FIBER DISTRIBUTED DATA INTERFACE

Introduction

This unit is on fiber distributed interface, where we shall talk about the transmission media and specification, station attachment types as well as optical bypass switches.

Learning Objectives

After reading this unit you should be able to:

1. Describe the FDDI Infrastructure
2. Explain how to use this dual network topology

Unit content

SESSION 1-4: FIBER DISTRIBUTED DATA INTERFACE (FDDI)

1-4.1 Background

1-4.2 FDDI Transmission media

1-4.3 FDDI Specification

1-4.4 FDDI Station Attachment Types

SESSION 2-4: OPTICAL BYPASS SWITCH

2-4.1 Optical Bypass Switch

SESSION 1-4: FIBER DISTRIBUTED DATA INTERFACE (FDDI)

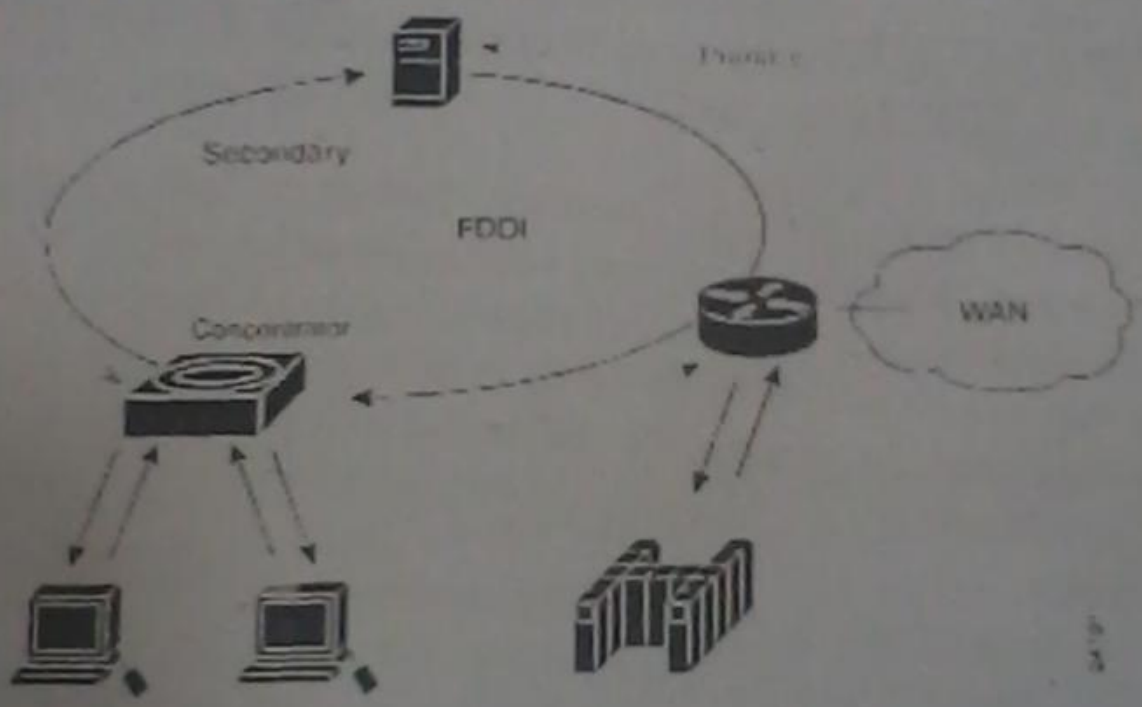
1-4.1 Background

The Fiber Distributed Data Interface (FDDI) specifies a 100-Mbps token-passing, dual-ring LAN using fiber-optic cable. FDDI is frequently used as high-speed backbone technology because of its support for high bandwidth and greater distances than copper. It should be noted that relatively recently, a related copper specification, called Copper Distributed Data Interface (CDDI) has emerged to provide 100-Mbps service over copper. CDDI is the implementation of

FDDI protocols over twisted-pair copper wire. This chapter focuses mainly on FDDI specifications and operations, but it also provides a high-level overview of CDDI.

FDDI uses dual-ring architecture with traffic on each ring flowing in opposite directions (called *counter-rotating*). The dual-rings consist of a primary and a secondary ring. During normal operation, the primary ring is used for data transmission, and the secondary ring remains idle. The primary purpose of the dual rings, as will be discussed in detail later in this chapter, is to provide superior reliability and robustness. Figure 1 shows the counter-rotating primary and secondary FDDI rings.

Figure 1: FDDI uses counter-rotating primary and secondary rings.



Standards

FDDI was developed by the American National Standards Institute (ANSI) X3T9.5 standards committee in the mid-1980s. At the time, high-speed engineering workstations were beginning to tax the bandwidth of existing local area networks (LANs) based on Ethernet and Token Ring). A new LAN media was needed that could easily support these workstations and their new distributed applications. At the same time, network reliability had become an increasingly important issue as system managers migrated mission-critical applications from large computers to networks. FDDI was developed to fill these needs. After completing the FDDI specification, ANSI submitted FDDI to the International Organization for Standardization (ISO), which created an international version of FDDI that is completely compatible with the ANSI standard version.

4-4.2 FDDI Transmission Media

FDDI uses optical fiber as the primary transmission medium, but it also can run over copper cabling. As mentioned earlier, FDDI over copper is referred to as *Copper-Distributed Data Interface* (CDDI). Optical fiber has several advantages over copper media. In particular, security, reliability, and performance all are enhanced with optical fiber media because fiber does not emit electrical signals. A physical medium that does emit electrical signals (copper) can be tapped and therefore would permit unauthorized access to the data that is transiting the medium. In addition, fiber is immune to electrical interference from radio frequency interference (RFI) and electromagnetic interference (EMI). Fiber historically has supported much higher bandwidth (throughput potential) than copper, although recent technological advances have made copper capable of transmitting at 100 Mbps. Finally, FDDI allows two kilometers between stations using multi-mode fiber, and even longer distances using a single mode.

