MT834

Unit I

The Web and the Internet
Course team
Developer: Jenny Lim, Consultant
Designer: Chris Baker, OUHK
Coordinator: Dr Li Tak Sing, OUHK
Member: Dr Andrew Lui Kwok Fai, OUHK

External Course Assessor
Prof. Mingshu Li, Institute of Software, Chinese Academy of Sciences

Production
ETPU Publishing Team

Copyright © The Open University of Hong Kong, 2004.
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>What is the Web?</td>
<td>4</td>
</tr>
<tr>
<td>Design and structure</td>
<td>5</td>
</tr>
<tr>
<td>HTML</td>
<td>6</td>
</tr>
<tr>
<td>URLs</td>
<td>9</td>
</tr>
<tr>
<td>HTTP</td>
<td>16</td>
</tr>
<tr>
<td>The Internet</td>
<td>19</td>
</tr>
<tr>
<td>Design and structure</td>
<td>20</td>
</tr>
<tr>
<td>Communication protocols</td>
<td>22</td>
</tr>
<tr>
<td>Internet Protocol (IP)</td>
<td>23</td>
</tr>
<tr>
<td>Ports</td>
<td>29</td>
</tr>
<tr>
<td>Transmission Control Protocol (TCP)</td>
<td>30</td>
</tr>
<tr>
<td>Web servers</td>
<td>33</td>
</tr>
<tr>
<td>Role of browsers and servers</td>
<td>33</td>
</tr>
<tr>
<td>Installing a local Web server</td>
<td>34</td>
</tr>
<tr>
<td>Summary</td>
<td>39</td>
</tr>
<tr>
<td>Suggested answers to self-tests</td>
<td>40</td>
</tr>
<tr>
<td>References</td>
<td>43</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 that this course includes 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 how the Web works, you first need to have a good understanding of the underlying network which serves as its delivery medium — the Internet. Here are the aspects of the Internet we shall examine in this unit:

• the physical network;
• design and structure of the Internet;
• Domain Name Service (DNS); and
• Transmission (or Transport) Control Protocol/Internet Protocol (TCP/IP).

Next, we’ll discuss the technologies used to provide the World Wide Web service over the Internet:

• HyperText Transfer Protocol (HTTP);
• Uniform Resource Locator (URL);
• Web client and server software; and
• HyperText Markup Language (HTML) documents.

To learn these concepts, you will read selected online readings and conduct a series of hands-on exercises 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 the 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 may begin Unit 1 now by reading the unit’s learning objectives.
Objectives

By the end of this unit, you should be able to:

1. *Explain* how the World Wide Web, Domain Name Service, FTP and other applications are made available over the medium of the Internet.

2. *Discuss* the design and structure of the Internet, and the use of TCP/IP as its transmission protocol.

3. *Create* basic HTML documents and serve these documents from a local Web server.

4. *Describe* the role of Web browsers and Web servers.
Introduction

As a distance education student, the World Wide Web is an integral part of your life. In courses such as this one, you rely on it to communicate with tutors and fellow students, download study materials, and view online resources. This experience is bound to make you very familiar with the Web from an end-user’s point of view, that is, from the perspective of someone who requests information via a Web browser.

But what really goes on behind the scenes after you click on a hyperlink? How does your Web browser locate the document that you want and how does this document get transmitted to your machine? Where are these documents stored and how are they organized? For someone who is about to embark on a course in Web server technologies, you need a deeper understanding of the Web and the Internet beyond that of an experienced Web user.

In this unit, we will take our first steps towards gaining this knowledge. We will start with an in-depth discussion of the design principles, technologies and protocols behind the Web and the Internet. These fundamental concepts will underlie the Web server technologies to be discussed throughout the course.
What is the Web?

Computers have been connected to the Internet since the 1970s, and data exchange between networked computers has been around for just as long.

However, the launch of the World Wide Web in the early 1990s offered the prospect of something totally new. It allowed the entire Internet to be viewed as a single information space, where users accessing data could move seamlessly and transparently from machine to machine by following links. Before the Web, individual Internet computers had windowing systems and graphical capabilities, but networking applications such as email and FTP were still text-based. The ease, convenience and graphical nature of the Web has made it the ‘killer’ application of the Internet.

The Web, as it is commonly called, is a collection of interlinked information that is accessible through a worldwide network. It is a digital ‘information space’ with a means for users to access and retrieve documents from it. Here are the components of the Web (also shown figure 1.1 below) which make all this information storage, organization and retrieval possible:

1. Information in the form of multimedia documents. Multimedia means that these documents can be composed of text, images, animation, audio, video, and other types of content.

2. Computers where these documents are stored (known as servers or providers) and computers from which these documents are accessed (known as customers or clients).

3. A networking medium which connects these clients and servers so that data can travel between them, namely, the Internet. The Web was built to run ‘on top’ of the Internet, which exists independently of the Web.

4. Web client and server software which allows webpages on any Web server to be accessed from around the world. From the user’s point of view, all that’s needed is a Web browser and an Internet connection in order to get on the Web. From the information provider’s point of view, he/she needs a Web server which is connected to the Internet.

![Figure 1.1 Overview of the Web and its components](image-url)
For the rest of Unit 1, we will discuss the technologies, protocols and standards that are used by these different Web components.

**Design and structure**

Tim Berners-Lee proposed the World Wide Web in 1989 because he wanted a better way of sharing and retrieving information among the people who worked at the CERN (European Laboratory for Nuclear Research) office in Geneva, Switzerland. When he was designing the Web, these were some of his goals:

- To allow access to *different kinds of information* stored on *disparate computing platforms*. Common protocols had to be used to provide a bridge between different computer operating systems and networks.

- To use *hypertext*, or nonlinear text, that allows related documents to be tied together via ‘active links’, and that users can ‘follow’ by clicking on the links. The Web browser then fetches and displays the document pointed to by the link.

- To *decentralize* control and access. In order to get on the Web, all that was needed was access to the networking medium (e.g., the Internet) and software to retrieve and view the documents on it (e.g., the browser). There was no central node or computer to which everything had to be connected.

