KARNATAKA STATE OPEN UNIVERSITY

MUKTHAGANGOTRI, MYSORE- 570 006

DEPARTMENT OF STUDIES IN INFORMATION TECHNOLOGY

M.Sc IN INFORMATION TECHNOLOGY

IV SEMESTER

WEB TECHNOLOGIES

MSIT- 119

MODULE: 1-4

MSIT-119:
Web Technologies

Course Design and Editorial Committee

Prof. M.G.Krishnan
Vice Chancellor
Karnataka State Open University
Mukthagangotri, Mysore – 570 006

Prof. Vikram Raj Urs
Dean (Academic) & Convener
Karnataka State Open University
Mukthagangotri, Mysore – 570 006

Head of the Department and Course Co-Ordinator

Rashmi B.S
Assistant Professor & Chairperson
DoS in Information Technology
Karnataka State Open University
Mukthagangotri, Mysore – 570 006

Course Editor

Ms. Nandini H.M
Assistant professor of Information Technology
DoS in Information Technology
Karnataka State Open University
Mukthagangotri, Mysore – 570 006

Course Writers

Dr. Vinay
Assistant Professor,
PG Department of Computer Science,
JSS College of Arts, Commerce & Science,
Ooty road, Mysore

Dr. Chethan H K
Associate Professor,
Dept of Computer Science,
Maharaja Institue of Technology,
Mysore.

Publisher

Registrar
Karnataka State Open University
Mukthagangotri, Mysore – 570 006

Developed by Academic Section, KSOU, Mysore
Karnataka State Open University, 2014

Printed and Published on behalf of Karnataka State Open University, Mysore-6 by the **Registrar (Administration)**

Karnataka State Open University

Muktagangothri, Mysore – 570 006

Master of Science in Information Technology

MSIT – 119 Web Technologies

Module 1

Module 2

Module 3

Module 4

Preface

There are many web technologies simple to complex and explaining them in detail is a primary objective of this study material. Understanding web technologies will help one to develop their own web sites. This book provides brief definitions of the major Web technologies along with reference to the external links for advance studies.

Overall structure of the study materials is organized into four modules. Each module consist of four units. Every modules is designed in such a way that it introduces one technology and discusses the merits and demerits in comparison with existing problems.

In brief, module1 discusses fundamentals of web, web browser, web servers and markup languages such as HTML and XHTML. Module 2 covers Introduces different levels of style sheets, style specification formats, selector forms and property value forms. In this module we also cover the some advance usage of tags in demonstration conflict resolution examples. In module 3 we introduce and explore one of the popular web technology called java scripts. This covers very basic programming to advanced concepts like pattern matching and expression evaluation. In the last unit of this module we have brief about XML technology and cascade style sheets. In the last module, we introduce another web technology program PERL. Perl is regarded as one of the one powerful and widely used web technology language for scripting World Wide Web. This module also covers CGI, servlets and java server pages. Some advance algorithms for pattern matching problems are also discussed. In the materials we have provided sufficient programming example to clearly demonstrate the work flow of the technologies. In the reference section we have given an external links for the readers to get addition resources on the topic.

Wish you all happy reading.

Module-1

UNIT 1:

Structure:

1.0 OBJECTIVES

At the end of this unit you will be able to know:

- Understand the fundamental of web and Internet
- Explanation of World Wide Web
- Web Browsers
- Web Servers
- Understand the working of Uniform Resource Locator

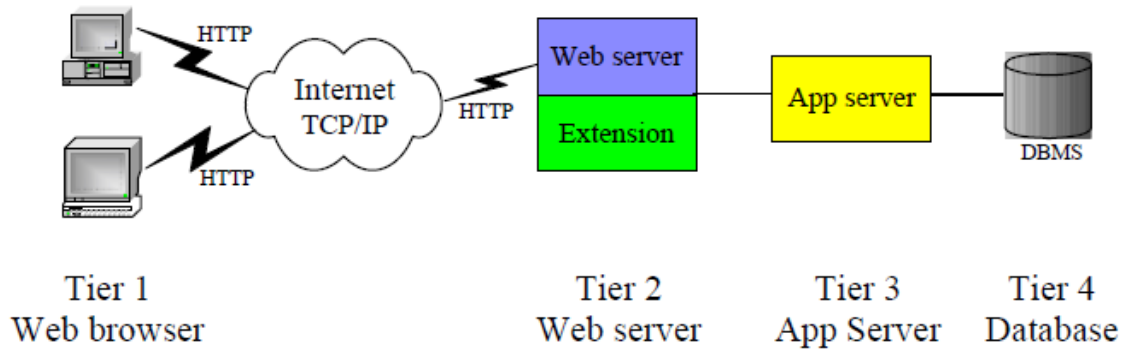
1.1 INTRODUCTION

Web servers and web browsers are communicating client-server computer programs for distributing documents and information, generally called web data, over the Internet. Web data are marked up in the HTML language for presentation and interaction with people in web browsers. Each web server uses an IP address or domain name as well as a port number for its identification. People use web browsers to send data requests to web servers with the HTTP protocol, and the web servers running on server computers either retrieve the requested data from local disks or generate the data on-the-fly, mark up the data in HTML, and send the resulting HTML files back to the web browsers to render. *Apache*, *Tomcat* and *IIS* are popular web server programs, and *IE* and *Firefox* are popular web browsers.

1.2 FUNDAMENTAL OF WEB

1.2.1 WEB ARCHITECTURE

A typical web application involves four tiers as depicted in the following web architecture figure: web browsers on the client side for rendering data presentation coded in HTML, a web server program that generates data presentation, an application server program that computes business logic, and a database server program that provides data persistency. The three types of server programs may run on the same or different server machines.



Web browsers can run on most operating systems with limited hardware or software requirement. They are the graphic user interface for the clients to interact with web applications. The basic functions of a web browser include:

- Interpret HTML markup and present documents visually;

- Support hyperlinks in HTML documents so the clicking on such a hyperlink can lead to the corresponding HTML file being downloaded from the same or another web server and presented;
- Use HTML form and the HTTP protocol to send requests and data to web applications and download HTML documents;
- Maintain cookies (name value pairs, explained later) deposited on client computers by a web application and send all cookies back to a web site if they are deposited by the web application at that web site (cookies will be further discussed later in this chapter);
- Use plug-in applications to support extra functions like playing audio-video files and running Java applets;
- Implement a *web browser sandbox* security policy: any software component (applets, JavaScript, ActiveX …) running inside a web browser normally cannot access local clients' resources like files or keyboards, and can only communicate directly with applications on the web server from where it is downloaded.