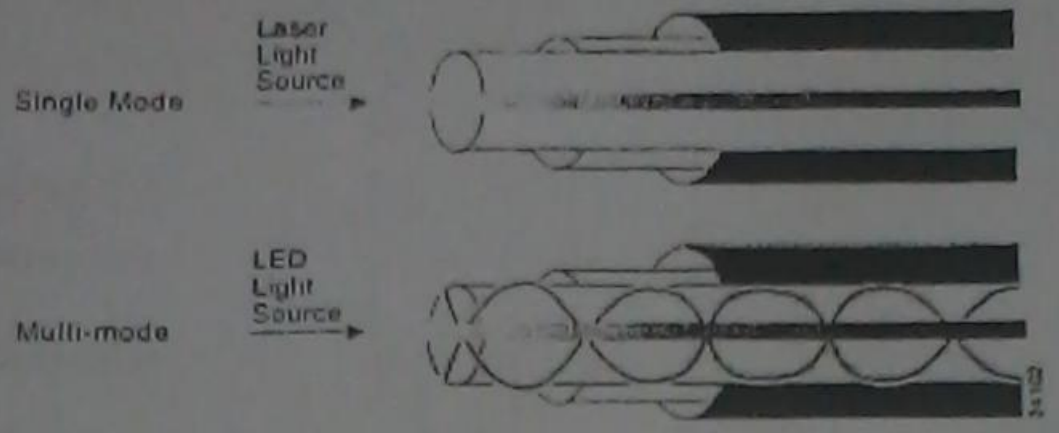
FDDI defines two types of optical fiber: single-mode and multi-mode. A *mode* is a ray of light that enters the fiber at a particular angle. Multi-mode fiber uses LED as the light-generating devices, while single-mode fiber generally uses lasers.

Multi-mode fiber allows multiple modes of light to propagate through the fiber. Because these modes of light enter the fiber at different angles, they will arrive at the end of the fiber at different times. This characteristic is known as *modal dispersion*. Modal dispersion limits the bandwidth and distances that can be accomplished using multi-mode fibers. For this reason, multi-mode fiber is generally used for connectivity within a building or within a relatively geographically contained environment.

Single-mode fiber allows only one mode of light to propagate through the fiber. Because only a single mode of light is used, modal dispersion is not present with single-mode fiber. Therefore, single-mode is capable of delivering considerably higher performance connectivity and over much larger distances, which is why it generally is used for connectivity between buildings and within environments that are more geographically dispersed.

Figure 2 depicts single-mode fiber using a laser light source and multi-mode fiber using a light emitting diode (LED) light source.

Figure 2: Light sources differ for single-mode and multi-mode fibers



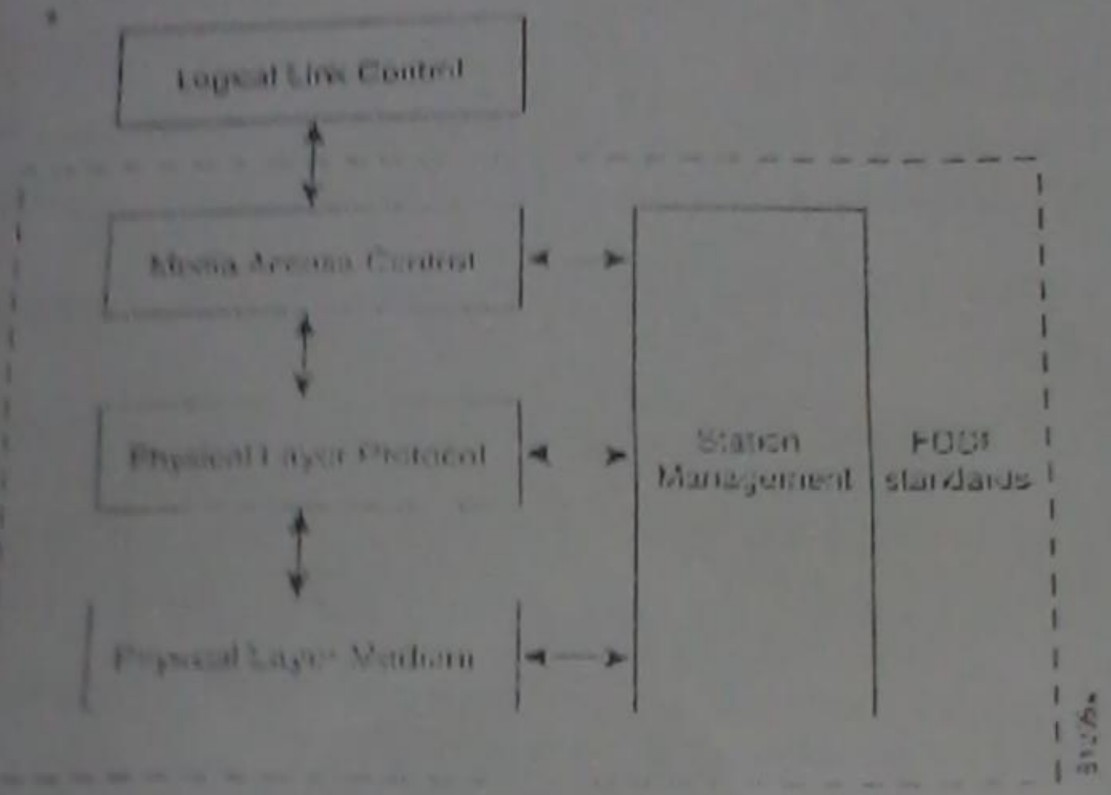
1-4.3 FDDI Specifications

FDDI specifies the physical and media-access portions of the OSI reference model. FDDI is not actually a single specification, but it is a collection of four separate specifications each with a specific function. Combined, these specifications have the capability to provide high-speed connectivity between upper-layer protocols such as TCP/IP and IPX, and media such as fiber-optic cabling.

