BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 4

DCA2201

COMPUTER NETWORKING

Unit 12

DNS and World Wide Web

Table of Contents

1. INTRODUCTION

In the previous unit, we have discussed various application layer protocols. All user applications are found in the Application layer. Application layer specifies the protocols and interfaces used by the users (hosts) in the network. In this unit, we will discuss one real application known as World Wide Web and one accessing mechanism known as Domain Name System. Programs access resources (web pages, mailboxes etc.) by using the network address of the computers, but these addresses are hard to remember and use, so one alternative mechanism for mapping host names to network address is required which is known as Domain Name System (DNS).

In this unit, we will start with a discussion on namespace. In the next section, we will discuss Domain Name System. In the last section, we will have a discussion on the World Wide Web and its features.

1.1 Objectives:

After studying this unit, you should be able to:

- *Describe namespace*
- *Explain domain name system*
- *Describe the World Wide Web*

2. NAME SPACE

To avoid ambiguity, the names assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses. A name space that maps each IP address to a unique name can be organized in two ways: flat or hierarchical.

Flat Name Space: In this scheme, a name is assigned to an IP address. A name in this space is a sequence of characters without structure. The names may or may not have a common section; if they do, it has no meaning. The main disadvantage of a flat name space is that it cannot be used in a large system such as the Internet because it may be centrally controlled to avoid ambiguity and duplication.

Hierarchical Name Space: In this scheme, each name is made of several parts. The first part may define the nature of the organization, the second part may define the name of the organization, the third part defines the departments in the organization, and so on. In this case, the authority to assign and control the name spaces can be decentralized. A central authority can assign the part of the name that defines the nature of the organization and the name of the organization. The responsibility of the rest of the name can be given to the organization itself. The organization can add suffixes (or prefixes) to the name to define its host or resources.

3. THE DOMAIN NAME SYSTEM

We can refer web pages, mailboxes and other resources by using the network address (IP address) of the computers on which they are stored. But these addresses are hard for people to remember and reproduce. Consequently, high-level, readable names were introduced in order to decouple machine names from machine addresses.

The initial solution for name resolution on the Internet was a file named Hosts.txt that was used on the now obsolete Advanced Research Projects Agency network (ARPANET), the predecessor of the modern-day Internet. When the number of hosts on the ARPANET was small, the Hosts.txt file was easy to manage because it consisted of unstructured names and their corresponding IP v4 addresses.

Computers on the ARPANET periodically downloaded Hosts.txt from a central location and used it for local name resolution. As the ARPANET grew into the Internet, the number of hosts began to increase dramatically and the centralized administration and manual distribution of a text file containing the names for computers on the Internet became unmanageable. To solve these problems, ***DNS (Domain Name System)*** was invented in 1983. The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme. It is primarily used for mapping host names to IP addresses.

The way DNS is used is described here. To map a name onto an IP address, an application program calls a library procedure called the *resolver*, passing it the name as a parameter. The resolver sends a query containing the name to a local DNS server, which looks up the name and returns a response containing the IP address to the resolver, which then returns it to the caller. The query and response messages are sent as UDP packets.

DNS Name Space

Managing a large and constantly changing set of names is an insignificant problem. DNS uses a hierarchical addressing similar to the postal system. In a postal system, name management is done by requiring letters to specify the country, state or province, city, street address, and name of the addressee.

For the Internet, the top of the naming hierarchy is managed by an organization called *ICANN (Internet Corporation for Assigned Names and Numbers)*. ICANN was created in 1998, as part of the maturing of the Internet to a worldwide, economic concern. The Internet is divided into over *250 top-level domains,* where each domain covers many hosts. Each domain is partitioned into subdomains, and these are further partitioned, and so on. All these domains can be represented by a tree as shown in figure 12.1. The leaves of the tree represent domains that have no subdomains. A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.