The Web’s architecture follows a standard *client-server* model. In this model, a user relies on a program (the client) to connect to a remote machine (the server), where the requested resource is stored. Possible resources could be text files, multimedia documents or dynamically generated pages.

Web clients, such as Internet Explorer and Netscape Navigator, know how to *present* data but do not need to know the details of how this data is stored or generated. Web servers, such as Apache and Internet Information Server (IIS), know how to *extract* data, but are ignorant of the details of how it will be presented to the user.

Tim Berners-Lee came up with three important new technologies for creating the Web:

1. HyperText Markup Language (HTML);
2. Uniform Resource Locators (URLs); and
3. HyperText Transfer Protocol (HTTP).

These were based on ideas which emerged in the last few decades. However, the technology needed to make hypertext systems a reality was only brought together in the early 1990s, with the birth of the Web. In the remaining part of this section, we will discuss these three technologies in more detail. Now answer the following questions to test your understanding of the Web’s design and structure.
Self-test 1.1

1 The Web uses the client-server architecture model. What does this mean and how can clients and servers become part of the Web?

2 What is the relationship between the Web and the Internet?

HTML

HTML, or HyperText Markup Language, is the markup language used to create documents on the Web. A markup language allows authors to highlight portions of the content and assign meaning to these portions by tagging them.

HTML documents are plaintext or ASCII files that can be created using any text editor on any machine. Tim Berners-Lee chose the plaintext format because it could be understood by all computers, regardless of their operating system or hardware platform. This is an important factor behind the universality and accessibility of the Web.

HTML documents contain a combination of text and markup tags. The markup tags specify the logical structure and organization of a document, for example, which parts belong to the head and body of the page. Here are the basic tags which must be found in every HTML document.

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

Figure 1.2 Basic HTML document tags

Aside from these basic tags, there are other tags which can be used to mark up parts of the Web document body, such as the paragraphs, headings, lists, quotes, definitions, citations, etc.

The interpretation of these marked elements are left to the browser. This choice was made because the same HTML document may be viewed by different browsers of varying abilities.

Here is a more detailed example of the elements that may be found in an HTML document.
### HTML is Easy To Learn

Welcome to the world of HTML. This is the first paragraph. This *text is bold.*

And this is the second paragraph. This text is *emphasized.*

Here are 3 reasons to learn HTML:

- You can build your own home page.
- You can start an online business.
- You can share your photo albums.