FDDI's four specifi tions are the Media Access Co trol (MAC), Physical Layer Pr tocol (PHY), Physical-Medium Dependent (PMD), and Station Management (SMT). The MAC specification defines how the medium is accessed, i..luding frame format, token han ing, addressing, algorithms for calculating cyclic redundancy check (CRC) value, and error-recovery mechanisms. The PHY specification defines data encoding/decoding procedures, clocking requirements, and framing, among other functions. The PMD specification defines the characteristics of the transmission medium, including fiber-optic links, power levels, bit-error rates, optical components, and connectors. The SMT specification defines FDDI station configuration, ring configuration, and ring control features, including station insertion and removal, initialization, fault isolation and recovery, scheduling, and statistics collection.

FDDI is similar to IEEE 802.3 Ethernet and IEEE 802.5 Token Ring in its relationship with the OSI model. Its primary purpose is to provide connectivity between upper OSI layers of common protocols and the media used to connect network devices. Figure 3 illustrates the four FDDI specifications and their relationship to each other and to the IEEE-defined Logical-Link Control (LLC) sublayer. The LLC sublayer is a component of Layer 2, the MAC layer, of the OSI reference model.

Figure 3: FDDI specifications map to the OSI hierarchical model.



1-4.4 FDDI Station-Attachment Types

One of the unique characteristics of FDDI is that multiple ways actually exist by which to connect FDDI devices. FDDI defines three types of devices: single-attachment station (SAS), dual-attachment station (DAS), and a concentrator.

An SAS attaches to only one ring (the primary) through a concentrator. One of the primary advantages of connecting devices with SAS attachments is that the devices will not have any effect on the FDDI ring if they are disconnected or powered off. Concentrators will be discussed in more detail in the following discussion.

Each FDDI DAS has two ports, designated A and B. These ports connect the DAS to the dual FDDI ring. Therefore, each port provides a connection for both the primary and the secondary ring. As you will see in the next section, devices using DAS connections will affect the ring if they are disconnected or powered off. Figure 4 shows FDDI DAS A and B ports with attachments to the primary and secondary rings.

ingle-ring topology. Data continues to be transmitted on the FDDI ring without performance
act during the wrap condition. Figure 6 and Figure 7 illustrate the effect of a ring wrapping in
FDDI.

gure 6: A ring recovers from a station failure by wrapping.