The web server is mainly for receiving document requests and data submission from web browsers through the HTTP protocol on top of the Internet's TCP/IP layer. The main function of the web server is to feed HTML files to the web browsers. If the client is requesting a static existing file, it will be retrieved on a server hard disk and sent back to the web browser right away. If the client needs customized HTML pages like the client's bank statement, a software component, like a JSP page or a servlet class (the

"Extension" box in the web architecture figure), needs to retrieve the client's data from the database and compose a response HTML file on-the-fly.

The application server is responsible for computing the business logics of the web application, like carrying out a bank account fund transfer and computing the shortest route to drive from one city to another. If the business logic is simple or the web application is only used by a small group of clients, the application server is usually missing and business logics are computed in the web server extensions (PHP, JSP or servlet …). But for a popular web application that generates significant computation load for serving each client, the application server will take advantage of a separate hardware server machine to run business logics more efficiently. This is a good application of the divide-and-conquer problem solving methodology.

1.2.2 UNIFORM RESOURCE LOCATORS (URL)

A web server program runs multiple web applications (sites) hosted in different folders under the web server program's document root folder. A server computer may run multiple server programs including web servers. Each server program on a server computer uses a port number, between 0 and 65535, unique on the server machine as its local identification (by default a web server uses port 80). Each server computer has an IP address, like 198.105.44.27, as its unique identifier on the Internet. Domain names, like www.pace.edu, are used as user-friendly identifications of server computers, and they are mapped to

IP addresses by a Domain Name Server (DNS). A Uniform Resource Locator (URL) is an address for uniquely identifying a web resource (like a web page or a Java object) on the Internet, and it has the following general format:

http://domain-name:port/application/resource?query-string

where *http* is the protocol for accessing the resource (*https* and *ftp* are popular alternative protocols standing for *secure HTTP* and *File Transfer Protocol*); *application* is a server-side folder containing all resources related to a web application; *resource* could be the name (alias or nickname) of an HTML or script/program file residing on a server hard disk; and the optional query string passes user data to the web server. An example URL is http://www.amazon.com/computer/sale?model=dell610.

There is a special domain name "localhost" that is normally defined as an alias of local IP address 127.0.0.1. Domain name "localhost" and IP address 127.0.0.1 are for addressing a local computer, very useful for testing web applications where the web browser and the web server are running on the same computer.

Most computers are on the Internet as well as on a local area network (LAN), like home wireless network, and they have an external IP address and a local IP address. To find out what is your computer's external IP address on the Internet, use a web browser to visit http://whatismyip.com. To find out what is your local (home) IP address, on Windows, run "ipconfig" in a DOS window; and on Linux, run "sudo ifconfig" in a terminal window.

1.2.3 HTML BASICS

HTML is a markup language. An HTML document is basically a text document marked up with instructions as to document logical structure and document presentation. The following is the contents of file "~/tomcat/webapps/demo/echoPost.html" in the ubuntu10 VM.

```
<html>
<head>
<body>
    <form method="post" action="http://localhost:8080/demo/echo">
        Enter your name: <input type="text" name="user"/> <br/><br/>
        <input type="submit" value="Submit"/>
        <input type="reset" value="Reset"/>
    </form>
</body>
</head>
</html>
```

An HTML *tag name* is a predefined keyword, like html, body, head, title, p, and b, all in lower-case.

A tag name is used in the form of a *start tag* or an *end tag*. A start tag is a tag name enclosed in angle brackets < and >, like <html> and . An end tag is the same as the corresponding start tag except it has a forward slash / immediately before the tag name, like </html> and .

An *element* consists of a start tag and a matching end tag based on the same tag name, with optional text or other elements, called *element value*, in between them. The following are some element examples:

```
<p>This is free text</p>
<p>This element has a nested <b>element</b></p>
```

While the elements can be nested, they cannot be partially nested: the end tag of an element must come after the end tags of all of its nested elements (*first starting last ending*). The following example is not a valid element because it violates the above rule:

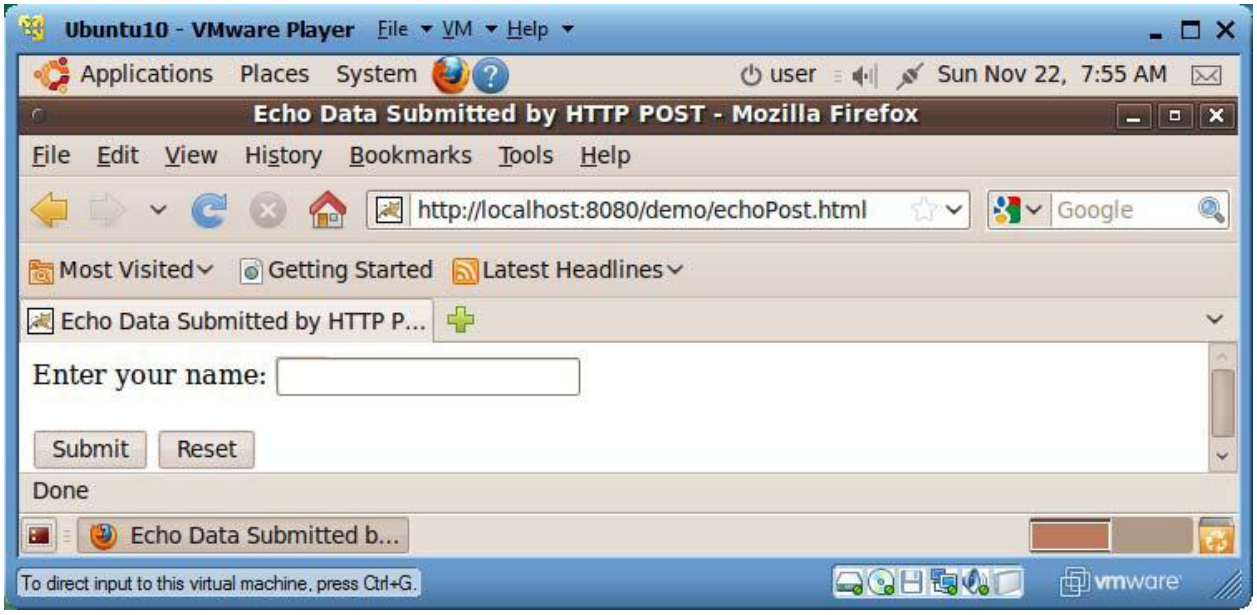
```
<p>This is not a valid <bold>element<p><bold>
```

The *newline* character, the *tab* character and the *space* character are collectively called the *white-space characters*. A sequence of white-space characters acts like a single space for web browser's data presentation. Therefore, in normal situations, HTML document's formatting is not important (it will not change its presentation in web browsers) as long as you don't remove all white-space characters between successive words.