---

**Figure 1.3** Elements that may be found in an HTML document

Aside from defining the structure of the document, markup tags can also be used to identify the hyperlinks within the page. Just as there are HTML tags for representing formatting directives, there are HTML tags called *anchor tags* for representing HTML links, i.e., anchors, to other Web resources. When you use your mouse to point-and-click 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 *The Open University of Hong Kong* with a hypertext link to the OUHK’s homepage would be written like this:

```html
<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

We will discuss how a Web address, or Uniform Resource Locator (URL) is formed a little later in the section titled ‘URLs’.

Although you’ve now had a basic introduction to HTML, we’ve only covered a very small portion of its available tags. From its origins as a structural markup language, HTML has also grown to include formatting tags which describe how elements should appear (e.g., in what colours and sizes).

If this is your first exposure to HTML, the following reading will give you a good foundation in the important tags that are needed to build webpages. The first item in the reading is a highly readable introduction to HTML. I’ve also provided a shorter item, item 2, that you can skim through if you want to refer to some basic guidelines for constructing a simple HTML document.

---

**Reading 1.1**


---

When you have completed Reading 1.1 you should be able to construct an HTML document containing HTML tags for these elements:

1. bold text;
2. italicized text;
3. underlined text;
4. paragraph;
5. heading;
6. hypertext link or hyperlink;
7. inline image;
8. background graphic;
9. font colour; and
10. background colour.
Activity 1.1

The files needed for the following activity can be found on the course website.

ABC Books has just decided to establish an online presence and they have given you a document containing the information they wish to appear on their homepage.

1 Please download the document called abc_home.doc (MS Word format).

2 Convert this information into an HTML document called abc_home.html. Code this page by hand, using the tags you’ve learned in this section.

Note: If you have difficulty with this or any of the other activities, please seek help on the MT834 discussion board, or contact your tutor.

URLs

In the previous section, you learned that HTML uses the anchor tag <A> in order to tell the Web browser where an information resource is located on the Internet. Uniform Resource Locators (URLs) are used within anchor tags to specify a unique online address for each resource, just as street addresses express the unique location of a place in our physical world.

From your travels on the Web you are probably familiar with the basic form of a URL, as follows:

http://hostname/path/filename.html

A Web URL includes the protocol name (http), followed by a colon and a double slash (://), followed by the Web server name or address, and the location of the file on the server. If a filename is not specified, then the server will return the default homepage. Using the above example, you should recognize that the URL for the Open University of Hong Kong homepage would be http://www.ouhk.edu.hk.

The next reading gives you more details on how different kinds of URLs can be coded in HTML pages.
Now that you’ve got an idea of how webpages and hyperlinks are coded using HTML, let’s take a closer look at the software that is used to interpret and display HTML documents, namely, the Web browser.

Activity 1.2

This activity requires you to have two Web browsers installed on your machine so that you can contrast how each Web browser handles common browser functions. The browsers used for the screen shots in this activity are Internet Explorer 5.5 and Netscape Navigator 6.0.

You may use any reasonably current version of these two browsers. If you need to upgrade to a later version or you need to acquire new browsers, you can download

- Netscape Navigator Web browser at http://channels.netscape.com/ns/browsers/default.jsp; and


Follow the instructions for downloading and installing a new version of each Web browser for your operating system.

Aside from displaying HTML documents, Web browsers also come with common menu-based functions. Functions are accessed via menu bars along the top of the browser window. An example of a common function would be configuring the browser to use certain fonts, text sizes or languages. Browsers from different vendors may label these menu options differently or place them in slightly different locations. Let’s compare and contrast how Netscape Navigator and Internet Explorer handle some common functions.

1 Viewing or altering the configuration settings of the Web browser

   - For Internet Explorer: **Tools → Internet Options.**
   - For Netscape Navigator: **Edit → Preferences.**
You will see common configuration settings for the browser such as font colours and language preferences. We will be altering some of these configuration settings in future units. Let’s change the default font colours.
2 Viewing the properties of an image

Point each Web browser at the URL for the Open University of Hong Kong’s homepage at http://www.ouhk.edu.hk. Position your mouse over an image in the webpage.

- For Internet Explorer, right-click on the mouse ➔ select Properties.
- For Netscape, right-click on the mouse ➔ select View Info.

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

Here’s what the Properties window looks like after I right-click on an image from OUHK’s homepage.

![Properties Window](image)

Figure 1.6 The image Properties window

3 Viewing the source of the HTML document

Point each Web browser at the URL for the Open University of Hong Kong’s homepage at http://www.ouhk.edu.hk.

- For Internet Explorer: View ➔ Source.
- For Netscape Navigator: 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.

Can you find the basic tags `<HTML>`, `<HEAD>`, `<TITLE>`, and `<BODY>` in the HTML source of the Open University of Hong Kong’s homepage?

**MIME types**

The HTML documents we’ve seen so far are all written and encoded using the ASCII or plaintext format. However, URLs not only point to HTML documents, but may also point to multimedia resources that are encoded in formats other than text. For example, images are usually in GIF, JPEG, or PNG format, while audio files might use MPEG, AU, or MP3.

It is impossible for Web browsers to have the built-in capability to render every media format that is available. So how does the browser know how to handle those formats for which it does not have a native rendering capability?

Multipurpose Internet Mail Extensions (MIME) types are the answer. MIME is an international standard that defines the rules for exchanging information that uses non-ASCII text encoding. It enables the client to know what type of file to expect and what software to use to interpret the file, in case the client is not capable of understanding this type of encoding.

MIME also defines a standard set of names for the different data formats that could be transmitted over networks. The names come in two parts: the file type, followed by a slash (/), followed by a subtype. The following table lists some of the better-known MIME file types and subtypes, together with helper applications that can be invoked by the client to display them.

<table>
<thead>
<tr>
<th>MIME file types and subtypes</th>
<th>File extensions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application/msword</td>
<td>doc</td>
<td>Microsoft Word document</td>
</tr>
<tr>
<td>Application/zip</td>
<td>zip</td>
<td>Compressed file that can be opened using PKZip, WinZip or other file compression software</td>
</tr>
<tr>
<td>Image/jpeg</td>
<td>jpeg, jpg</td>
<td>JPEG image file, which can be opened natively by the browser and by graphics editors such as Adobe Photoshop, Macromedia Fireworks, etc.</td>
</tr>
</tbody>
</table>
The Web server can be configured to send the correct MIME type to the Web browser along with the requested resource. The Web browser examines the MIME type and displays the resource if it has the native capability to do so. If not, it can be configured to launch the appropriate helper application that can handle this resource. This is illustrated in the next activity.

### Activity 1.3

Let’s compare the approach used by Netscape Navigator and Internet Explorer in setting up and recognizing MIME types. This activity also illustrates how different vendors implement the same features in their software in different ways, further reinforcing what you’ve seen in the previous activity.

1. Using both browsers, visit the OUHK website and click on the Students link. Then, choosing the Information tab, go to Prospectus and then select View online. This link will take you to the Adobe Acrobat document (PDF) that contains the University prospectus. How did both browsers handle this file type?

2. Now let’s see how Netscape and IE were configured to use Adobe Acrobat to render the PDF file.

   For Netscape Navigator, the configuration screen can be accessed via these menu options: **Edit → Preferences → Navigator → Applications.**
For Internet Explorer, the MIME types are embedded in the Windows operating system. On my system (Windows 2000 Professional), the MIME types can be found by clicking on: My Computer → Tools → Folder Options → File Types. The resulting display shows a list of all the registered MIME types on the system.
One of the most important concepts when learning how clients and servers communicate over a network is protocol.

Douglas Comer defines a protocol as ‘a formal description of message formats and the rules two or more machines must follow to exchange those messages’ (cited in Connected: An Internet Encyclopedia; Programmed Instruction Course, ‘Protocols’, http://freesoft.org/CIE/Course/Section1/3.htm). Protocols usually exist in two forms. First, they exist in a textual form for humans to understand. Second, they exist as programming code for computers to execute.

Whenever a computer needs to send data to or receive data from another host, a protocol is needed to specify how every bit of every message should be written and interpreted. The protocol also describes how to handle error conditions. HyperText Transfer Protocol, or HTTP, is the protocol used by the World Wide Web service. HTTP describes how Web clients make requests for information and how servers respond to these requests. Other services, such as email and FTP, also have their own protocol specifications.

HTTP is an openly published standard protocol, which allows browsers and servers written by different vendors to communicate with each other, as long as their software speaks and understands HTTP.

Like most network protocols, HTTP uses the client-server model: an HTTP client opens a connection and sends a request message to an HTTP server; the server then returns a response message, usually containing the resource that was requested. After delivering the response, the server closes the connection. This makes HTTP a stateless protocol which does not retain any connection information between transactions.

HTTP messages are text-based and can be made up of several lines. Figures 1.9 and 1.10 are sneak previews of what HTTP messages can look like. Figure 1.9 shows an HTTP request sent by a Web client. The first line describes what the client wants to do (e.g., get a document) and includes the URL of the desired document. This could be followed by an optional number of headers. In figure 1.10, you see that successful requests return a status code of 200 in the first line. There could be other response headers attached, followed by the requested document itself.