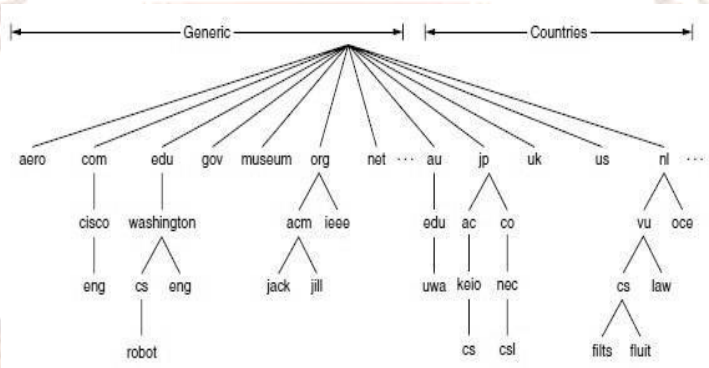

Fig 12.1: A portion of Internet domain name space

The top-level domains come in two categories. They are generic and country.

Table 12.1: Generic top-level domains

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

The generic domains, as shown in table 12.1, include original domains from the 1980s and domains introduced via applications to ICANN. The country domains include one entry for every country.

Getting a second-level domain, such as *name-of-company.com,* is easy. The top-level domains are run by *registrars* appointed by ICANN. Getting a name only requires going to a corresponding registrar (for com in this case) to check if the desired name is available and not somebody else's trademark. If there are no problems, the requester pays the registrar a small annual fee and gets the name.

Each domain is named by the path upward from it to the (unnamed) root. The components are separated by periods('dot'). Thus, the course details of SMU might be *course.smu.edu.* Domain names can be either *absolute or relative.* An absolute domain name always ends with a period *(e.g., course.smu.edu.),* whereas a relative one does not. Relative names have to be interpreted in some context to uniquely determine their true meaning. In both cases, a named domain refers to a specific node in the tree and all the nodes under it. Domain names are case-insensitive. Component names can be up to 63 characters long, and full path names must not exceed 255 characters.

Domain Resource Records

Each domain has a set of resource records associated with it. These records are from the DNS database. For a single host, the most common resource record is its IP address. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. The primary function of DNS is to map domain names onto resource records. A resource record is a *five-tuple*. Format is given below:

Domain_name Time_to_live Class Type Value

Domain_name tells the domain to which this record applies. Generally, many records exist for each domain and each copy of the database holds information about multiple domains. *Time_to_live* indicates how stable the record is. Information that is highly stable is assigned a large value and highly volatile information will get a small value. For internet information, class field is always *IN*. The *type* field tells the type/kind of record. There are many kinds of DNS records. The important types are shown in table 12.2.

Table 12.2: DNS resource record types

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

The most important record type is the A (Address) record. It holds a 32-bit IPv4 address of an interface for some hosts. *Value* field can be a number, a domain name, or an ASCII string. A short description of the Value fields for each of the principal record types is given in table 12.2.

Name Servers

A single name server could contain the entire DNS database and respond to all queries about it. But practically, this server would be so overloaded as to be useless. To avoid this problem associated with having only a single source of information, the DNS name space is divided into non-overlapping **zones**. Namespace of figure 12.1 can be divided as shown in figure 12.2. Each circled zone contains some part of the tree.

Zone boundaries are decided by zone's administrator. Consider figure 12.2, the University of Washington has a zone for *Washington.edu* that handles *eng.washington.edu* but does not handle *cs.washington.edu.* That is a separate zone with its own name servers. This decision has been made by zone's administrator when English department does not wish to run its own name server but Computer science department does. Each zone is associated with one or more name servers. These are hosts that hold the database for the zone. Normally, a zone will have one primary name server, which gets its information from a file on its disk, and one or more secondary name servers, which get their information from the primary name server.

