1. INTRODUCTION

1.1. INTERNET PRINCIPLES

This page reviews the central concepts of software on the internet. It briefly explains:

- **TCP/IP**
- **UDP**
- **IP Addresses**
- **domain names**
- **the domain name system**
- **ports**
- **sockets**
- **URL's**

1.1. 1. TCP/IP

The Internet is the network that connects computers all over the world. It works according to a set of agreed protocols. **TCP (Transmission Control Protocol)** and **IP(Internet Protocol)** are the most commonly-used protocols for using the Internet. (But there are others at lower levels.) The combination is simply known as **TCP/IP.**

The Internet is a **packet switching** system. Any message is broken into packets that are transmitted independently across the interment (sometime by different routes). These packets are called **datagrams.** The route chosen for each datagram depends on the traffic at any point in time. Each datagram has a header of between 20 and 60 bytes, followed by the payload of up to 65,515 bytes of data. The header consists of, amongst other data:

1. the version number of the protocol in use
2. IP address of sender
3. IP address of destination

TCP breaks down a message into packets. At the destination, it re-assembles packets into messages. It attaches a checksum to each packet. If the checksum doesn't match the computed checksum at the destination, the packet is re-transmitted. Thus TCP ensures reliable transmission of information. In summary, TCP:

1. provides re-transmission of lost data
2. delivery of data in the correct order

1.1.2. UDP

Most applications use TCP. However, an example of a situation in which it is desirable to use a lower-level protocol is the case of audio **streaming.** If you want to download a sound file, it can take some time, even though it may be compressed. You have to wait (maybe some time) for the complete file to download before it can be played. An alternative is to listen to the sound as it is being downloaded - streaming. One of the most popular technologies is called RealAudio.

RealAudi does not use TCP because of its overhead. The sound file is sent in IP packets using the **UDP (User Datagram Protocol)** instead of TCP. UDP is an unreliable protocol:

- it doesn't guarantee that a packet will arrive

- it doesn't guarantee that packets are in the right order

UDP doesn't re-send a packet if it is missing or there is some other error, and it doesn't assemble packets into the correct order. But it is faster than TCP. In this application, losing a few bits of data is better than waiting for the re-transmission of some missing data. The application's major mission is to keep playing the sound without interruption. (In contrast, the main goal of a file transfer program is to transmit the data accurately.)

The same mechanism is used with video streaming.

UDP is a protocol at the same level as TCP, above the level of IP.

1.1.3. Domain name

The **domain name** is the user-friendly equivalent of an IP address. Used because the numbers in an IP address are hard to remember and use. Also known as a host name. Example:

shu.ac.uk

Such a name starts with the most local part of the name and is followed by the most general. The whole name space is a tree, whose root has no name. The first level in the tree has com, org, edu, UK, etc.

The parts of a domain name don't correspond to the parts of an IP address. Indeed

domain names don't always have 4 parts - they can have 2, 5 or whatever.

All applications that use an address should work whether an IP address or a domain name is used. In fact, a domain name is converted to an IP address before it is used.

Exercise: Compare and contrast IP addresses with domain names.

1.1.4. Domain Name System

A program, say a Web browser, that has a domain address usually needs to convert it into an IP address before making contact with the server. The domain name system (DNS) provides a mapping between IP addresses and domain names. All this information cannot be all in one place and so it is a distributed database.

1.1.5. Clients, Servers and Peers

A network application usually involves a client and a server. Each is a process (an independently running program) running on a (different) computer.

A server runs on a host and provides some particular service, e.g. e-mail, access to local Web pages. Thus a Web server is a server. A commonly-used web server program is called Apache.

A client runs on a host but needs to connect with a sever on another host to accomplish its task. Usually, different clients are used for different tasks, e.g. Web browsing and e-mail. Thus a Web browser is a client.

Some programs are not structured as clients and servers. For example a game, played across the internet by two or more players is a peer to peer relationship. Other examples: chat, internet phone, shared whiteboard.

1.1.6. Port Numbers

To identify a host machine, an IP address or a domain name is needed. To identify a particular server on a host, a port number is used. A port is like a logical connection to a machine. Port numbers can take values from 1 to 65,535. It has no correspondence with the physical connections, of which there might be just one. Each type of service has, by convention, a standard port number. Thus 80 usually means Web serving and 21 means file transfer. If the default port number is used, it can be omitted in the URL (see below). For each port supplying a service there is a server program waiting for any requests. Thus a web server program listens on port 80 for any incoming requests. All these server programs run together in parallel on the host machine.

When a packet of information is received by a host, the port number is examined and the packet sent to the program responsible for that port. Thus the different types of request are distinguished and dispatched to the relevant program.

The following table lists the common services, together with their normal port numbers. These conventional port numbers are sometimes not used for a variety of reasons. One example is when a host provides (say) multiple web servers, so only one can be on port 80. Another reason is where the server program has not been assigned the privilege to use port 80.