```
GET /index.html HTTP/1.0
Host: www.ouhk.edu.hk
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
```

Figure 1.9  Sample HTTP request sent by a Web client
HTTP/1.1·200·OK
Date:·Fri,·10·Oct·2003·07:44:55·GMT
Server:·Apache/1.3.22·(Unix)·mod_jk/1.1.0·mod_ssl/2.8.5·OpenSSL/0.9.6b
Last-Modified:·Wed,·18·Jun·2003·09:43:11·GMT
ETag:"250b81-5a62-3ef0342f"
Accept-Ranges:·bytes
Content-Length:·23138
Connection:·close
Content-Type:·text/html

<html>
<head><title>Sample Response</title></head>
<body>Hello, world !</body>
</html>

Figure 1.10  Sample HTTP response sent by Web server

The rules for constructing valid HTTP messages will be explained in more detail in Unit 2. However, the next activity allows you to experiment with different requests and examine the HTTP messages that are generated.

Activity 1.4

In this activity, you view the HTTP messages exchanged by your Web browser and the Web server using the online HTTP viewer at http://www.rexswain.com/httpview.html.

1 Here are some pages to access: www.ouhk.edu.hk and www.yahoo.com.

2 For each HTTP request, identify where the URL is located.

3 For each HTTP response, identify where the status code and the HTML document is located.

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. Suggested answers are provided at the back of the unit.

Self-test 1.2

1 What three new technologies were created to build the Web? What does 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 is a network. More accurately, the Internet is a network made up of many connected and cooperating computer networks. All these networks communicate using the same methods or protocols. These interconnected networks spanning the entire globe are called an internetwork, hence the term Internet.

You can visualize the Internet as a giant, global plumbing system, similar to the pipes that bring water to your homes, except it’s used to carry digitized data. The Internet in itself is useless if there is no traffic travelling on it, and as you’ve seen previously, email and the Web are the most popular services which use the Internet as their transport medium.

The Internet was a product of the Cold War. It began as an effort to create a communication network that could withstand nuclear attacks. 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. None of the existing networks of the time could handle this requirement, including the telephone system which was vulnerable to attacks on its switching stations. Before the Internet, computers had to be directly connected to each other if they needed to communicate. Messages sent from host computer A to host computer B could only travel on a single, fixed route. There were no alternative paths that could be used in case this direct route was destroyed.

Therefore, a new type of network was designed and built that could fulfill the requirements of a wartime network: one that we now call the Internet.

The following online reading outlines the history of the Internet and its unique design which allows it to handle failure and re-route traffic around trouble spots.

Reading 1.3


Note: You do not have to remember the historical facts and organizations in the article such as RAND, MIT, UCLA, ARAPANET and Pentagon.

There are two important characteristics of the Internet which stand out in Reading 1.3.

First, the Internet is decentralized. There are no top-level computers or routes that can fail and stop the operation of the entire network. There
can also be multiple, redundant routes between two computers, so if one route becomes unavailable, other routes can still be used.

Second, the Internet is a packet-switching network. Messages are broken into discrete units called ‘packets’ which contain the addresses of the source and destination host. These packets are routed separately through intermediate nodes, hopping from one to another until they reach their intended destination. If a packet fails in traversing the network via a particular node, it is simply resent, again, and again, across another network path or ‘route’ until it reaches its destination. When all the packets have arrived at their destination, they are reassembled to form the original message. This design makes it possible for two computers to communicate with each other even if there is no direct, dedicated network cable running between them.

![Figure 1.11](image)

*Figure 1.11* Packets travel from the source to the destination via intermediate gateways, or routers, in a packet-switching network such as the Internet

Nobody owns or controls the Internet, although companies and even governments might be responsible for building and maintaining portions of it. Rather, any network that voluntarily implements the Internet’s standard protocols may participate in it. Many Internet providers not only adhere to these standards, but also open up their networks to data traffic from the general public. The voluntary interconnections and cooperation between these network providers make the global Internet possible.

**Design and structure**

The Internet is a network of 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 or computers on the network ‘connect’ to exchange messages in some form. A network connection can have a variety of physical forms as shown in figure 1.12.
Data is transmitted over various carriers, such as telephone lines, cable TV wires, and satellite channels. When you dial in to your Internet Service Provider (ISP), your computer actually becomes a node on the ISP’s network, and from here, you gain access to 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 components can ‘talk’ to one another. The Internet network software operates on top of these communication layers. The magic of the Internet is how these very different computer networks cooperate to form one internetwork — the Internet.

The next reading describes the networks and the interconnections between them which form the Internet’s infrastructure.

**Reading 1.4**


*Note:* Please only read the first three sections: (1) ‘Introduction to how Internet infrastructure works’, (2) ‘A hierarchy of networks’, and (3) ‘Bridging the divide’.

Now complete the following self-test to assess your understanding of the Internet’s design and structure.

**Self-test 1.3**

1. Describe two design characteristics of the Internet that make it well-suited for carrying wartime communications.

2. What is the relationship between the networks that make up the Internet?
Communication protocols

The Internet has the ability to provide a bridge between different computer operating systems and networks. This is why Tim Berners-Lee was interested in providing the World Wide Web service over the Internet.

It is relatively easy to add a new network to the Internet. The communication protocols used are built on standards that are open and publicly available. Owners of diverse physical network types or different computer operating systems can join the Internet simply by implementing or purchasing the appropriate protocols.

The task of transmitting data across the Internet is divided up among several network protocols. Protocol layering is a common technique for simplifying networking designs. The work is divided into functional layers, and separate protocols are assigned to perform each layer’s job. This approach leads to a set of simple protocols, each taking care of a few well-defined tasks.

A layer communicates with the layer above and below it, but it is not aware of layers which are not directly adjacent to it. The key protocols for the Internet are IP (Internet Protocol) and TCP (Transmission or Transport Control Protocol), with IP operating on the layer below TCP. The next reading describes how the Internet’s network model is organized as four layers of protocols.

Reading 1.5


*Note:* A link at the bottom of this reading takes you to a discussion on the topic of encapsulation. We refer to encapsulation a little later, so please take a quick look at this link also.

In the previous reading, the topmost layer of the networking model is where Internet services such as telnet, FTP, WWW and email (SMTP) operate.

The lowest layer is responsible for passing data packets on to the physical network cabling media. The popular protocols at the network access layer are PPP (point-to-point protocols) for Internet connection over regular phone lines and Ethernet protocol over Ethernet-based local area networks.

IP is concerned with the routing of data between sender and recipient. It does this by attaching a source and destination address to each packet, like an envelope.