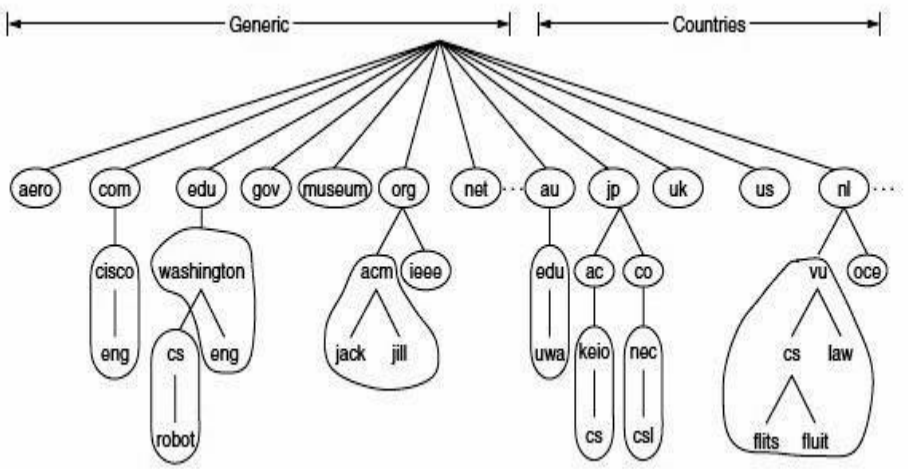

Fig 12.2: Part of the DNS name space divided into zones (which are circled)

The process of looking up a name and finding an address is called *name resolution*. When a resolver has a query about a domain name, it passes the query to a local name server. If that domain falls under that name server, it returns the authoritative resource records. An *authoritative* record is one that comes from the authority that manages the record and is thus always correct. If the information is not available in the local name server, then the local name server contacts the root name server in a hierarchical manner.

SELF-ASSESSMENT QUESTIONS – 1

1. A name space that maps each address to a unique name can be organized in two ways, they are _____or _____.
2. In namespace, a name is assigned to an address.
3. In namespace, each name is made of several parts.
4. DNS (Domain Name System) was invented in _____.
5. To map a name onto an IP address, an application program calls a library procedure called_____
6. For the Internet, the top of the naming hierarchy is managed by an organization called _____.
7. How many top-level domains does the Internet have?
   a) 120
   b) 250
   c) 140
   d) 280
8. A _____record contains five tuples.
9. is divided into non-overlapping zones.

4. THE WORLD WIDE WEB

World Wide Web (WWW) is an architectural framework for accessing linked content spread out over millions of machines all over the Internet. In 1994, CERN (the European Center for Nuclear Research) and M.I.T. signed an agreement setting up the *W3C (World Wide Web Consortium),* an organization devoted to further developing the Web, standardizing protocols, and encouraging interoperability between sites. Berners- Lee was the director. After that, several universities joined the consortium. www.w3c.org is the consortium's home page.

4.1 Architectural Overview

The web consists of a large, worldwide collection of content in the form of web pages, which are often called *pages.* Each page can have many links to other pages located anywhere in the world. By clicking a link, a user can access a link.one page point to another is called a hypertext, and the idea of hypertext was invented by Vannevar Bush (M.I.T. professor of electrical engineering), in 1945.

Figure 12.3 shows an example architecture. This page shows text and graphical elements. Some parts of the page are associated with links to other pages. A piece of text, icon, image, and so on associated with another page is called a *hyperlink.* To follow a link, the user places the mouse cursor on the linked portion of the page area and clicks.