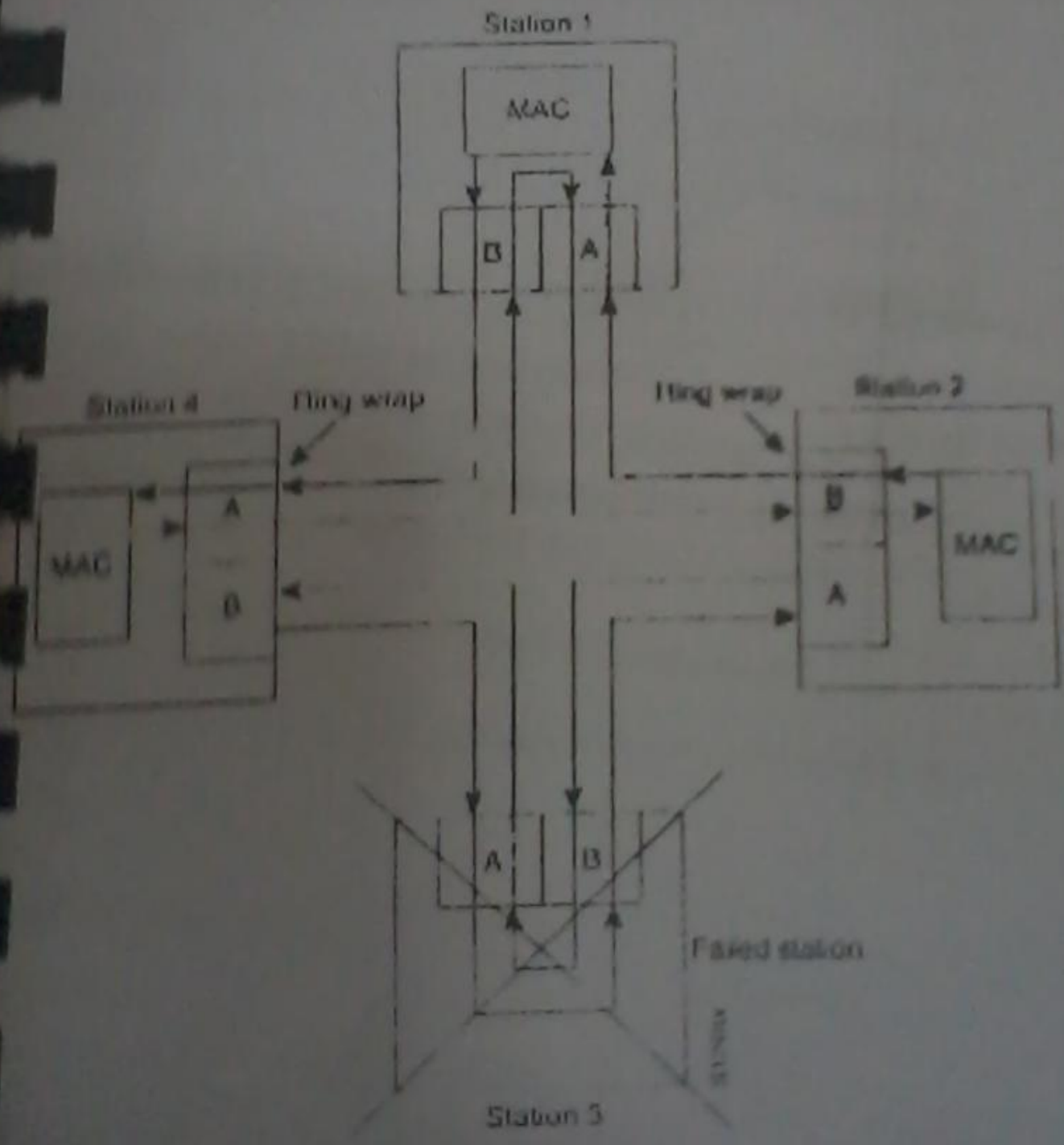
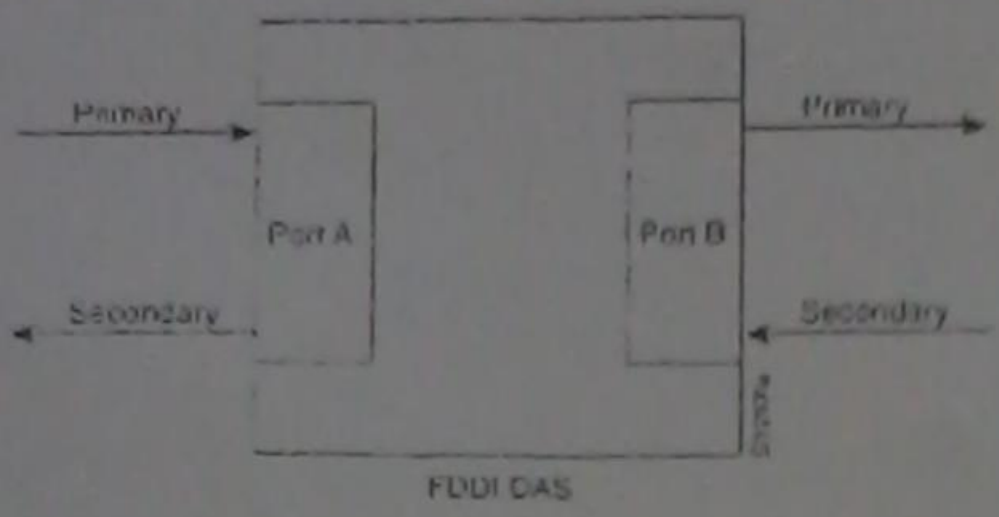
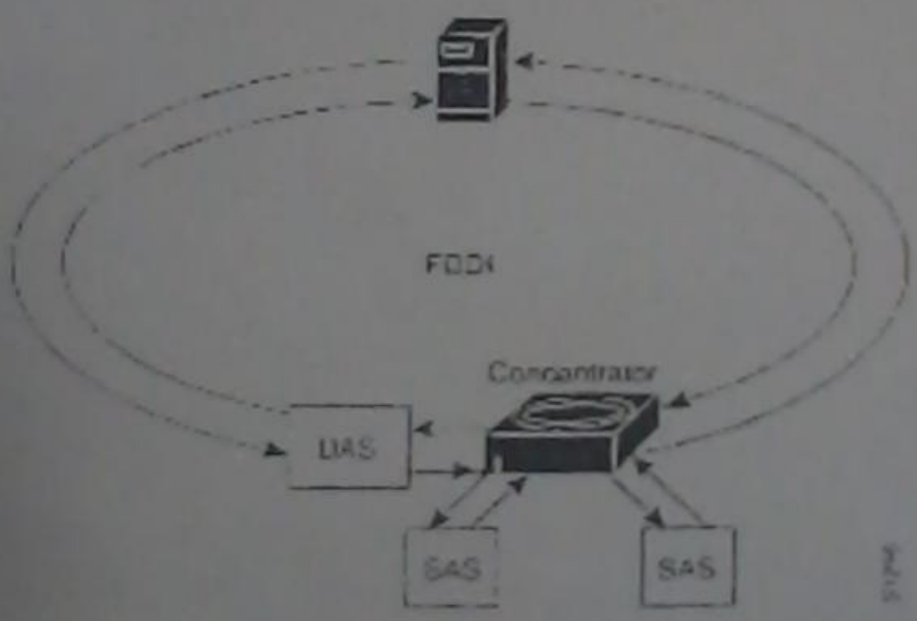


Figure 4: FDDI DAS ports attach to the primary and secondary rings.



An FDDI concentrator (also called a *dual-attachment concentrator* [DAC]) is the b of an FDDI network. It attaches directly to both the primary and secondary rings and the failure or power-down of any SAS does not bring down the ring. This is partic when PCs, or similar devices that are frequently powered on and off, connect to th 5 shows the ring attachments of an FDDI SAS, DAS, and concentrator.

Figure 5: A concentrator attaches to both the primary and secondary rings.



FDDI Fault Tolerance

FDDI provides a number of fault-tol ant features. In particular, FDDI's dual-ring the implementation of the optical bypass switch, and dual-homing suppo make FD media technology.

Dual Ring

FDDI's primary fault-tolerant feature is the dual ring. If a station on the dual r powered down, or if the cable is damaged, the dual ring is automatically *wrapped* (onto itself) into a single ring. When the ring is wrapped, the dual-ring topolo

...ring topology. Data continues to be transmitted on the FDDI ring without performance ...during the wrap condition. Figure 6 and Figure 7 illustrate the effect of a ring wrapping in ... FDDI. ...

...ork. A ring recovers from a station failure by *wrapping*.