If an element contains no value, the start tag and the end tag can be combined into a single one as <tagName/>. As an example, we use
 to insert a line break in HTML documents.

The start tag of an element may contain one or more *attributes*, each in the form "attributeName="attributeValue"". The above form element has two attributes: method and action.

An HTML document must contain exactly one top-level html element, which in turn contains exactly one body element. Most of the other contents are nested in the body element. If you load the above file "echoPost.html" in a web browser you will see the following:

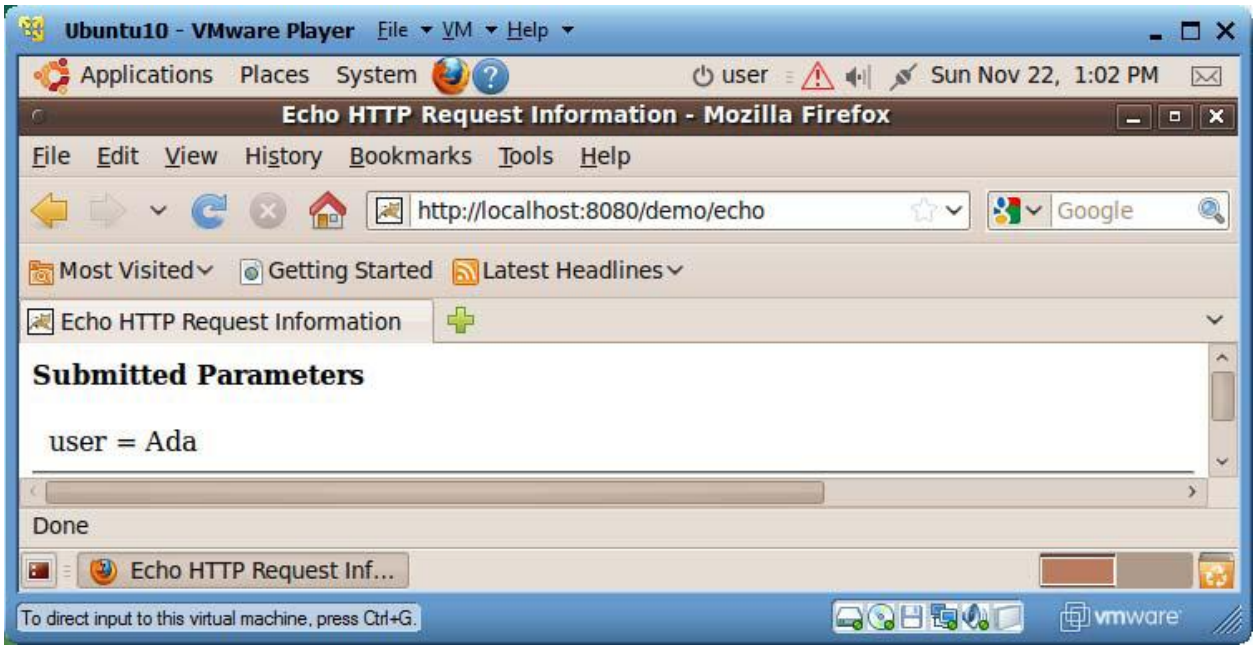


The form element is the most important mechanism for interaction between people and web applications.

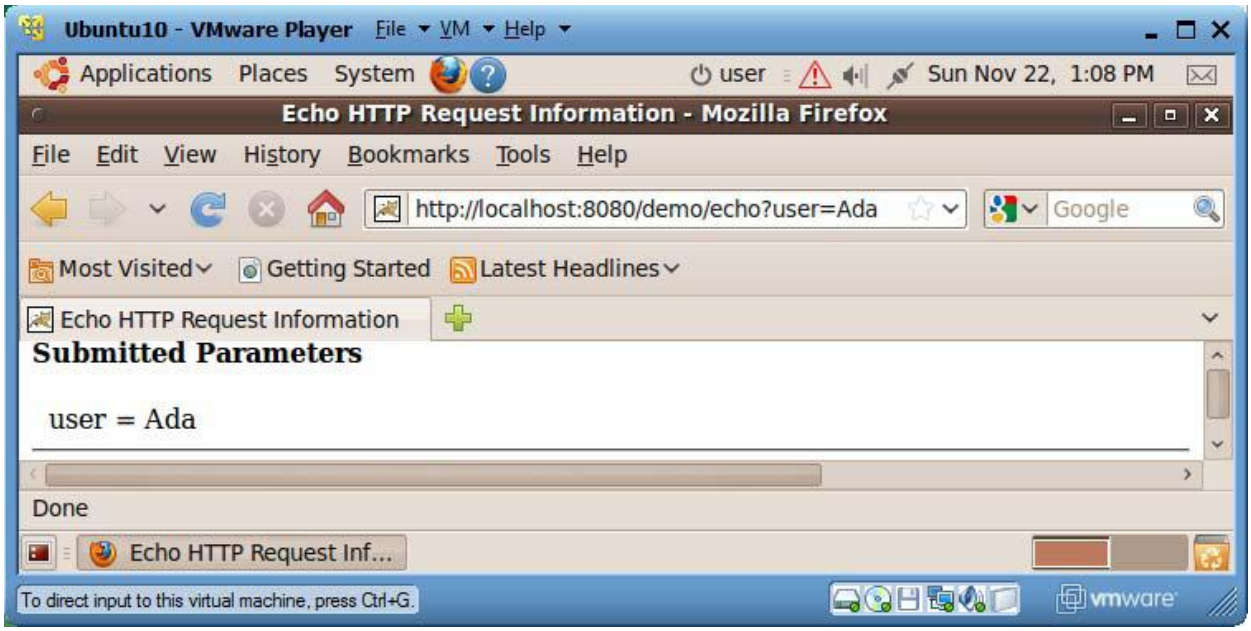
A form typically contains a few input elements and at least one submit button. A form element usually has two attributes: the method attribute for specifying HTTP method for submitting the form data to the web application (only values normally used are "get" and "post"); and the action attribute for specifying the form data submission destination, or the URL of a web application. In this example, when people click on the submit button, the form data will be sent to resource "echo" of the same web application "demo" deployed on your Ubuntu VM's Tomcat web server, which will echo back all information the web browser sent to the web server. If the action value doesn't specify the domain name/IP address or the web application, then the

web application from where this HTML file came from will receive the form data. The first input element of type "text" has been rendered as a text field, the second input element of type "submit" has been rendered as a submit button, and the third input element of type "reset" has been rendered as a reset button. The value attribute of the input elements determines what string will be displayed on the element's image. The name attribute of the input element specifies the variable name with which web server programs can access what people type/enter in the element. When the submit button is clicked, the form data will be packaged as an HTTP request and sent to the web resource specified by the action attribute with the method specified by the method attribute.