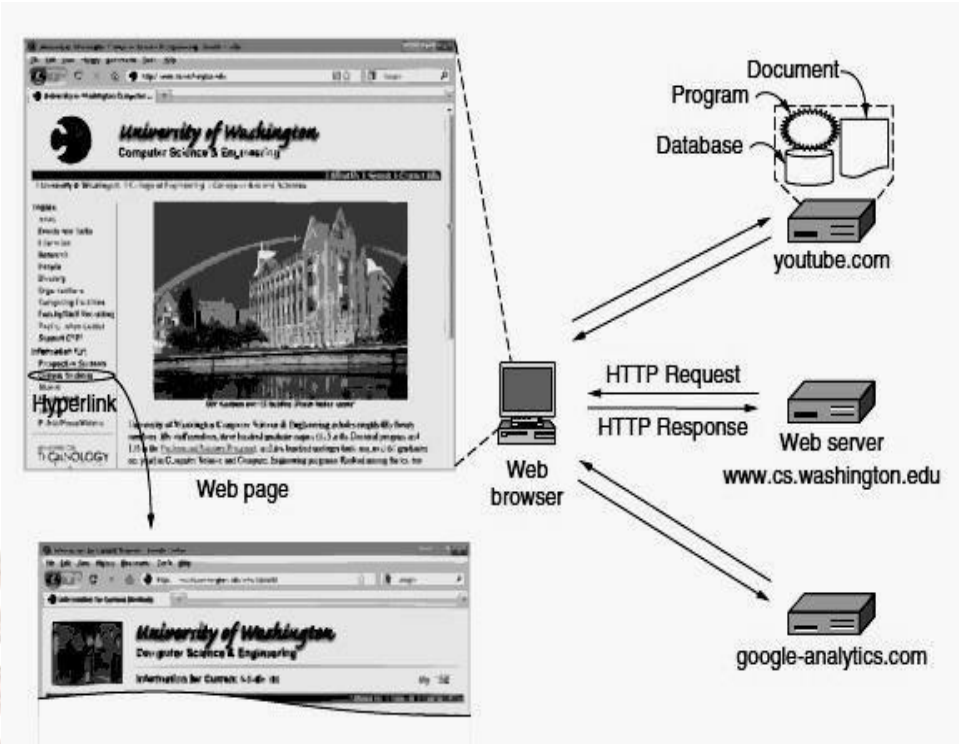

Fig 12.3: Architecture of the Web.

The basic model behind the display of pages is shown in figure 12.3. The browser is displaying a Web page on the client machine. Each page is fetched by sending a request to one or more servers, which respond with the contents of the page. The request-response protocol for fetching pages is a simple text-based protocol that runs over TCP, is called HTTP (HyperText Transfer Protocol). The content may be a page from the disk or the result of a database query and program execution. The page is a *static page* if it is a document that is the same every time it is displayed. The page is a *dynamic page,* it was generated on demand by a program or contains a program.

The Client Side

Consider the web browser side, a browser is a program that can display a web page and catch mouse clicks to items on the displayed page. When an item is selected, the browser follows the hyperlink and fetches the page selected. Each page is assigned a ***URL (Uniform Resource Locator)*** that effectively serves as the page's worldwide name. URLs have three parts: the protocol, the DNS name of the machine on which the page is located, and the path uniquely indicating the specific page. For example, the URL of a page is

http://www.smu.edu/courses.html. This URL has three parts, the protocol (http), the DNS name of the host *(www.smu.edu),* and the path name *(courses.html).*

Many browsers display which step they are currently executing in a status line at the bottom of the screen. In this way, when the performance is poor, the user can see if it is due to DNS not responding, a server not responding, or simply page transmission over a slow or congested network. Few common URL schemes are shown in table 12.3.

Table12.3: Some common URL schemes

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

The http protocol is the web's native language to communicate with web servers. The https is the protocol which provides encrypted transmissions for security. The ftp is file transfer protocol, which is used to access files by internet's file transfer protocol. It is possible to access a local file as a Web page by using the *file* protocol. The *mailto* protocol allows users to send email from a web browser. The *rtsp* and sip protocols are for establishing streaming media sessions and audio and video calls. The about protocol is a convention that provides information about the browser.

The Server side

When the user types in a URL or clicks on a line of hypertext, the browser parses the URL and interprets the part between *http://* and the next slash as a DNS name to look up. Armed with the IP address of the server, the browser establishes a TCP connection to port 80 on that server. Then it sends over a command containing the rest of the URL, which is the path to the page on that server. The server then returns the page for the browser to display. Different processes on the server side are listed below:

- Accept a TCP connection from a browser (client)
- Get to the path to the page, which is the name of the file requested.

- Get the file.
- Send the contents of the file to the client.
- Release the TCP connection.

For dynamic content, the third step may be replaced by the execution of a program (determined from the path) that returns the contents. The problem while accessing a file is that disk reads are very slow compared to program execution, and the same files may be read repeatedly from disk using operating system calls. We can solve the problem of serving a single request at a time by making the server multithreaded. In one design, the server consists of a front-end module that accepts all incoming requests and k processing modules as shown in figure 12.4.

The k + 1 threads all belong to the same process, so the processing modules all have access to the cache within the process' address space. When a request comes in, the front end accepts it and builds a short record describing it. It then hands the record to one of the processing modules. The processing module first checks the cache to see if the file needed is there. If so, it updates the record to include a pointer to the file in the record. If it is not there, the processing module starts a disk operation to read it into the cache. When the file comes in from the disk, it is put in the cache and also sent back to the client.