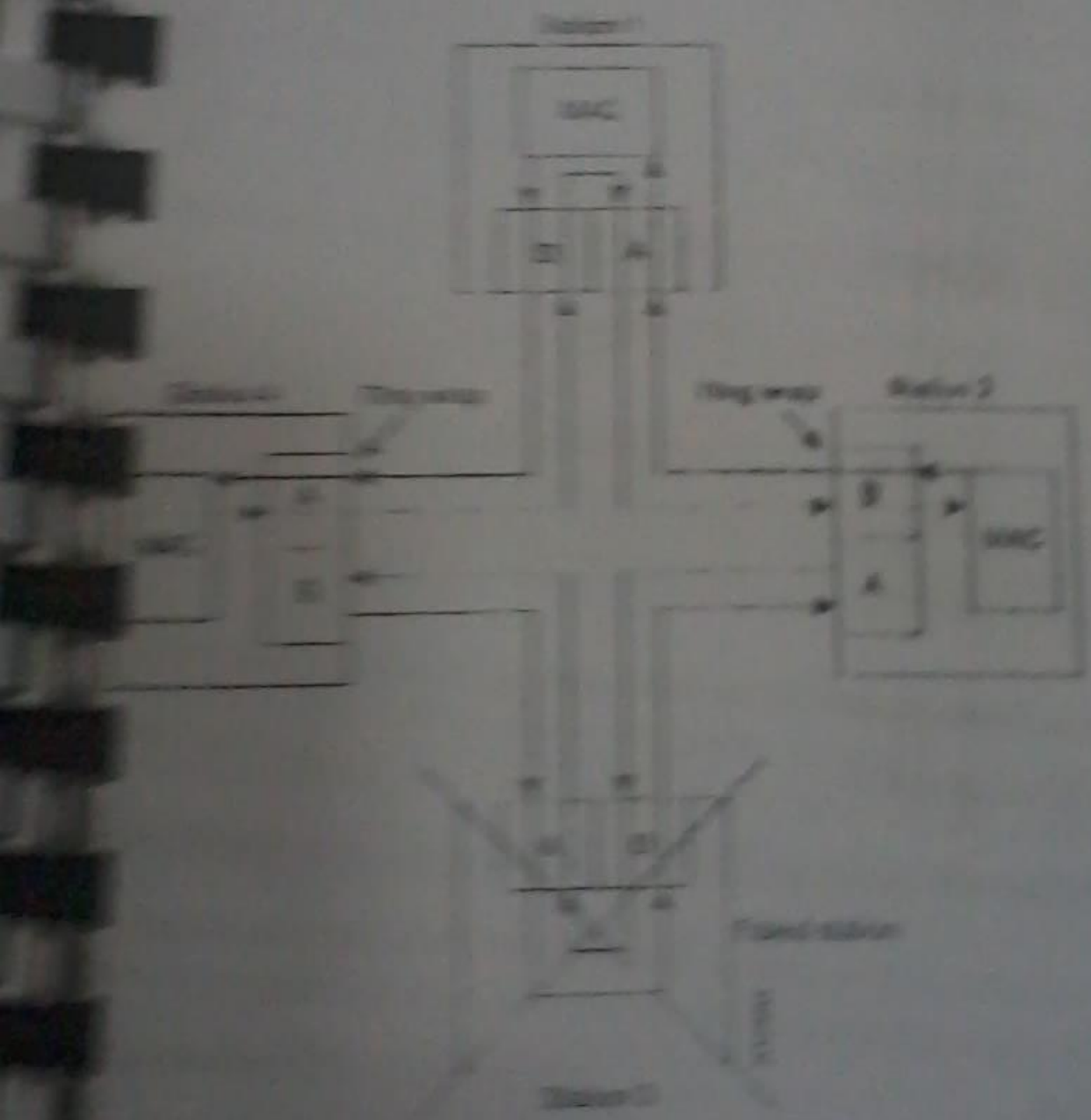
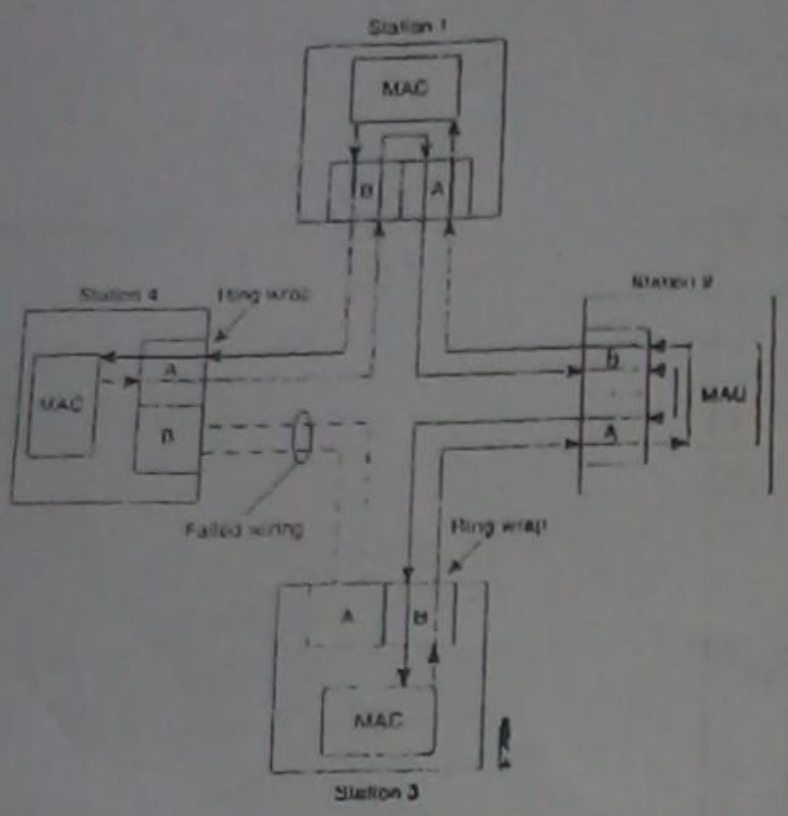


Figure 7: A ring also wraps to withstand a cable failure



When a single station fails, as shown in Figure 6, devices on either side of the failed (or powered down) station wrap, forming a single ring. Network operation continues for the remaining stations on the ring. When a cable failure occurs, as shown in Figure 7, devices on either side of the cable fault wrap. Network operation continues for all stations.

It should be noted that FDDI truly provides fault-tolerance against a single failure only. When two or more failures occur, the FDDI ring segments into two or more independent rings that are unable to communicate with each other.