If you type "Ada" in the name filed and click on the submit button, you will receive the HTTP response partially displayed below.



If you load file "echoGet.html" from the same web application folder "demo", the HTML file contents is basically the same except the method attribute for the form is changed from "post" to "get". If you enter "Ada" in the name field and click on the submit button again, you will notice that the query string "?user=Ada" has been appended to the end of the URL. This is a major difference from HTTP POST method, and you will learn more about HTTP GET/POST soon.



An HTML file can contain hyperlinks to other web pages so users can click on them to visit different web pages. A hyperlink has the general structure of Hyperlink Text. The following is an example hyperlink. Since its href value is not a web page, the *welcome page* of the Google web site, which is the default page sent back if a browser visits the web site without specifying a specific interested page, will be sent back to the web browser.

Google

When you click on a hyperlink, an HTTP GET request will be sent to the web server with all values to be submitted in the form of query strings.

1.2.4 HTTP PROTOCOL

Web browsers interact with web servers with a simple application-level protocol called HTTP (HyperText Transfer Protocol), which runs on top of TCP/IP network connections. When people click on the submit button of an HTML form or a hyperlink in a web browser, a TCP/IP virtual communication channel is created from the browser to the web server specified in the URL; an HTTP GET or POST request is sent through this channel to the destination web application, which retrieves data submitted by the browser user and composes an HTML file; the HTML file is sent back to the web browser as an HTTP response through the same TCP/IP channel; and then the TCP/IP channel is shut down.

The following is the HTTP POST request sent when you type “Ada” in the text field and click on the submit button of the previous file “echoPost.html”.

POST /demo/echo HTTP/1.1

Accept: text/html

Accept: audio/x

User-agent: Mozilla/5.0

Referer: http://localhost:8080/demo/echoPost.html

Content-length: 8 user=Ada

The first line, the request line, of a HTTP request is used to specify the submission type, GET or POST; the specific web resource on the web server for receiving and processing the submitted data; and the latest HTTP version that the web browser supports. As of 2010, version 1.1 is the latest HTTP specification.

The following lines, up to before the blank line, are HTTP *header lines* for declaring web browser capabilities and extra information for this submission, each of form "name: value". The first two Accept headers declare that the web browser can process HTML files and any standard audio file formats from the web server. The User-agent header declares the software architecture of the web browser. The

Referer (yes this misspelled word is used by the HTTP standard) header specifies the URL of a web page from which this HTTP request is generated (this is how online companies like Amazon and Yahoo collect money for advertisements on their web pages from their sponsors). Any text after the blank line below the header lines is called the *entity body* of the HTTP request, which contains user data submitted through HTTP POST. The Content-length header specifies the exact number of bytes that the entity body contains. If the data is submitted through HTTP GET, the entity body will be empty and the data go to the query string of the submitting URL, as you saw earlier.

In response to this HTTP POST request, the web server will forward the submitted data to resource echo of web application demo, and the resource echo (a Java servlet) will generate dynamically an HTML page for most data it can get from the submission and let the web server send the HTML page back to the web browser as the entity body of the following HTTP response.

HTTP/1.1 200 OK

Server: NCSA/1.3

Mime_version: 1.0

Content_type: text/html

Content_length: 2000

<HTML>

……

</HTML>

The first line, the response line, of an HTTP response specifies the latest HTTP version that the web server supports. The first line also provides a web server processing status code, the popular values of which include 200 for OK, 400 if the server doesn't understand the request, 404 if the server cannot find the requested page, and 500 for server internal error. The third entry on the first line is a brief message explaining the status code. The first two header lines declare the web server capabilities and meta-data for the returned data. In this example, the web server is based on a software architecture named "NCSA/1.3", and it supports *Multipurpose Internet Mail Extension* (MIME) specification v1.0 for web browsers to submit text or binary data with multi-parts. The last two header lines declare that the entity body contains HTML data with exactly 2000 bytes. The web browser will parse this HTTP response and present the response data.

The HTTP protocol doesn't have memory: the successive HTTP requests don't share data.

HTTP GET was initially designed for downloading static web pages from web servers, and it mainly used short query strings to specify the web page search criteria. HTTP POST was initially designed for submitting data to web servers, so it used the request entity body to send data to the web servers as a data stream, and its response normally depended on the submitted data and the submission status. While both HTTP GET and HTTP POST can send user requests to web servers and retrieve HTML pages from web servers for a web browser to present, they have the following subtle but important differences:

- HTTP GET sends data as query strings so people can read the submitted data over submitter's shoulders.
- Web servers have limited buffer size, typically 512 bytes, for accommodating query string data. If a user submits more data than that limit, either the data would be truncated, or the web server would crash, or the submitted data could potentially overwrite some computer code on the server and the server was led to run some hideous code hidden as part of the query string data. The last case is the so-called *buffer overflow*, a common way for hackers to take over the control of a server and spread virus or worms.

- By default web browsers keep (cache) a copy of the web page returned by an HTTP GET request so the future requests to the same URL can be avoided and the cached copy could be easily reused. While this can definitely improve the performance if the requested web page doesn't change, it could be disastrous if the web page or data change with time.

1.2.5 SESSION DATA MANAGEMENT

Most web applications need a user to interact with it multiple times to complete a business transaction.

For example, when you shop at Amazon, you choose one book at a time by clicking on some HTML form's submission buttons/hyperlinks in a web browser, and Amazon will process your submitted data and send you another HTML form for further shopping. A sequence of related HTTP requests between a web browser and a web application for accomplishing a single business transaction is called a session. All data specified by the user is called the session data. Session data are private so they must be protected from other users. A session normally starts when you first visit a web site in a particular day, and terminates when you pay off your purchase or shut down your web browser. Since the HTTP protocol has no memory, web applications have to use some special mechanisms to securely maintain the user session data.