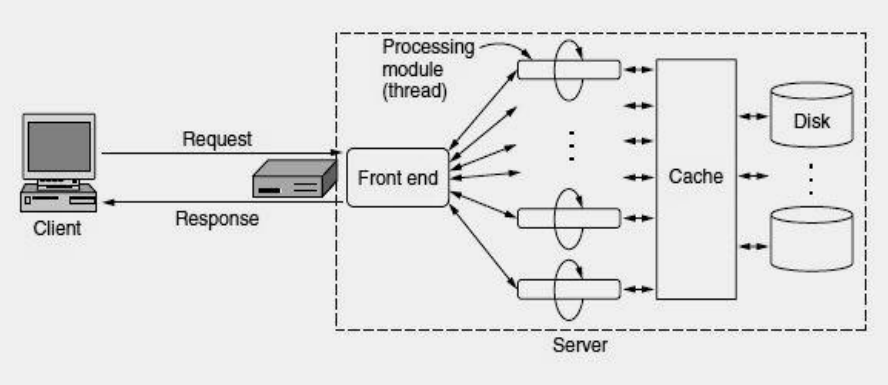

Fig 12.4: A multithreaded Web server with a front end and processing modules

The advantage of this scheme is that while one or more processing modules are blocked waiting for a disk or network operation to complete, other modules can be actively working on other requests. Modern Web servers do more than just accept path names and return files. In many servers each processing module performs a series of steps as shown below.

- Resolve the name of the Web page requested.
- Perform access control on the Web page.
- Check the cache.
- Fetch the requested page from disk or run a program to build it.
- Determine the rest of the response.
- Return the response to the client.
- Make an entry in the server log.

Step 1 is needed because the incoming request may not contain the actual name of a file or program as a literal string, instead it may contain built-in shortcuts that need to be translated. Step 2 checks to see if any access restrictions associated with the page are met. Steps 3 and 4 involve getting the page. Step 5 is about determining other parts of the response that accompany the contents of the page. The MIME type is one example. Step 6 is returning the page across the network. Step 7 makes an entry in the system log for administrative purposes, along with keeping any other important statistics.

4.2 Static Webpages

In simplest form, web pages are static. That is, they are just files sitting on some server that present themselves in the same way each time they are fetched and viewed. HTML (Hyper Text Markup language) allows users to create webpages that contain text, graphics, video, pointers to other webpages etc. Static or passive indicates that these webpages do not change frequently. This is similar to a printed page in a book. Once printed, it remains the same until another edition comes out where in pages could be revised. Similarly, until the time the author of a static webpage decides to change its contents, it will remain the same. This means that every time you access it, the same output is produced by static webpage. The nature of a static webpage is predictable.

Let's discuss how static webpages work. Suppose we are browsing through a website using a web browser. We request a specific webpage either by entering its URL or by clicking on the appropriate link. A summary of major steps involved is shown in figure 12.5.