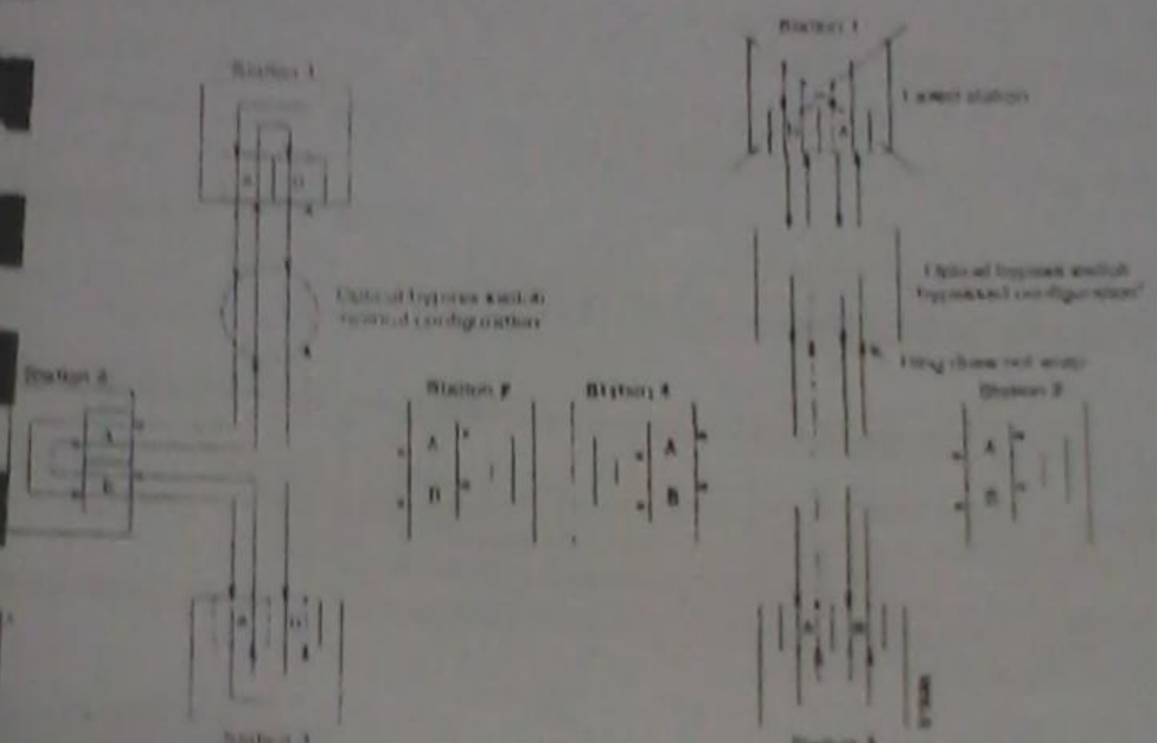
SESSION 2-4: OPTICAL BY PASS SWITCH

2-4.1 Optical Bypass Switch

An optical bypass switch provides continuous dual-ring operation if a device on the dual ring fails. This is used both to prevent ring segmentation and to eliminate failed stations from the ring. The optical bypass switch performs this function through the use of optical mirrors that pass light from the ring directly to the DAS device during normal operation. In the event of a failure of the DAS device, such as a power-off, the optical bypass switch will pass the light through itself by using internal mirrors and thereby maintain the ring's integrity. The benefit of this

capability is that the ring will not enter a wrapped condition in the event of a device failure. Figure 8 shows the functionality of an optical bypass switch in an FDDI network.

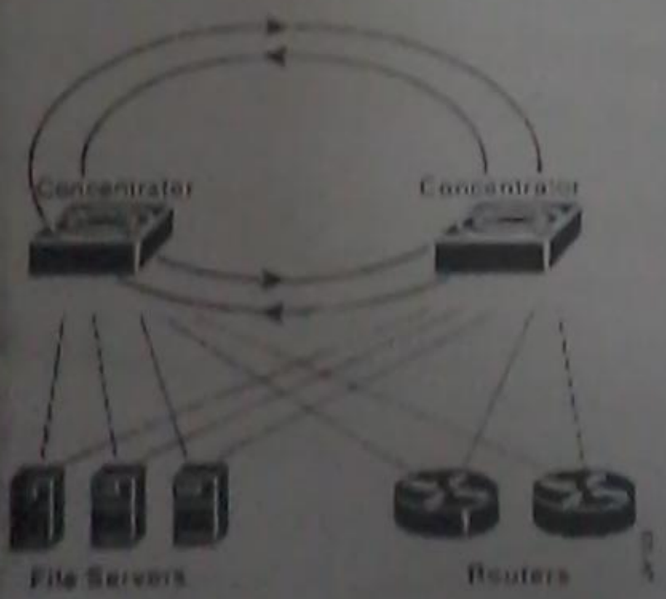
Figure 8: The optical bypass switch uses internal mirrors to maintain a network.



Dual Homing

Critical devices, such as routers or mainframe hosts, can use a fault-tolerant technique called *dual homing* to provide additional redundancy and to help guarantee operation. In dual-homing situations, the critical device is attached to two concentrators. Figure 9 shows a dual-homed configuration for devices such as file servers and routers.

Figure 9: A dual-homed configuration guarantees operation.