TCP relies on IP to handle the details of getting data from one place to another. On top of it, TCP provides mechanisms for establishing connections between host computers, ensuring that data arrives in the correct sequence, and retransmitting packets that are not received correctly or promptly.

Depending on the literature, different names may be given to the layers in the network model shown in Reading 1.5. To avoid confusion, I’d like to present the model here again, along with other names that are commonly used for the various layers.

![Four-layer Internet network model](image)

**Figure 1.13** Four-layer Internet network model

The way that the application, transport, Internet and physical layers work together is called *encapsulation*. The application produces some data, adds a header to it and hands off the result to the transport layer. The transport layer adds another header, and hands the result off to the Internet layer. It’s like putting a letter in an envelope, then putting that envelope in a bigger envelope, and so on. On the receiving end, the network software unpacks the envelopes one layer at a time until the original data is handed to the receiving application.

Now that you have a high-level idea of how TCP and IP are related to each other, let’s look at the details of how they work together.

**Internet Protocol (IP)**

IP manages the transfer of data across physically diverse networks. IP transfers data in pieces, called *packets* or *datagrams*. Each packet is encapsulated within an envelope of data describing where the packet came from and where it wants to go. Packets are transferred from one network to another according to the rules of the IP protocol, until it arrives at its destination. Networks are fallible, though, so some packets may be lost, delayed, or garbled along the way.
Back in Reading 1.4 ‘How Internet infrastructure works’, you learned that routers are special gateway hosts which join different networks together. Routers are like traffic cops who stand at intersections on the Internet highway and decide if a packet is intended for a host within its own network or needs to be routed to a different network. If the packet needs to be forwarded to another network, the router uses its own routing protocols to determine where to send it next.

IP specifies the formatting used to create packets and the addressing scheme which gives every computer on the Internet a unique address. A detailed discussion on the formatting specification of IP datagrams is beyond the scope of this course, but the next figure is meant to illustrate how an IP packet may look like once it is broken down into its component fields.

<table>
<thead>
<tr>
<th>VERSION</th>
<th>HEADER LENGTH</th>
<th>SERVICE TYPE</th>
<th>TOTAL LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTIFICATION</td>
<td>FLAGS</td>
<td>FRAGMENT OFFSET</td>
<td></td>
</tr>
<tr>
<td>TIME TO LIVE</td>
<td>PROTOCOL</td>
<td>HEADER CHECKSUM</td>
<td></td>
</tr>
<tr>
<td>SOURCE IP ADDRESS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESTINATION IP ADDRESS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP OPTIONS (IF ANY)</td>
<td>PADDING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA / PAYLOAD.....</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.14** Format of an IP datagram, containing the header and data area

*Source: Comer 1991, 92.*

Some of the key fields in the IP packet are:

- **Time-to-live (TTL)** — limits the number of routers that a packet may go through before reaching its destination. This prevents IP packets from traveling on the Internet forever.

- **Protocol** — lets the networking layer know what kind of transport layer protocol is in the data segment of the IP packet. Common transport layer protocols which use IP are TCP and User Datagram Protocol (UDP).

- **Source and destination IP addresses** — are the IP addresses of the source and destination machines.

- **Data/Payload** — contains the data which needs to be transmitted to another computer. This data is passed down to IP by a transport protocol such as TCP or UDP, as indicated by the protocol field.
Activity 1.5

The Traceroute utility allows you to follow the path, or route, traveled by a packet through the network. We will use this command to view the number of hops that are traveled by a particular packet over the Internet.

1. If you are on a Windows machine, go to the MS-DOS command prompt and type the following command:

   `tracert www.google.com`

   If you are on UNIX, type the following command:

   `traceroute www.google.com`

   The output will display the intermediate routers that your packet has to go through in order to arrive at its destination (www.google.com). It also records the round-trip travel time for each router. My packet had to travel more than 15 hops to reach Google.

2. Now let’s view the route to a Hong Kong-based server, namely, OUHK. If you are on a Windows system, type this command:

   `tracert www.ouhk.edu.hk`

   If you are on UNIX, type this command:

   `traceroute www.ouhk.edu.hk`

   Here’s the output of traceroute against OUHK’s Web server.

   ![Output of traceroute against OUHK's Web server](image)

   My packets had to go through many fewer hops (only eight) in order to reach OUHK as compared with Google. This explains why a website should be hosted near its primary audience for better download times.
Traceroute is very useful for debugging problems within a network. If you are unable to reach a destination server or if response time is slow, you can use this utility to pinpoint problem areas and slow links.

**IP addressing**

We have learned that IP defines a system for assigning unique addresses to all devices connected to the Internet. This is analogous to the system used by Hong Kong’s postal service to locate residences and businesses through street names and numbers. The next reading shows you how IP addresses are formed.

---

**Reading 1.6**


The reading describes how IP addresses are organized into two parts. The first part identifies the network and the last part identifies a specific host computer on that network. When data is routed on the Internet, the network portion of the IP address is used to locate the correct network. Once the data has arrived at the local network, the host portion of the IP address is used to identify the correct computer within this network for which the data is intended.

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, more general 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; and
- 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 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.

**Activity 1.6**

In this activity, you will view the current IP address assigned to your machine. You must connect to the Internet first so that your computer now becomes a host on the network. Depending on your Internet connection, your ISP may assign you a static IP address or a dynamically assigned IP address. A static address means your machine will always be assigned the same IP address, while a dynamically assigned address will be chosen from the pool of available addresses at the time.

1. If you are working on a Windows machine, you can view your current IP address by typing `IPCONFIG` on the command line. If this does not work on your system, try `WINIPCFG`.

2. If you are on a UNIX machine, type `nslookup` to display the IP address of the machine.

**Domain names**

IP addresses such as 202.40.157.163 are difficult to memorize and discuss, unless you’re a computer ☺. An alternate way of addressing hosts using alphabetic or word-based names, called domain names, was created. Examples of popular domains include amazon.com, yahoo.com, and google.com. OUHK’s Web server corresponds to the domain name ouhk.edu.hk.

Domain names are not just easier to remember, they also allow the underlying IP addresses to be changed without affecting the name by
which the outside world knows them. You can think of the domain name as the pseudonym or alias; the real name of the host computer is still its IP address. It is up to the Domain Name Service to translate a given domain name to its actual IP address.

The next reading describes how this translation process is accomplished.

**Reading 1.7**


*Note*: Please only read the fifth section, ‘What’s in a name?’

Let’s conduct a few experiments to see how DNS maps domain names to Internet addresses.

**Activity 1.7**

In this activity, we will use the `nslookup` utility to translate domain names to their corresponding IP addresses. The same command can be used on both Windows and UNIX systems.

1. In the command line, type `nslookup www.yahoo.com`. Here’s the output on my system.