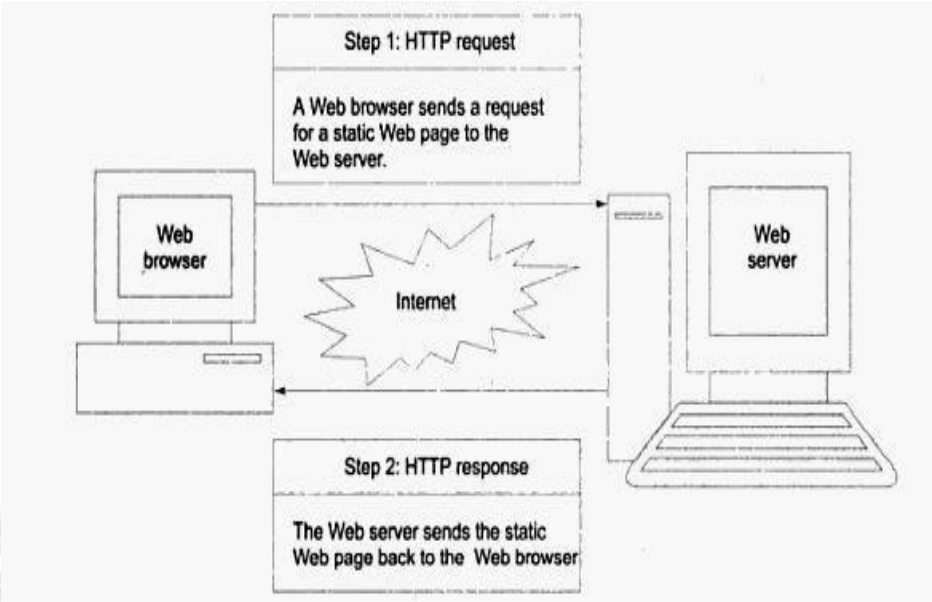

Fig 12.5: The process involved in the retrieval of a static web page

Different steps that take place after requesting a webpage is described below.

1. The web browser obtains the IP address corresponding to the DNS name portion of the URL, with the help of the DNS server.
2. Using the IP address thus obtained, a TCP connection is established between the web browser and the webserver.
3. The web browser sends a request for the web page to the web server using http protocol and GET instruction.
4. The webserver locates the web page associated with the URL on its hard disk with the help of the operating system running on the server and having found it, sends it back to the web browser.
5. The web browser accepts the web page, interprets it and displays the contents thus obtained on the display screen of the web browser.

Advantages and Disadvantages of static webpages.

Static web pages are very simple to create. Using plain text HTML tags, it can be easily created. Static web pages are simpler to maintain too. The time taken by a web browser to obtain a static page from a web site is much less compared to the other kinds of webpages.

Limitations of static web pages are: since the contents of a static web page have to be edited manually, it is not suitable for the information that changes frequently. Because a static web page received by a web browser cannot change on its own, it cannot have animation. Static

webpages cannot have audio capabilities as audio also means some changes and therefore executing a program, this means that we cannot play any kind of sound. In the case of a static web page, information travels in one direction only, from web server to web browser and there is no interaction between the user and the web pages.

4.3 Dynamic Webpages

A dynamic webpage is one that is created at run time, by executing some program on the server and then converting the output into HTML format and sending it to the browser to display it on client machine. Technologies such as frames and forms add interactivity to web pages.

Dynamic webpages are extremely important in the case of e-commerce. The general situation of dynamic web page is shown in figure 12.6. Consider the example of a map service application that allows users to enter an address and provides a corresponding map of the location. The web server must use a program to create a page that indicates the location. In figure 12.6, this action is shown in steps 1 to 3. Step 1 is a request to the server which causes a program to run on server. In step 2, the program consults a database to generate an appropriate page. In step 3, the requested page is returned to the browser (client).

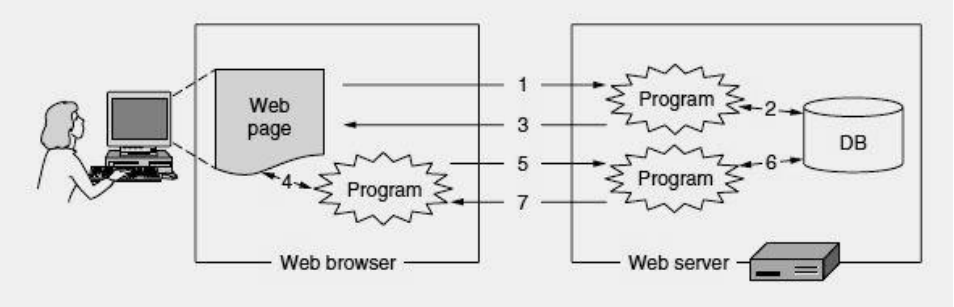

Fig 12.6: Dynamic pages

In this example, application program allows the user to find routes and explore nearby areas. Step 4 shows the update or zooming in or out of the page as directed by user. If a client user needs more interactions, it will send a request to server as shown in step 5. Server will retrieve more information from database as shown in step 6 and returns the response (step 7). Users may not be aware of these requests and response that happen in background.

Server-Side Dynamic Web Page Generation

Many standard APIs (Application Programming Interface) have been developed for web servers to invoke programs. The existence of these interfaces makes it easier for developers to extend different servers with web applications. First API method for handling dynamic pages is called the CGI (Common Gateway Interface). This provides an interface to allow web servers to interact with back-end programs and scripts that can accept the input and generate HTML pages in response. This program may be written in any of the scripting languages like Ruby, Perl etc.