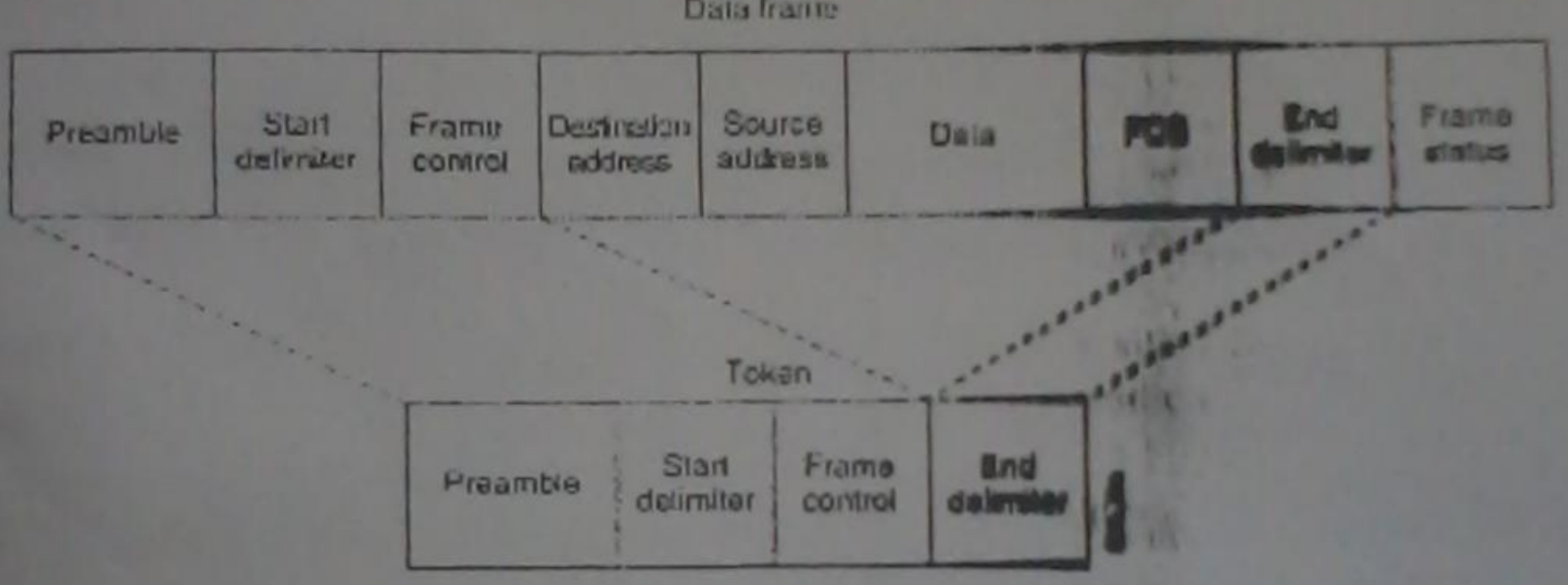


One pair of concentrator links is declared the active link; the other pair is declared passive. The passive link stays in back-up mode until the primary link (or the concentrator to which it is attached) is determined to have failed. When this occurs, the passive link automatically activates.

FDDI Frame Format

The FDDI frame format is similar to the format of a Token Ring frame. This is one of the areas where FDDI borrows heavily from earlier LAN technologies, such as Token Ring. FDDI frames can be as large as 4,500 bytes. Figure 10 shows the frame format of an FDDI data frame and token.

Figure 10: The FDDI frame is similar to that of a Token Ring frame.



FDDI Frame Fields

The following descriptions summarize the FDDI data frame and token fields illustrated in Figure 10.

- *Preamble*---A unique sequence that prepares each station for an upcoming frame.
- *Start Delimiter*---Indicates the beginning of a frame by employing a signalling pattern that differentiates it from the rest of the frame.
- *Frame Control*---Indicates the size of the address fields and whether the frame contains asynchronous or synchronous data, among other control information.
- *Destination Address*---Contains a unicast (singular), multicast (group), or broadcast (every station) address. As with Ethernet and Token Ring addresses, FDDI destination addresses are 6 bytes long.

- *Source Address*---Identifies the single station that sent the frame. As with Ethernet and Token Ring addresses, FDDI source addresses are 6 bytes long.
  Data---Contains either information destined for an upper-layer protocol or control information.
  Frame Check Sequence (FCS)---Filed by the source station with a calculated *cyclic redundancy check* value dependent on frame contents (as with Token Ring and Ethernet). The destination address recalculates the value to determine whether the frame was damaged in transit. If so, the frame is discarded.
- *End Delimiter*---Contains unique symbols, which cannot be data symbols, that indicate the end of the frame
- *Frame Status*---Allows the source station to determine whether an error occurred and whether the frame was recognized and copied by a receiving station.

per Distributed Data Interface (CDDI)

per Distributed Data Interface (CDDI) is the implementation of FDDI protocols over sted-pair copper wire. Like FDDI, CDDI provides data rates of 100 Mbps and uses a dual- ing architecture to provide redundancy. CDDI supports distances of about 100 meters from ktop to concentrator.

DDI is defined by the ANSI X3T9.5 Committee. The CDDI standard is officially named the *isted-Pair Physical Medium Dependent* (TP-PMD) standard. It is also referred to as the *Twisted-Pair Distributed Data Interface* (TP-DDI), consistent with the term *Fiber-Distributed ta Interface* (FDDI). CDDI is consistent with the physical and media-access control layers efined by the ANSI standard.

e ANSI standard recognizes only two types of cables for CDDI: shielded twisted pair (STP) nd unshielded twisted pair (UTP). STP cabling has 150-ohm impedance and adheres to IA/TIA 568 (IBM Type 1) specifications. UTP is data-grade cabling (Category 5) consisting of ur unshielded pair using tight-pair twists and specially developed insulating polymers in lastic jackets adhering to EIA/TIA 568B specifications.

igure 11 illustrates the CDDI TP-PMD specification in relation to the remaining FDDI pecifications.

which
DNS name server
of an IP address, the
The resolver then access one

may run on

reasons of availability, (a

INTRODUCTION TO

Introduction

In this unit, we are going to talk about

Learning Objectives

After reading this

- State the
  web
- Evaluate
- Publish
- Explain
  web-published
- Describe
  software
- Evaluate
- State some
  software

Unit content

SESSION 1-5: INTRODUCTION TO

1-5.1 Resources

1-5.2 Web Application Architecture

1-5.3 Web Design, Publishing and Testing

SESSION 2-5: INTRODUCTION

2-5.1 HTML

2-5.2 HTML

SESSION 1-5: HTML AND WEB DESIGN

1-5.1 Resources

In the early days of the Internet, there was no unifying way of referring to information resources on the Internet. The Uniform Resource Locator (URL) standard changed that, by defining a standard means of specifying any kind of resource on the Internet. A URL (sometimes known as a Uniform Resource Identifier, URI) can be split initially into two parts: a *scheme* and a *scheme-specific part*.

