer architecture?
Summary

This unit has given you an overview of the components of the Web and answered the question ‘What is the Web? You have seen that the Web is the most popular Internet application and that the Web uses the Internet but it is not the Internet. The Web works on top of the Internet and cannot function without it. You should now know how pieces of the Web work together to form the information space that is delivered to your door as you start up your Web browser.

We have seen that three technologies make the Web ‘a web’ and distinguish it from other Internet services. These technologies are:

- the Web’s document language: the HyperText Markup Language (HTML);
- the Web’s system for addressing and locating documents: Universal Resource Locators (URLs); and
- the communication language between the Web client and Web server: the HyperText Transfer Protocol (HTTP).

In this first unit we explored HTML and URLs and you created your own HTML document. HTTP will be discussed in the next unit of this course.

The Internet, the Web browser and Web server software make the Web ‘world-wide’. Web server software delivers information and Web browser software receives the information and displays the results. To appreciate how the Internet, Web browser and Web server interact, you installed and configured an Apache Web server to serve your HTML document. The next unit in this course examines Web server software in detail and will look at the step-by-step mechanism by which a Web browser ‘talks’ to a Web server according to the HyperText Transfer Protocol (HTTP).

Whether your goal is to administer Web services or just be a more informed Web user, this unit has given you to the tools to dig deeper and investigate the building blocks of the Web.
Solutions to self-tests

Self-test 1.1

1. These three new technologies were created to build the Web:

   1. The HyperText Markup Language (HTML). HTML is the Web’s native document format which contains hypertext links.

   2. Universal Resource Locators (URLs) are the Web’s system for addressing and locating documents. Hypertext links contain URLs.

   3. The HyperText Transfer Protocol (HTTP). HTTP is the communication language or protocol between the Web client (the Web browser) and the Web server.

2. Hypertext is a system of a collection of documents that are associated through active links. When a user chooses a link, that link is followed, and the document that link pointed to is fetched and displayed. Hypertext is a non-linear text system that creates an ‘information space’.

3. A URL is the address of a unique location of an Internet resource. ‘http://www.lycos.com’ is the URL for the Lycos search engine. There are many examples. ‘ftp://ftp.ncsa.uiuc.edu’ is the URL for NCSA’s ftp site.

4. Hypertext links are represented as anchor tags in HTML. An anchor tag takes this form in HTML:

   <A HREF="http://www.ouhk.edu.hk"> The Open University of Hong Kong</A>

5. An inline image takes this form in a HTML document:

   <IMG SRC = "picture.gif"

6. These HTML tags should always be in a HTML document:

   <HTML></HTML>
   <HEAD></HEAD>
   <TITLE></TITLE>
   <BODY></BODY>

Self-test 1.2

1. A protocol is a set of conventions or rules specifying how each party should communicate. The details of Internet protocols are public domain, open standards. A network joins the Internet if it follows the communication rules specified in the TCP/IP protocol. An individual host computer can join the Web if it follows the communication rules specified in the HTTP protocol.
These are a few Internet services:

1. Domain Name Service (DNS) is an Internet service which maps Internet names to Internet addresses. Given an Internet domain name DNS will return an Internet address. Given an Internet address DNS will return an Internet domain name.

2. File Transfer Protocol (FTP) is an Internet service which allows users to copy files from one computer to another across the Internet.

3. Telnet is an Internet service which enables a person to set up a connection, login, and conduct an interactive session with a remote computer on the Internet.

4. The World Wide Web is a hypertext based Internet service. The WWW uses URLs as its addressing system and HTTP as its communication protocol.

3. Internet Protocol (IP) handles the addressing and coordinates the routing of packets across multiple Internet nodes.

4. TCP establishes a connection from one point on the Internet to another point on the Internet. Once the connection is established TCP is responsible for breaking the data up into packets and ensuring the reliable transfer of the packets over the network. TCP is responsible for detecting and correcting errors in the data transfer process. IP routes the packets across the nodes of the Internet. IP is the packet mover of the Internet.

Self-test 1.3
Client-server and peer-to-peer are two network service models. In both the client-server and peer-to-peer architectures two pieces of software collaborate to exchange messages. In the peer-to-peer architecture, the two pieces of software are identical. In the client-server architecture the client software and the server software are different; each piece of software has a distinct function. In the case of the Web client-server model, the Web browser (client) requests and receives information over the network, and then displays that information to the user. The Web server’s main function is to deliver information over the network.