Another popular language for writing this script is PHP (Hypertext Preprocessor). Server identify PHP pages by using the file extension php. Compared to CGI, PHP is easy to use and it is a powerful programming language for interfacing the web and server database.

Client-Side Dynamic Web Page Generation

On the server side, input and interactions with the database and web server is handled by CGI scripts or PHP. By using these two languages, we can accept incoming information from forms, search this information in database and generate HTML pages as result. But these languages cannot be used to respond to mouse movements or interact with users directly. For this interaction, scripts embedded in HTML pages that are executed on the client machine is required. The technologies used to develop these interactive web pages are known as *dynamic HTML.*

The most popular client-side scripting language is Javascript. JavaScript is entirely different from the Java programming language. It is a very high-level language, like other scripting languages. An alternative client-side programming language is VBScript, which is based on visual basics. Another popular method across platforms is the use of applets. Applets are very small java code that can be converted into machine language using JVM (Java Virtual Machine). Applets can be embedded in HTML pages and interpreted by JVM compatible browsers.

4.4 Mobile Web and Web Search

Mobile web refers to accessing the World Wide Web from a handheld mobile device. Browsing the web using a mobile can be very useful. But when compared to desktop computers, mobile phones provide several difficulties in web browsing such as: relatively

small screen prevents large pages or images display, limited input facilities make it difficult to enter lengthy URLs, network bandwidth is limited and is expensive too, connectivity problems, computing power is limited due to battery life. So, we can say that using desktop content for mobile web will provide a frustrating user experience.

Early approaches to the mobile web devised a new protocol stack tailored to wireless devices with limited capabilities. Most well-known example of this strategy is WAP (Wireless Application Protocol). The approach which is increasingly used now is to run the same protocols for mobiles and desktops. Websites will deliver mobile friendly content when the user raise request from a mobile device. Web server is able to detect the request coming from mobile phones by looking at the request headers. The user-agent header is useful in this regard because it identifies the browser software. So, one of the best approach is to reduce the size of pages by the use of compression, since the cost of communication is higher than those of computation. To help mobile users, there is a logo to indicate pages that can be viewed on the mobile web.

Another useful tool is a minimal version of HTML called ***XHTML basic.*** This language is a subset of XHTML that is intended for use by mobile phones, televisions, PDAs, cars etc. This version does not support style sheets, scripts, or frames, but most of the standard tags are there.

However, all pages are not designed to work well on the mobile web. So, a complementary approach is the use of ***content transformation*** or ***transcoding.*** In this approach, a middleware computer which is located between the mobile and the server, accepts requests from the mobile, fetches content from the server and transforms it to mobile web content. While considering the case of protocols, the HTTP, TCP and IP protocols used by the web may consume a significant amount of bandwidth. To face this problem, WAP and other solutions defined special purpose protocols. Header compression technologies such as ROHC (RObust Header Compression) can also reduce the overhead of these protocols.

Web Search

In 1998, Sergey Brin and Larry Page, then graduate students at Stanford, formed a startup called Google to build a better web search engine. Google became a public company in 2004. Over one billion users are using web search every day. The major web search sites available

today are google, yahoo and Bing. To perform a web search, the user will input a URL of the web search site. Then, the user submits search terms using a form. Search engine will perform a query on the database using the input and return the result as a dynamic web page. The Web search engine must have a database of pages to run a query. This means that it is theoretically possible to start with a handful of pages and find all other pages on the Web by doing a traversal of all pages and links. This process is called ***web crawling***. All Web search engines use web crawlers.

Many webpages are dynamic pages with programs that display different pages based on user interaction. One issue with crawling is that searching dynamic web pages is difficult from static pages which are easy to traverse. Web crawlers cannot find these dynamic pages. This hidden content is called the ***deep web.*** Another consideration is how to process all the crawled data. Making sense of this data is another issue. There is an issue of conversion between formats or translation between languages. Another feature of web search is that it provides a higher level of naming. We can search by name instead of URL, we can better remember name than URLs. Advertising is the economic engine that has driven the growth of Web search. Search queries can be matched with most valuable advertisements, and this has given rise to new problems such as click fraud, in which programs make users click on advertisements to get payments that have not been fairly earned.