```plaintext
C:\>nslookup www.yahoo.com
Server:  ns1.so-net.com.hk
Address:  203.99.142.8

Non-authoritative answer:
Name:  www.yahoo.akadns.net
Addresses:  66.218.70.49, 66.218.71.86,
             66.218.71.90, 66.218.71.95,
             66.218.71.80, 66.218.71.92,
             66.218.71.91, 66.218.70.48
Aliases:  www.yahoo.com
```

**Figure 1.16** Output of `nslookup www.yahoo.com`

It turns out that there are multiple IP addresses which correspond to the domain name www.yahoo.com. This means that Yahoo uses more than one Web server to handle the high volume of incoming traffic on its website.

The `nslookup` command also returns the name and IP address of the domain name server (DNS server) that was used to do the translation. Can you tell what my DNS server is from the above display?
2 Instead of using the domain name http://www.yahoo.com in the URL, enter two of the numeric IP addresses above into your Web browser to access Yahoo’s website (e.g., http://66.218.71.80 and http://66.218.70.48). Both URLs should display Yahoo’s homepage, which proves that they both map to the same domain name.

3 Now find out the IP address for www.kfbg.org.hk (the website of Kadoorie Farm and Botanic Garden). In this case, the domain name maps to a single numeric address.

## Ports

Ports are numbers assigned to software applications or services running on a computer. For example, a computer at an Internet Service Provider (ISP) might be running a Web server, an email (SMTP) server and an FTP server. How does a Web browser on a client PC say, ‘I want to speak to the Web server,’ and an email client say, ‘I want to speak to the email server.’? In order to identify the desired service, the client also needs to know which port number on the remote server has been assigned to the desired service.

Table 1.2 shows the list of established port numbers for common services. For example, if a server machine is running a Web server and a File Transfer Protocol (FTP) server, the Web server would typically be available on port 80, and the FTP server would be available on port 21. Clients connect to a service at a specific IP address and on a specific port number. Once a client has connected to a service on a particular port, it accesses the service using the protocol for that service.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>echo</td>
<td>7</td>
<td>Allows one machine to ‘echo’ back the input received from another machine</td>
</tr>
<tr>
<td>FTP</td>
<td>20, 21</td>
<td>Allows files to be exchanged between machines</td>
</tr>
<tr>
<td>telnet</td>
<td>23</td>
<td>Used to log in to remote machines</td>
</tr>
<tr>
<td>SMTP</td>
<td>25</td>
<td>Sends email between machines</td>
</tr>
<tr>
<td>HTTP</td>
<td>80</td>
<td>Used by Web browsers and servers</td>
</tr>
<tr>
<td>POP3</td>
<td>110</td>
<td>Transfers emails stored on a host machine to a client machine</td>
</tr>
</tbody>
</table>

Some of these Internet services are described further in the next reading.
Reading 1.8

Web Developer’s Virtual Library, ‘Internet protocols’ (by Alan Richmond), http://www.wdvl.com/Internet/Protocols/.

As you can see, the Web is only one of many services that run over the Internet!

Transmission Control Protocol (TCP)

IP provides a best-effort service which strives to deliver packets to their destination but does not guarantee that the delivery will be successful. For an application such as the World Wide Web, IP’s best-effort level of service is not enough.

Transmission (or Transport) Control Protocol (TCP) is needed in order to manage and provide a reliable connection between two computers. You can visualize a TCP connection between two hosts as a pipeline with two endpoints. TCP segments or packets are put in one end and come out of the other end. Data can be exchanged in both directions at the same time.

Figure 1.17 TCP is a connection-oriented protocol
Source: Kurose and Ross 2000.

TCP runs on top of IP and ensures that all IP packets which make up the same message are transmitted safely, completely and correctly to their destinations. TCP waits for the recipient to send back an acknowledgement message for each packet that has been sent before it sends out the next group of packets. If the recipient does not acknowledge receipt within a designated amount of time, the client TCP will resend the packet. The recipient may also send back an acknowledgement message which asks the sender to retransmit the packet if data has been corrupted or was not received in the correct sequence.

The TCP protocol only runs in the source and destination host computers. Intermediate network elements such as routers and bridges merely forward IP packets without knowing which ones belong to the same message or are part of the same ‘connection’. This demonstrates protocol layering at work for you.
Having taken a brief look at the TCP connection, let’s examine the TCP segment structure. Similar to IP, the TCP segment consists of header fields and a data field.

<table>
<thead>
<tr>
<th>SOURCE PORT</th>
<th>DESTINATION PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQUENCE NUMBER</td>
<td></td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT NUMBER</td>
<td></td>
</tr>
<tr>
<td>HLEN</td>
<td>RESERVED</td>
</tr>
<tr>
<td>CHECKSUM</td>
<td>URGENT POINTER</td>
</tr>
<tr>
<td>OPTIONS (IF ANY)</td>
<td>PADDING</td>
</tr>
<tr>
<td>DATA / PAYLOAD....</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.18** Format of a TCP segment with a TCP header followed by data  
*Source: Comer 1991, 183.*

Some of the key fields are:

- Source and destination port numbers — used by the two communicating hosts.

- Sequence number — shows the position of this data segment within a group of segments. TCP breaks up a message into packets and then labels each packet with a sequence number. This allows the two communicating hosts to know what packets have been received and which ones have not. It is also used to determine the order in which packets should be reassembled.

- Acknowledgement number — returned to the sender to acknowledge receipt of a particular packet. It also informs the sender of the sequence number of the next byte that the sender expects from the recipient.

- Checksum — used to verify that the data was not corrupted during the transmission.

At this point you should understand TCP/IP, IP addresses, ports and domain names. This should give you a basic understanding of how the Web depends on the Internet’s protocols, infrastructure and services. Do the following self-test to check your knowledge, and then check your answers against those at the end of the unit.
Self-test 1.4

1. What is a protocol? Why are protocols important to the Internet?

2. Name three Internet services that use TCP/IP and describe what each service does.

3. What is the role of IP within the TCP/IP suite of protocols?

4. What is the role of TCP within the TCP/IP suite of protocols?

5. What are some of the things that can go wrong when IP packets are transmitted over a network?
Web servers

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 can participate 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 written in a wide variety of computing languages: C, C++, Java, etc.