Cookies

A cookie is a pair of name and value, as in (name, value). A web application can generate multiple cookies, set their life spans in terms of how many milliseconds each of them should be alive, and send them back to a web browser as part of an HTTP response. If cookies are allowed, a web browser will save all cookies on its hosting computer, along with their originating URLs and life spans. When an HTTP request is sent from a web browser of the same type on the same computer to a web site, all live cookies originated from that web site will be sent to the web site as part of the HTTP request. Therefore session data can be stored in cookies. This is the simplest approach to maintain session data. Since the web server doesn't need to commit any resources for the session data, this is the most scalable approach to support session data of large number of users. But it is not secure or efficient for cookies to go between a web browser and a web site for every HTTP request, and hackers could eavesdrop for the session data along the Internet path.

Hidden Fields

Some web users have great concern of the cookie's security implications and they disable cookie support on their web browsers. A web application can check the header fields of HTTP

requests to detect whether cookies are supported by the requesting web browser. If the cookies are disabled, the web application will normally use form hidden fields to store session data. Upon receiving submitted data through an HTTP request, the web application will generate a new HTML form for the user to continue the business transaction, and it will populate all useful session data in the new HTML form as hidden fields (input elements of type "hidden"). When the user submits the form again, all the data that the user just entered the form, as well as all data saved in the form as hidden fields, will be sent back to the web application again. Therefore this hidden fields approach for maintaining session data shares most of the advantages and disadvantages of the cookie approach.

Query Strings

Sometimes query strings can also be used to maintain small amount of session data. This is particular true for maintaining the short session IDs that will be introduced below. But since most business transactions are implemented with HTML forms, this approach is less useful.

Server-Side Session Objects

For improving the security of session data and avoiding the wasted network bandwidth for session data to move back and forth between a web browser and a web server, you can also save much of the session data on the web server as server-side *session objects*. A session object has a unique session ID for identifying a specific user. A session object is normally implemented as a hash table (lookup table) consisting of (name, value) pairs. A single cookie, hidden field of a form, or query string of a hyperlink can be used to maintain the session ID. Since session ID is a fixed size small piece of data, it will not cause much network overhead for going between a web browser and a web server for each HTTP request. For securing the session data, you need to make sure that the session ID is unique and properly protected on the client site. Since this approach stores all session data on the web server, it takes the most server resources and is relatively harder to serve large number of clients concurrently.

1.3 INTERNET

The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to serve several billion users worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless, and

optical networking technologies. The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents of the World Wide Web (WWW), the infrastructure to support email, and peer-to-peer networks.

Most traditional communications media including telephone, music, film, and television are being reshaped or redefined by the Internet, giving birth to new services such as voice over Internet Protocol (VoIP) and Internet Protocol television (IPTV). Newspaper, book, and other print publishing are adapting to website technology, or are reshaped into blogging and web feeds. The Internet has enabled and accelerated new forms of human interactions through instant messaging, Internet forums, and social networking. Online shopping has boomed both for major retail outlets and small artisans and traders. Business-to-business and financial services on the Internet affect supply chains across entire industries.

1.3.1 TECHNOLOGY

Protocols

The communications infrastructure of the Internet consists of its hardware components and a system of software layers that control various aspects of the architecture. While the hardware can often be used to support other software systems, it is the design and the rigorous standardization process of the software architecture that characterizes the Internet and provides the foundation for its scalability and success. The responsibility for the architectural design of the Internet software systems has been delegated to the Internet Engineering Task Force (IETF). The IETF conducts standard-setting work groups, open to any individual, about the various aspects of Internet architecture. Resulting discussions and final standards are published in a series of publications; each called a Request for Comments (RFC), freely available on the IETF web site.

The principal methods of networking that enable the Internet are contained in specially designated RFCs that constitute the Internet Standards. Other less rigorous documents are simply informative, experimental, or historical, or document the best current practices (BCP) when implementing Internet technologies.

The Internet standards describe a framework known as the Internet protocol suite. This is a model architecture that divides methods into a layered system of protocols. The layers correspond to the environment or scope in which their services operate. At the top is

the application layer, the space for the application-specific networking methods used in software applications, e.g., a web browser program uses the client-server application model and many file-sharing systems use a peer-to-peer paradigm. Below this top layer, the transport layer connects applications on *different hosts* via the network with appropriate data exchange methods. Underlying these layers are the core networking technologies, consisting of two layers.

The internet layer enables computers to identify and locate each other via Internet Protocol (IP) addresses, and allows them to connect to one another via intermediate (transit) networks. Last, at the bottom of the architecture, is a software layer, the link layer, that provides connectivity between hosts on the same local network link, such as a local area network (LAN) or a dial-up connection. The model, also known as TCP/IP, is designed to be independent of the underlying hardware, which the model therefore does not concern itself with in any detail. Other models have been developed, such as the Open Systems Interconnection (OSI) model, but they are not compatible in the details of description or implementation; many similarities exist and the TCP/IP protocols are usually included in the discussion of OSI networking.

The most prominent component of the Internet model is the Internet Protocol (IP), which provides addressing systems (IP addresses) for computers on the Internet. IP enables internetworking and in essence establishes the Internet itself. IP Version 4 (IPv4) is the initial version used on the first generation of today's Internet and is still in dominant use. It was designed to address up to ~4.3 billion (10^9) Internet hosts. However, the explosive growth of the Internet has led to IPv4 address exhaustion, which entered its final stage in 2011, when the global address allocation pool was exhausted. A new protocol version, IPv6, was developed in the mid-1990s, which provides vastly larger addressing capabilities and more efficient routing of Internet traffic. IPv6 is currently in growing deployment around the world, since Internet address registries (RIRs) began to urge all resource managers to plan rapid adoption and conversion.