SELF-ASSESSMENT QUESTIONS – 2

10. World Wide Web (WWW) is an architectural framework for accessing linked content spread out over millions of machines all over the Internet *(True/False)*
11. A piece of text, icon, image, and so on associated with another page is called a _____.
12. A page which is a document that is the same every time it is displayed is called _____.
13. The page which is generated on demand by a program or contains a program is called _____.
14. Web crawlers cannot find all dynamic pages. This hidden content is called _____

5. SUMMARY

Let us recapitulate the important concepts discussed in this unit:

- A name space that maps each address to a unique name can be organized in two ways. They are flat name space and hierarchical name space.
- DNS (Domain Name System) is primarily used for mapping host names to IP addresses.
- DNS (Domain Name System) was invented in 1983.
- For the Internet, the top of the naming hierarchy is managed by an organization called ICANN (Internet Corporation for Assigned Names and Numbers).
- The Internet is divided into over 250 top-level domains.
- The top-level domains come in two categories. They are generic and countries.
- Each domain has a set of resource records associated with it.
- To avoid this problem associated with having only a single source of information, the DNS name space is divided into non-overlapping zones.
- The process of looking up a name and finding an address is called name resolution.
- World Wide Web (WWW) is an architectural framework for accessing linked content spread out over millions of machines all over the Internet.
- Each web page is assigned a URL (Uniform Resource Locator) that effectively serves as the page's worldwide name.

- A static web page is one which do not change frequently. A dynamic webpage is one that is created at run time.

6. TERMINAL QUESTIONS

1. Differentiate between flat and hierarchical name space.
2. Write short notes on domain resource records.
3. Explain name servers.
4. Differentiate between static and dynamic web pages.
5. Explain about mobile web.

7. ANSWERS

Self-Assessment Questions

1. Flat, hierarchical
2. Flat
3. Hierarchical
4. 1983
5. Resolver
6. ICANN (Internet Corporation for Assigned Name and Number)
7. (b) 250
8. Resource record
9. DNS namespace
10. (a) True
11. Hyperlink
12. Static page
13. Dynamic page
14. Deep web

Terminal Questions

1. In flat namespace, a name is assigned to an address. A name in this space is a sequence of characters without structure. In hierarchical name space, each name is made of several parts. The first part may define the nature of the organization, the second part

may define the name of the organization, and the third part defines the departments in the organization, and so on… (Refer section 2 for more details).

2. Each domain has a set of *resource records associated with it.* These records are from the DNS database. For a single host, the most common resource record is its IP address. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. The primary function of DNS is to map domain names onto resource records. (Refer section 3 for more details).
3. A single name server could contain the entire DNS database and respond to all queries about it. But practically, this server would be so overloaded as to be useless. To avoid this problem associated with having only a single source of information, the DNS name space is divided into non-overlapping zones. (Refer section 3 for more details).
4. In simplest form, web pages are static. That is, they are just files sitting on some server that present themselves in the same way each time they are fetched and viewed. A dynamic webpage is one that is created at run time, by executing some program on the server and then converting the output into HTML format and sending it to the browser to display it on client machine. (Refer section 4.2 and 4.3 for more details).
5. Browsing the web using a mobile can be very useful. But when compared to desktop computers, mobile phones provide several difficulties in web browsing such as: relatively small screen prevents large pages or images display, limited input facilities make it difficult to enter lengthy URLs, network bandwidth is limited and is expensive too, connectivity problems, computing power is limited due to battery life. (Refer section 4.4 for more details).

References

- Andrew S Tanenbaum, David J. Wetherall, "*Computer Networks,"* Fifth edition.
- Larry L. Peterson, Bruce S. Davie, "*Computer Networks – a Systems Approach*," Fifth edition.
- James F. Kurose, Keith W. Ross, "*Computer Networking – A top-down approach*," Sixth edition.
- Behrouz A. Forouzan, Sophia Chung Fegan, "*Data Communication and Networking*," Fourth edition.

- William Stallings, "*Computer Networking With Internet Protocols and Technology,*" Third edition.