This section presents a conceptual overview of how browsers and servers communicate with each other, and how they perform their functions by making use of lower-level services such as TCP/IP and DNS.

Role of browsers and servers

A browser is an HTTP client because it sends requests to an HTTP server (Web server), which then sends responses back to the client. The standard (and default) port for HTTP servers to listen for incoming requests is port 80.

![Diagram of Web browser requesting documents from a Web server](http://computer.howstuffworks.com)

Figure 1.19 Web browser requesting documents from a Web server


Please note that there can be HTTP clients that are not Web browsers. Programs written in a language such as Java or C can also issue HTTP requests to a Web server, thereby acting as an HTTP client. For the purposes of MT834, however, we will deal mostly with Web browsers acting as Web clients.

Here are the basic steps that take place behind the scenes in order to satisfy a Web request:

1. The browser gets the server name and the filename (including the path) of the requested resource from the URL.
2. The browser asks a Domain Name Server to translate the server name www.ouhk.edu.hk into an IP Address, which uniquely identifies the Web server on the Internet.

3. The browser connects to the server on port 80.

4. The browser sends a request to the server which is written according to the HTTP specification. The request will ask for the file http://www.ouhk.edu.hk/index.html.

5. The server sends the HTML text for the webpage to the browser using the HTTP protocol as well.

6. The browser interprets the HTML tags and displays the page on your screen.

Steps 2 to 6 require the use of TCP/IP’s services. Let’s tie everything together and see how these protocols are used in a typical Web surfing session. The protocols are shown using the Internet four-layer model.

![Diagram of TCP/IP layers](image)

**Figure 1.20** Web clients and servers use lower-level services in order to communicate over the network

### Installing a local Web server

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). However, Apache was originally written to run on UNIX servers, and new versions of Apache stabilize much faster on UNIX than Windows. For the sake of reliability and security, it is recommended to run Apache on the UNIX operating system.

The activities and practical work for the rest of this unit will assume that you are using the Apache Web server on the Linux operating system. Linux is an operating system that was developed under the GNU General Public License and its source code is freely available to everyone. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems.

Since there is no single company that controls Linux, several organizations and individuals have developed their own ‘versions’ of the Linux operating system, known as distributions. A Linux distribution is based on Linus Torvalds’s Linux kernel, which contains the core functions of the operating system.

We will assume that most of you are going to use Linux on PCs with Intel-based CPUs. Companies that have created distributions for Intel-based PCs include Debian, Red Hat and SuSE. You are free to choose which distribution to acquire and install, but it’s a good idea to acquire a copy of Linux that already includes Apache and mySQL. If you set up your Linux machine with Internet access, you can also download the necessary software later on.

Describing how to acquire Linux and anticipating all the different variations during the installation process is beyond the scope of this course. If you plan to install Linux on your own, you may refer to the comprehensive online resources in the next reading. To keep it simple, I’ve only included links for the Debian/GNU Linux and Red Hat distributions.

**Reading 1.9**


Linux can be installed on a dedicated system or on a dual-boot configuration, where it co-exists with another operating system such as Windows on the same machine. If you are planning to install Linux on a dual-boot configuration, make sure you do a complete backup of your
existing system. There is always a possibility that you may lose all the data contained on your drive when you work with the hard disk partition table. Because of this, it’s usually recommended for beginners to install Linux on a dedicated machine.

If you’re planning to connect your Linux box to the Internet, here are the different ways to do so:

- LAN;
- dialup; and
- broadband.

If you are using broadband, you need to know which technology is used by your ISP. The following link shows this information:


Note that there are mainly two technologies used, namely PPPoE and DHCP. The following links show how to configure the network connection:

- For Redhat: Select RedHat (small icon on the left bottom corner) → System settings → Network. You will be prompted for root’s password. If you are using LAN, then you should double click on the network connection to configure the IP address and the gateway. If you are using dialup connection or PPPoE, you should click on the New button and then select modem connection and xDSL connection respectively. Then follow the instructions to complete the configuration. If you are using DHCP, you should double click on the Ethernet connection and then configure the connection to use DHCP.

After you have acquired a Linux machine, you will install the Apache Web server and browse the pages on it through the loopback address, 127.0.0.1. If you’ll recall from the section on IP Addressing, this is a specially reserved address which directs TCP/IP traffic back to the local machine. The loopback address corresponds to the name alias of localhost. Entering http://127.0.0.1 or http://localhost in your browser will both display the homepage of your local Web server.

If your machine is currently on the Internet, your ISP may also have dynamically assigned an IP address to your home computer. You cannot depend on this address to be the same every time you connect. The loopback address is the only fixed IP address that your Web server can rely on to be available session after session, and this is what we’ll use to test it.

HTML documents served by your localhost Web server can only be retrieved by Web browsers running on the same machine. Web browsers on the Internet will not be able to communicate with your localhost Web server since they do not know your IP address.
If your computer is on a local area network and has a fixed IP address, you will use that IP address to configure your Web server and then the documents you placed on it will become available to the entire LAN (and possibly, depending on your network set-up, the Internet).

Most Linux distributions already come with the Apache Web server. You may skip the next activity if you already have Apache installed on your Linux machine.

**Activity 1.8**

2. Read and follow the instructions to install and configure the Apache Web server. If you have a static IP address, you can enter it during the setup process. Otherwise, you can use 127.0.0.1.
3. Install your Apache Web server in any directory on your computer. We will refer to this top-level directory as `$(Apache_rootdir)`.
4. Read the Web server’s configuration file in `$(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, or the address at which this application can be reached on a particular machine.