IPv6 is not interoperable with IPv4. In essence, it establishes a parallel version of the Internet not directly accessible with IPv4 software. This means software upgrades or translator facilities are necessary for networking devices that need to communicate on both networks. Most modern computer operating systems already support both versions of the Internet Protocol. Network infrastructures, however, are still lagging in this development. Aside from the complex array of physical connections that make up its infrastructure, the Internet is facilitated by bi- or

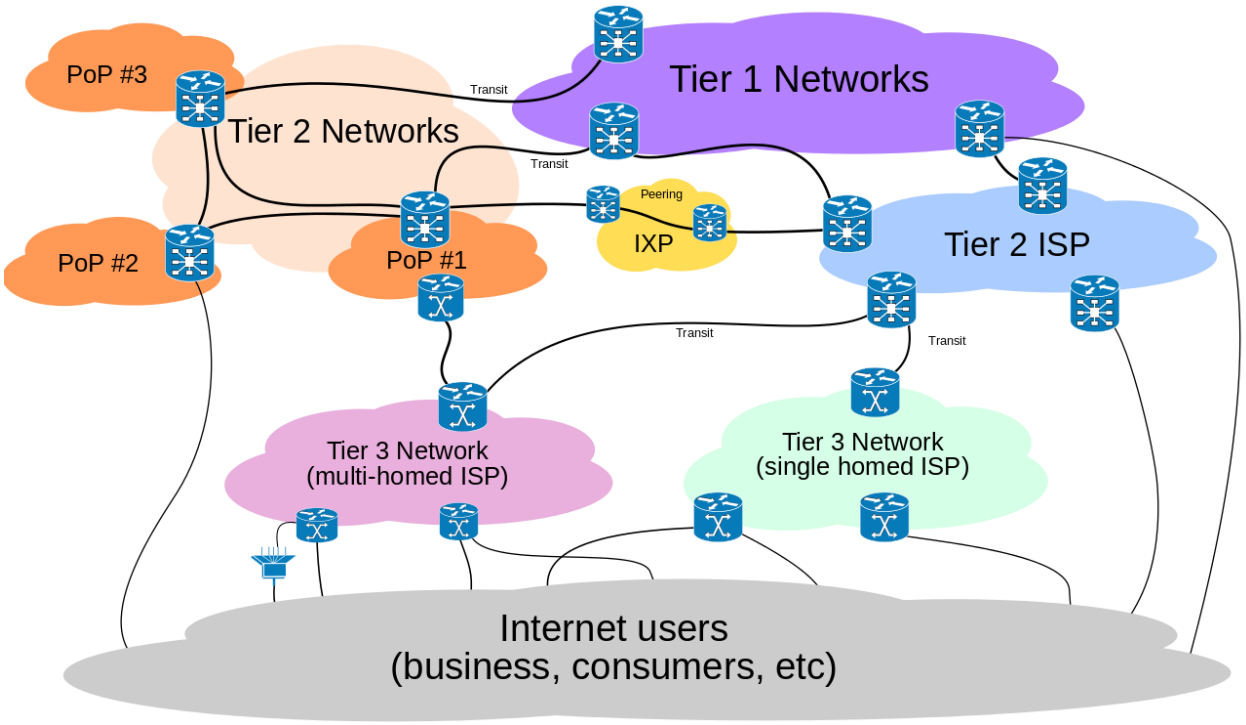
multi-lateral commercial contracts (e.g., peering agreements), and by technical specifications or protocols that describe how to exchange data over the network. Indeed, the Internet is defined by its interconnections and routing policies.

Routing

Internet service providers connect customers, which represent the bottom of the routing hierarchy, to customers of other ISPs via other higher or same-tier networks. At the top of the routing hierarchy are the Tier 1 networks, large telecommunication companies which exchange traffic directly with all other Tier 1 networks via peering agreements. Tier 2 networks buy Internet transit from other providers to reach at least some parties on the global Internet, though they may also engage in peering. An ISP may use a single upstream provider for connectivity, or implement multihoming to achieve redundancy. Internet exchange points are major traffic exchanges with physical connections to multiple ISPs.

Computers and routers use routing tables to direct IP packets to the next-hop router or destination. Routing tables are maintained by manual configuration or by routing protocols. End-nodes typically use a default route that points toward an ISP providing transit, while ISP routers use the Border Gateway Protocol to establish the most efficient routing across the complex connections of the global Internet.

Large organizations, such as academic institutions, large enterprises, and governments, may perform the same function as ISPs, engaging in peering and purchasing transit on behalf of their internal networks. Research networks tend to interconnect into large sub networks such as GEANT, GLORIAD, Internet2, and the UK's national research and education network, JANET.



1.3.2 USES OF INTERNET

The Internet allows greater flexibility in working hours and location, especially with the spread of unmetered high-speed connections. The Internet can be accessed almost anywhere by numerous means, including through mobile Internet devices. Mobile phones, datacards, handheld game consoles and cellular routers allow users to connect to the Internet wirelessly. Within the limitations imposed by small screens and other limited facilities of such pocket-sized devices, the services of the Internet, including email and the web, may be available. Service providers may restrict the services offered and mobile data charges may be significantly higher than other access methods.

Educational material at all levels from pre-school to post-doctoral is available from websites. Examples range from CBeebies, through school and high-school revision guides and virtual universities, to access to top-end scholarly literature through the likes of Google Scholar. For distance education, help with homework and other assignments, self-guided learning, whiling away spare time, or just looking up more detail on an interesting fact, it has never been easier for people to access educational information at any level from anywhere. The

Internet in general and the World in particular are important enablers of both formal and informal education.

The low cost and nearly instantaneous sharing of ideas, knowledge, and skills has made collaborative work dramatically easier, with the help of collaborative software. Not only can a group cheaply communicate and share ideas but the wide reach of the Internet allows such groups more easily to form. An example of this is the free software movement, which has produced, among other things, Linux, Mozilla Firefox, and OpenOffice.org. Internet chat, whether using an IRC chat room, an instant messaging system, or a social networking website, allows colleagues to stay in touch in a very convenient way while working at their computers during the day. Messages can be exchanged even more quickly and conveniently than via email. These systems may allow files to be exchanged, drawings and images to be shared, or voice and video contact between team members.

Content management systems allow collaborating teams to work on shared sets of documents simultaneously without accidentally destroying each other's work. Business and project teams can share calendars as well as documents and other information. Such collaboration occurs in a wide variety of areas including scientific research, software development, conference planning, political activism and creative writing. Social and political collaboration is also becoming more widespread as both Internet access and computer literacy spread.

The Internet allows computer users to remotely access other computers and information stores easily, wherever they may be. They may do this with or without computer security, i.e. authentication and encryption technologies, depending on the requirements. This is encouraging new ways of working from home, collaboration and information sharing in many industries. An accountant sitting at home can audit the books of a company based in another country, on a server situated in a third country that is remotely maintained by IT specialists in a fourth. These accounts could have been created by home-working bookkeepers, in other remote locations, based on information emailed to them from offices all over the world. Some of these things were possible before the widespread use of the Internet, but the cost of private leased lines would have made many of them infeasible in practice. An office worker away from their desk, perhaps on the other side of the world on a business trip or a holiday, can access their emails, access their data

using cloud computing, or open a remote desktop session into their office PC using a secure Virtual (VPN) connection on the Internet. This can give the worker complete access to all of their normal files and data, including email and other applications, while away from the office. It has been referred to among system administrators as the Virtual Private Nightmare, because it extends the secure perimeter of a corporate network into remote locations and its employees' homes.