1.1.7. Sockets

A socket is the software mechanism for one program to connect to another. A pair of programs opens a socket connection between themselves. This then acts like a

telephone connection - they can converse in both directions for as long as the connection is open. (In fact, data can flow in both directions at the same time.) More than one socket can use any particular port. The network software ensures that data is routed to or from the correct socket.

When a server (on a particular port number) gets an initial request, it often spawns a separate thread to deal with the client. This is because different clients may well run different speeds. Having one thread per client means that the different speeds can be accommodated. The new thread creates a (software) socket to use as the connection to the client. Thus one port may be associated with many sockets.

1.1.8. Streams

Accessing information across the Internet is accomplished using streams. A stream is a serial collection of data, such as can be sent to a printer, a display or a serial file. Similarly a stream is like data input from a keyboard or input from a serial file. Thus reading or writing to another program across a network is just like reading or writing to a serial file.

1.1.9. URL

A URL (Uniform Resource Locator) is:

- a unique identifier for any resource on the Internet
- typed into a Web browser
- used as a hyperlink within a HTML document
- quoted as a reference to a source

A URL has the structure:

protocol: //hostname[:port]/[pathname]/filename#section

The host name is the name of the server that provides the service. This can either be a domain name or an IP address.

The port number is only needed when the server does not use the default port number. For example, 80 is the default port number for HTTP.

1.2. BASIC WEB CONCEPTS

This section describes the essential concepts for working with PowerDynamo (Storing, locating, and transmitting information on the Web)

How information is located: the URL

To move from one page of a document to another page, or to another document on the same or another Web site, the user clicks a hyperlink (usually just called a link) in the document shown in their Web client. Documents and locations within documents are identified by an address, defined as a Uniform Resource Locator, or **URL**. The following URL illustrates the general form:

http://www.sybase.com/productsl

or

http://www.sybase.com/inc/corpinfo/mkcreate.html

URLs contain information about which server the document is on, and may also specify a particular document available to that server, and even a position within the document. In addition, a URL may carry other information from a Web client to a Web server, including the values entered into fields in an HTML form.

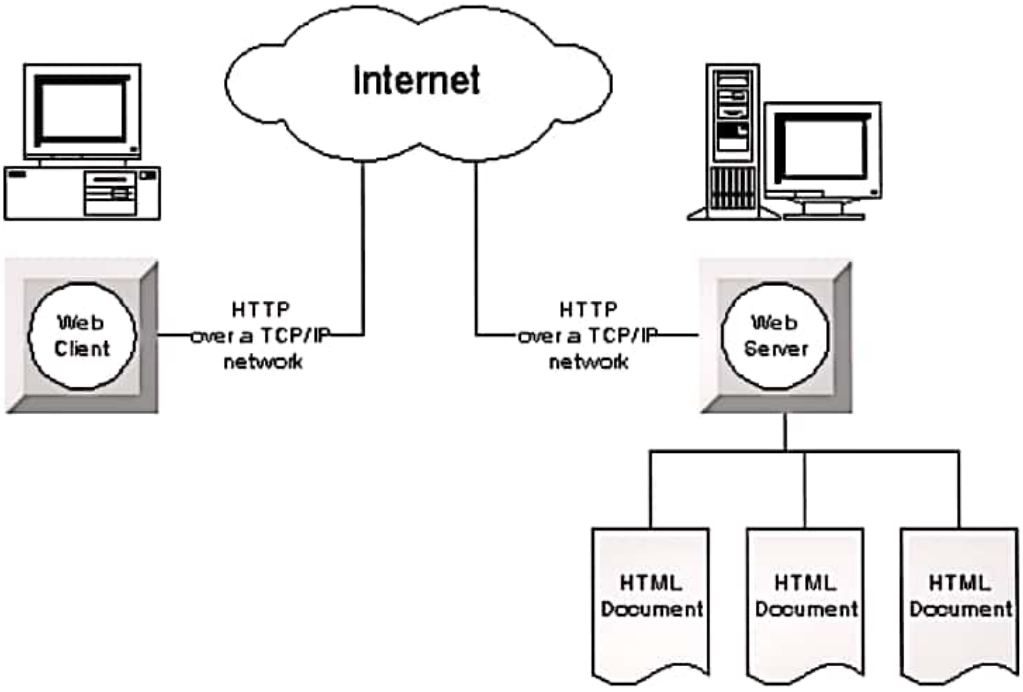
For more information about URLs and addresses on the Web, see the material on the World Wide Web Consortium pages, at the following address:

http://www.w3.org/pub/WWW/Addressing/

When a user clicks a link on a document on their Web client, the URL is sent to the server of the indicated Web site. The Web server locates the document, and sends the HTML to the Web client across the network.

The below figure illustrates, information is stored at **Web sites**. Access to the information is managed by a **Web server** for the site. Users access the information using **Web clients**, which are also called **browsers**.

Figure 2-1: Accessing information on the Web



Information on the Web is stored in documents, using a language called **HTML** (HyperText Markup Language). Web clients must interpret HTML to be able to display the documents to a user. The protocol that governs the exchange of information between the Web server and Web client is named **HTTP** (HyperText Transfer Protocol).