5. Copy the homepage you created for ABC Books in Activity 1.1, called `abc_home.html` into `$(Apache_rootdir)/htdocs`. You should also copy all necessary images into a separate folder, such as `$(Apache_rootdir)/htdocs/images`. Ensure that the HTML document refers to the images with the correct pathname.
6. Start the Apache server from the command line using the `httpd` command or the `apachectl` script, if available. Restarting or stopping the server can be done using the `kill` command or the `apachectl` script as well. Please consult the Apache documentation for more details.
7. Acquire and install a Web browser for your Linux machine, since you will be browsing your webpages on the local Web server. The most popular browsers for Linux are Mozilla (http://www.mozilla.org), Netscape Navigator (http://www.netscape.com) and Opera (http://www.opera.com).
8. Type in the URL [http://127.0.0.1/abc_home.html](http://127.0.0.1/abc_home.html) into your Web browser and see `abc_home.html` being served to you over the loopback network to your Web browser.
Type in the URL http://localhost/abc_home.html into your Web browser and retrieve the document. This demonstrates that 127.0.0.1 is the Internet address that maps to the localhost computer name.

From this demonstration you should understand how the Web client and Web server communicate over the network when exchanging an HTML document. Do the following self-test to check your understanding of the Web’s system architecture.

**Self-test 1.5**

Describe the steps that take place so that your requested document can be fetched and displayed on your computer.
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 one of the most popular applications using the Internet as its transmission medium. The three technologies that make up the Web 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 more detail in Unit 2 Web servers and HTTP.

We also discussed the Internet — its characteristics, design features, and how it is related to the Web. The Internet is the underlying network that carries Web traffic consisting of HTTP request and response messages. This network uses a unique design meant to fulfill the requirements of a wartime network. We also examined the TCP/IP network protocol used to transmit data over the Internet. Essentially, HTTP messages are broken down into packets, routed individually over the network, and then reassembled at their destination.

To appreciate how the Internet, Web browsers and Web servers 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 looks at the step-by-step mechanism by which a Web browser ‘talks’ to a Web server according to the HyperText Transfer Protocol (HTTP).
Suggested answers to self-tests

Self-test 1.1

1 In the client-server model, an end-user relies on a program (the client) to communicate with an application residing on a remote machine (the server), in order to retrieve the requested resource. The Web follows this model, splitting the system’s functions between two tiers — the client and the server.

From the client’s point of view, all that’s needed is a Web browser and an Internet connection in order to get on the Web. From the server’s point of view, what’s needed is a machine that’s connected to the Internet, runs Web server software and hosts the required documents.

2 The Web uses the Internet as its underlying network medium. Data exchanged between Web clients and servers travel over the Internet. However, the Internet and the Web are not one and the same. There are many other services aside from the Web which run over the Internet.

Self-test 1.2

1 Three new technologies were created to build the Web:
   
   • The HyperText Markup Language (HTML) defines how webpages and hypertext links are written.
   
   • Universal Resource Locators (URLs) define the Web’s system for addressing and locating documents. Hypertext links contain URLs.
   
   • The HyperText Transfer Protocol (HTTP) is the communication language or protocol between the Web client (the Web browser) and the Web server. It describes how clients make requests for information and how servers respond to them.

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. For example, http://www.lycos.com is the URL for the Lycos search engine and 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 an HTML document:

\[
\text{<IMG SRC = "picture.gif">}
\]

6 These HTML tags should always be in an HTML document:

\[
\text{<HTML>}</HTML>
\]

\[
\text{<HEAD><TITLE>}</TITLE>}</HEAD>
\]

\[
\text{<BODY>}</BODY>
\]

**Self-test 1.3**

1 There are two design characteristics that can help the Internet withstand catastrophic attacks in wartime. First, it is decentralized. Computers can communicate directly with each other without having to go through a central node. This ensures that communication can still take place even if some machines are destroyed. Second, the Internet is a packet-switching network which offers multiple, redundant routes between two endpoints. This ensures that packets can still be routed between two hosts even if certain sections of the network are rendered inoperable.

2 The Internet is a global network that is made up of many smaller networks. These smaller networks connect with each other to form bigger and higher-level networks. There is no overall controlling network. Instead, these higher-level networks connect to each other through Network Access Points (NAPs). The Internet is a collection of huge networks which implement the same protocols and agree to route data traffic to each other at these Network Access Points.

**Self-test 1.4**

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.

2 These are a few Internet services that use TCP/IP:

- 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.

- File Transfer Protocol (FTP) is an Internet service which allows users to copy files from one computer to another across the Internet.
• Telnet is an Internet service which enables a person to set up a connection, log in, and conduct an interactive session with a remote computer on the Internet.

• 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.

5 Packets can be lost or destroyed when network hardware fails, or when networks become too congested with traffic. Even when packets make it to their destination, they may be delivered out of order, after a long delay, or with duplicate copies. This is why a reliable delivery service such as TCP is needed on top of an unreliable, connectionless packet delivery service such as IP.

Self-test 1.5

1 The browser extracts the server name and the file name (including the path) from the URL.

2 The browser asks a Domain Name Server to translate the server name into a corresponding IP address. If the URL includes the numeric IP address instead of the server name, this step is not needed.

3 The browser establishes a connection to the server at its IP address on port 80.

4 The browser sends an HTTP request to the server. The request includes the path name and file name of the requested resource.

5 The server sends the HTML text for the webpage to the browser within an HTTP response message.

6 The browser receives the HTTP response over the network. It interprets the HTML tags and displays the page on the screen.
References