Many people use the terms *Internet* and *World Wide Web*, or just the *Web*, interchangeably, but the two terms are not synonymous. The World Wide Web is only one of hundreds of services used on the Internet. The Web is a global set of documents, images and other resources, logically interrelated by hyperlinks and referenced with Uniform Resource Identifiers (URIs). URIs symbolically identifies services, servers, and other databases, and the documents and resources that they can provide. Hypertext Transfer Protocol (HTTP) is the main access protocol of the World Wide Web. services also use HTTP to allow software systems to communicate in order to share and exchange business logic and data.

World Wide Web browser software, such as Microsoft's Internet Explorer, Mozilla Firefox, Opera, Apple's Safari, and Google Chrome, lets users navigate from one web page to another via hyperlinks embedded in the documents. These documents may also contain any combination of computer data, including graphics, sounds, text, video, multimedia and interactive content that runs while the user is interacting with the page. Client-side software can include animations, games, office applications and scientific demonstrations. Through keyword-driven Internet research using search engines like Yahoo! and Google, users worldwide have easy, instant access to a vast and diverse amount of online information. Compared to printed media, books, encyclopedias and traditional libraries, the World Wide Web has enabled the decentralization of information on a large scale.

The Web has also enabled individuals and organizations to publish ideas and information to a potentially large audience online at greatly reduced expense and time delay. Publishing a web page, a blog, or building a website involves little initial cost and many cost-free services are available. Publishing and maintaining large, professional web sites with attractive, diverse and up-to-date information is still a difficult and expensive proposition, however. Many individuals and some companies and groups use *web logs* or blogs, which are largely used as easily

updatable online diaries. Some commercial organizations encourage staff to communicate advice in their areas of specialization in the hope that visitors will be impressed by the expert knowledge and free information, and be attracted to the corporation as a result.

One example of this practice is Microsoft, whose product developers publish their personal blogs in order to pique the public's interest in their work. Collections of personal web pages published by large service providers remain popular, and have become increasingly sophisticated. Whereas operations such as Angelfire and GeoCities have existed since the early days of the Web, newer offerings from, for example, Facebook and Twitter currently have large followings. These operations often brand themselves as social network services rather than simply as web page hosts.

Advertising on popular web pages can be lucrative, and e-commerce or the sale of products and services directly via the Web continues to grow.

Communication

Email is an important communications service available on the Internet. The concept of sending electronic text messages between parties in a way analogous to mailing letters or memos predates the creation of the Internet. Pictures, documents and other files are sent as email attachments. Emails can be cc-ed to multiple email addresses.

Internet telephony is another common communications service made possible by the creation of the Internet. VoIP stands for Voice-over-Internet Protocol, referring to the protocol that underlies all Internet communication. The idea began in the early 1990s with walkie-talkie-like voice applications for personal computers. In recent years many VoIP systems have become as easy to use and as convenient as a normal telephone. The benefit is that, as the Internet carries the voice traffic, VoIP can be free or cost much less than a traditional telephone call, especially over long distances and especially for those with always-on Internet connections such as cable or ADSL. VoIP is maturing into a competitive alternative to traditional telephone service. Interoperability between different providers has improved and the ability to call or receive a call from a traditional telephone is available. Simple, inexpensive VoIP network adapters are available that eliminate the need for a personal computer.

Voice quality can still vary from call to call, but is often equal to and can even exceed that of traditional calls. Remaining problems for VoIP include emergency telephone

number dialing and reliability. Currently, a few VoIP providers provide an emergency service, but it is not universally available. Older traditional phones with no "extra features" may be line-powered only and operate during a power failure; VoIP can never do so without a backup power source for the phone equipment and the Internet access devices. VoIP has also become increasingly popular for gaming applications, as a form of communication between players. Popular VoIP clients for gaming include Ventrilo and Teamspeak. Modern video game consoles also offer VoIP chat features.

Data transfer

File sharing is an example of transferring large amounts of data across the Internet. A computer file can be emailed to customers, colleagues and friends as an attachment. It can be uploaded to a website or FTP server for easy download by others. It can be put into a "shared location" or onto a file server for instant use by colleagues. The load of bulk downloads to many users can be eased by the use of "mirror" servers or peer-to-peer networks. In any of these cases, access to the file may be controlled by user authentication, the transit of the file over the Internet may be obscured by encryption, and money may change hands for access to the file. The price can be paid by the remote charging of funds from, for example, a credit card whose details are also passed – usually fully encrypted – across the Internet. The origin and authenticity of the file received may be checked by digital signatures or by MD5 or other message digests. These simple features of the Internet, over a worldwide basis, are changing the production, sale, and distribution of anything that can be reduced to a computer file for transmission. This includes all manner of print publications, software products, news, music, film, video, photography, graphics and the other arts. This in turn has caused seismic shifts in each of the existing industries that previously controlled the production and distribution of these products.

Streaming media is the real-time delivery of digital media for the immediate consumption or enjoyment by end users. Many radio and television broadcasters provide Internet feeds of their live audio and video productions. They may also allow time-shift viewing or listening such as Preview, Classic Clips and Listen Again features. These providers have been joined by a range of pure Internet "broadcasters" who never had on-air licenses. This means that an Internet-connected device, such as a computer or something more specific, can be used to access on-line media in much the same way as was previously possible only with a television or radio receiver. The range of available types of content is much wider, from specialized technical webcasts to on-

demand popular multimedia services. Podcasting is a variation on this theme, where – usually audio – material is downloaded and played back on a computer or shifted to a portable media player to be listened to on the move. These techniques using simple equipment allow anybody, with little censorship or licensing control, to broadcast audio-visual material worldwide.

Digital media streaming increases the demand for network bandwidth. For example, standard image quality needs 1 Mbit/s link speed for SD 480p, HD 720p quality requires 2.5 Mbit/s, and the top-of-the-line HDX quality needs 4.5 Mbit/s for 1080p.

Webcams are a low-cost extension of this phenomenon. While some webcams can give full-frame-rate video, the picture either is usually small or updates slowly. Internet users can watch animals around an African waterhole, ships in the Panama Canal, traffic at a local roundabout or monitor their own premises, live and in real time. Video chat rooms and video conferencing are also popular with many uses being found for personal webcams, with and without two-way sound. YouTube was founded on 15 February 2005 and is now the leading website for free streaming video with a vast number of users. It uses a flash-based web player to stream and show video files. Registered users may upload an unlimited amount of video and build their own personal profile. YouTube claims that its users watch hundreds of millions, and upload hundreds of thousands of videos daily.