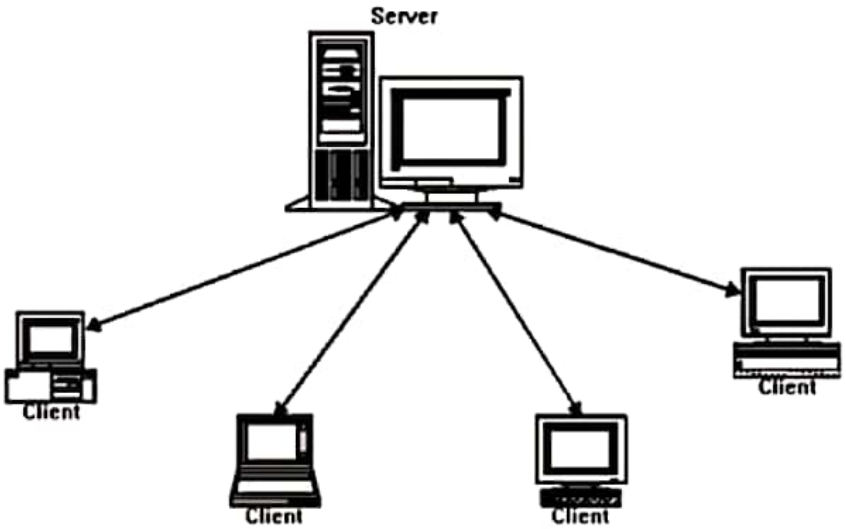
External data in HTML documents

HTML documents can include graphics or other types of data by referencing an external file (for example, a GIF or JPEG file for a graphic). Not all these external formats are supported by all Web clients. When the document contains such data, the Web client can send a request to the Web server to provide the relevant graphic. If the Web client does not support the format, it does not request the information from the server.

1.3. CLIENT / SERVER MODEL

1.3.1. Client server

Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfils the request. Although programs within a single computer can use the client/server idea, it is a more important idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. Computer transactions using the client/server model are very common. For example, to check your bank account from your computer, a client program in your computer forwards your request to a server program at the bank. That program might in turn forward the request to its own client program that sends a request to a database server at another bank computer to retrieve your account balance. The balance is returned back to the bank data client, which in turn serves it back to the client in your personal computer, which displays the information for you.



Client server model diagram

Source: http://www.javaworld.com/javaworld/jw-10-2001/jw-1019-jxta.html -

The communications method in computing includes Local procedure calls and Remote procedure calls.

1.3.2. Remote Procedure Call:

This is a protocol that one program can use to request the services from other located in other machine in a network without having to understand the network details. Usually when a program using RPC are compiled into an executable program, a stub is included which acts as the representative of remote procedure code. When the program is run, and the procedure issue the stub receives a request and forwards it to the client runtime in the local computer by the daemons. The client runtime program knows the address of the remote computer and server application. It then sends the request across the network .The server also have a runtime program and stub that interface with remote procedure. The result is returned the same way.

1.3.3. Local Procedure Call

A local procedure call (LPC) is an interprocess communication facility for high-speed message passing

In Windows NT, client-subsystem communication happens in a fashion similar to that in the MACH operating system. Each subsystem contains a client-side DLL that links with the client executable. The DLL contains stub functions for the subsystem's API. Whenever a client process–an application using the subsystem interface–makes an API call, the corresponding stub function in the DLL passes on the call to the subsystem process. The subsystem process, after the necessary processing, returns the results to the client DLL. The stub function in the DLL waits for the subsystem to return the results and, in turn, passes the results to the caller. The client process simply resembles calling a normal procedure in its own code. In the case of RPC, the client actually calls a procedure sitting in some remote server over the network–hence the name remote

procedure call. In Windows NT, the server runs on the same machine; hence the mechanism is called as a local procedure call.

LPC is designed to allow three methods of exchanging messages:

- Message that is shorter than 256 bytes can be sent by calling LPC with a buffer containing the message. That is a small message. This message is then copied from the address space of the sending process into system address space, and then to the address space of the receiving process.

- If a client and a server want to exchange more than 256 bytes of data, they can choose to use a shared section to which both are mapped. The sender places message data in the shared section and then sends a small message to the receiver with pointers to where the data is to be found in the shared section.

- When a server wants to read or write larger amount of data than will fit in a shared section, data can be directly read from or written to a client's address space. The LPC component supplies two functions that a server can use to accomplish this. A message sent by the first method is used to synchronize the message passing.

There are three types of LPC. The first type sends small messages up to 304 bytes. The second type sends larger messages. The third type of LPC is called as Quick LPC and used by the Win32 subsystem in Windows NT 3.51.

The first two types of LPC use port objects for communication. Ports resemble the sockets or named pipes in Unix. A port is a bi-directional communication channel between two processes. However, unlike sockets, the data passed through ports is not