1.4 WORLD WIDE WEB

The **World Wide Web** (abbreviated as **WWW** or **W3**, commonly known as **the web**) is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

1.4.1 FUNCTIONS

The terms Internet and World Wide Web are often used in everyday speech without much distinction. However, the Internet and the World Wide Web are not the same. The Internet is a global system of interconnected computer networks. In contrast, the web is one of the services that run on the Internet. It is a collection of text documents and other resources, linked by hyperlinks and URLs, usually accessed by web browsers from web servers. In short, the web can be thought of as an application "running" on the Internet.

Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser or by following a hyperlink to that page or resource. The web browser then initiates a series of communication messages, behind the scenes, in order to fetch and display it. The following example demonstrates how a web browser works. Consider accessing a page with the URL http://example.org/wiki/World_Wide_Web.

First, the browser resolves the server-name portion of the URL (*example.org*) into an Internet Protocol address using the globally distributed database known as the Domain Name System (DNS); this lookup returns an IP address such as *208.80.152.2*. The browser then requests the resource by sending an HTTP request across the Internet to the computer at that particular address. It makes the request to a particular application port in the underlying Internet Protocol Suite so that the computer receiving the request can distinguish an HTTP request from other network protocols it may be servicing such as e-mail delivery; the HTTP protocol normally uses port 80. The content of the HTTP request can be as simple as the two lines of text GET */wiki/World_Wide_Web*HTTP/1.1 Host: *example.org*

The computer receiving the HTTP request delivers it to web server software listening for requests on port 80. If the web server can fulfill the request it sends an HTTP response back to the browser indicating success, which can be as simple as HTTP/1.0 200 OK Content-Type: text/html; charset=UTF-8 followed by the content of the requested page. The Hypertext Markup Language for a basic web page looks like <html> <head> <title>Example.org – The World Wide Web</title> </head> <body> The World Wide Web, abbreviated as WWW and commonly known ... </body> </html>

The web browser parses the HTML, interpreting the markup (<title>, for paragraph, and such) that surrounds the words in order to draw the text on the screen.

Many web pages use HTML to reference the URLs of other resources such as images, other embedded media, scripts that affect page behavior, and Cascading Style Sheets that affect page layout. The browser will make additional HTTP requests to the web server for these other Internet media types. As it receives their content from the web server, the browser progressively renders the page onto the screen as specified by its HTML and these additional resources.

LINKING

Most web pages contain hyperlinks to other related pages and perhaps to downloadable files, source documents, definitions and other web resources. In the underlying HTML, a hyperlink looks like Example.org, a free encyclopedia

Such a collection of useful, related resources, interconnected via hypertext links is dubbed a web of information. Publication on the Internet created what Tim Berners-Lee first called the WorldWideWeb (in its original CamelCase, which was subsequently discarded) in November 1990.

The hyperlink structure of the WWW is described by the webgraph: the nodes of the webgraph correspond to the web pages (or URLs) the directed edges between them to the hyperlinks.

Over time, many web resources pointed to by hyperlinks disappear, relocate, or are replaced with different content. This makes hyperlinks obsolete, a phenomenon referred to in some circles as link rot and the hyperlinks affected by it are often called dead links. The ephemeral nature of the Web has prompted many efforts to archive web sites. The Internet Archive, active since 1996, is the best known of such efforts.

1.4.2 DYNAMIC UPDATES OF WEB PAGES

To make web pages more interactive, some web applications also use JavaScript techniques such as Ajax(asynchronous JavaScript and XML). Client-side script is delivered with the page that can make additional HTTP requests to the server, either in response to user actions such as mouse movements or clicks, or based on lapsed time. The server's responses are used to modify the current page rather than creating a new page with each response, so the server needs only to provide limited, incremental information. Multiple Ajax requests can be handled at the same time, and users can interact with the page while data is being retrieved. Web pages may also regularly poll the server to check whether new information is available.

1.4.3 WWW PREFIX

Many hostnames used for the World Wide Web begin with www because of the long-standing practice of naming Internet hosts (servers) according to the services they provide. The hostname for a web server is often www, in the same way that it may be ftp for an FTP server, and news or nntp for a USENET news server. These host names appear as Domain Name

System or (DNS) subdomain names, as in www.example.com. The use of 'www' as a sub-domain name is not required by any technical or policy standard and many web sites do not use it; indeed, the first ever web server was called nxoc01.cern.ch.

The use of a sub domain name is useful for load balancing incoming web traffic by creating a CNAME record that points to a cluster of web servers. Since, currently, only a sub domain can be used in a CNAME, the same result cannot be achieved by using the bare domain root.

When a user submits an incomplete domain name to a web browser in its address bar input field, some web browsers automatically try adding the prefix "www" to the beginning of it and possibly ".com", ".org" and ".net" at the end, depending on what might be missing. For example, entering 'microsoft' may be transformed to http://www.microsoft.com/ and 'openoffice' tohttp://www.openoffice.org. This feature started appearing in early versions of Mozilla Firefox, when it still had the working title 'Firebird' in early 2003, from an earlier practice in browsers such asLynx. It is reported that Microsoft was granted a US patent for the same idea in 2008, but only for mobile devices.

Use of the www prefix is declining as Web 2.0 web applications seek to brand their domain names and make them easily pronounceable. As the mobile web grows in popularity, services likeGmail.com, MySpace.com, Facebook.com and Twitter.com are most often discussed without adding www to the domain (or, indeed, the .com).

1.4.4 SCHEME SPECIFIERS: HTTP AND HTTPS

The scheme specifier http:// or https:// at the start of a web URI refers to Hypertext Transfer Protocol or HTTP Secure respectively. Unlike www, which has no specific purpose, these specify the communication protocol to be used for the request and response. The HTTP protocol is fundamental to the operation of the World Wide Web and the added encryption layer in HTTPS is essential when confidential information such as passwords or banking information are to be exchanged over the public Internet. Web browsers usually prepend http:// to addresses too, if omitted.

1.4.5 SECURITY

The web has become criminals' preferred pathway for spreading malware. Cybercrime carried out on the web can include identity theft, fraud, espionage and intelligence gathering. Web-based vulnerabilities now outnumber traditional computer security concerns, and