The scheme of a URL defines the method by which the resource is to be accessed. Example schemes are `http', `ftp', `mailto' and `telnet' (for access by HTTP, FTP, electronic mail, and TELNET respectively).

The form of the scheme-specific part of a URL depends on the scheme being used, but it in general will include the name of a server machine, and often the location of a particular resource (directory path and file) on that machine.

For example, consider the URL *http://www.anon.org/about/index.html*. This can be broken into four parts:

- The scheme `http', saying that the resource should be accessed using HTTP
- The server on which the resource can be found: `www.anon.org'
- A directory in which the resource is located: `about/'
- A filename within that directory: `index.html'

Scheme-specific parts are similar. In addition, information such as usernames, passwords, TCP port numbers, and labels may be included. For example, consider the two URLs *ftp://bob:sesame@ftp.blob.ac.uk/pub/* and *http://www.whoosh.org:8000/index.html#about*. The first may be broken down as follows:

ftp	the scheme (File Transfer Protocol)
bob	a username (separated from the server name by an at sign
sesame	a password (separated from the username by a colon
ftp.blob.ac.uk	the server name
pub/	a directory name

8000	the TCP port number (separated from the server name by a colon)
index.html	file name
About	a l...el within the file (separated from ...e filename by a hash sign)

When designing webpages, most of the URLs we use will use the 'http' scheme, but any of the other schemes may also specify an Internet resource.

RFC-822

Way back when the Internet was being used for only a few services, electronic mail among them, standards were developed for the format of electronic mail messages. RFC-822 (defined by the Internet Engineering Task Force) was the standard which specified what format these messages should take. Although since obsoleted by RFC-2822, the format of electronic mail messages remains the same: a set of headers, followed by a message. The header format defined in RFC-822 is used in a variety of other application protocols (including HTTP, which is used on the Web) and so we consider it briefly here.

RFC-822 specifies that an email message consist of a number of headers, each on a line on its own, followed by a blank line, followed by the message body. The headers themselves follow a specific format - the name of the header (which consists only of printable non-space characters, including the colon) followed by a colon, followed by the value of the header.

An example of an email message in RFC-822 format follows:

```
To: someone@example.com
Cc: someone-else@example.org
Subject: mints
Date: Tue, 01 Jan 2002 12:34:56 -0100

Good party, but I could do with some caffeine.  Do you have any of these caffeinated mints left?
```

Some types of field (for example the Date field) have a specified format they must follow, others (for example the Subject field) are unstructured and may contain arbitrary information.

While RFC-822 was originally developed to support the interchange of email messages, its use now extends too many other application-layer protocols, including HTTP, NNTP, and others.

Multipurpose Internet Mail Extensions

The Multipurpose Internet Mail Extension (MIME) standard was originally developed as a means by which binary data could be sent by electronic mail. The original mail protocol (SMTP) specified that only 7 bits of the information sent over a mail connection needed to be conserved - it was not 'eight-bit clean'. The MIME standard specified means of encoding this kind of binary information (including text in non-ASCII encodings, executable programs, images, sounds, and so on) and a means by which these could be attached to standard electronic mail messages.

As part of these provisions, MIME defines a set of *MIME types* - a method of labeling the attached information to indicate whether it is (for example) an image, a sound, or whatever. MIME defines main types, and requires that a *subtype* (or format) is also specified. The table below summarizes some of the available MIME types.

Type	Subtype	Contents
text	plain	text in plain ASCII format
	html	text encoded in HTML
image	png	an image in PNG format
	gif	a GIF image
	jpeg	an image in JPEG format
audio	basic	audio in 'au' format
	mpeg	MPEG audio information
	midi	audio in MIDI format
video	mpeg	a video clip in MPEG format
	quicktime	a 'QuickTime' movie
application	postscript	a PostScript document
	msword	a document in MS-Word format
	zip	ZIP-compressed data
	pdf	a document in PDF
	excel	an Excel-format spreadsheet

Those shown above are only a subset of the full number of formats which MIME supports - additional official types and subtypes exist, and a number of unofficial extensions are also in use. Additional information may be specified for some types, for example the character set used for text may be specified.

MIME types are used extensively on the Internet, and especially on the WWW, as a means by which the method of 'displaying' may be determined. MIME types are given in the form *type/subtype*, for example: text/html, image/png and so on.

Self Assessment 1-5

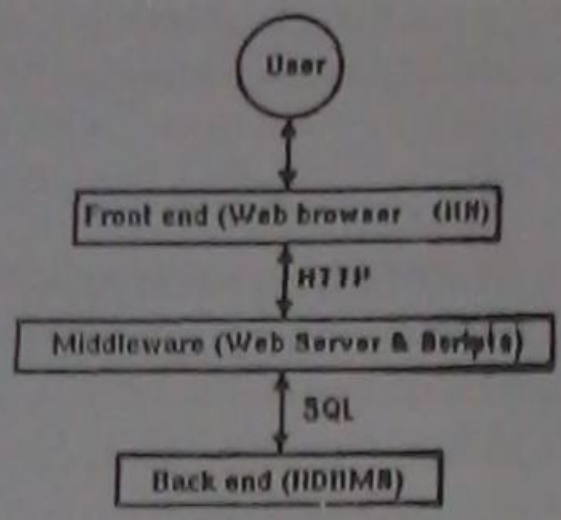
1. What is the difference between the Internet and the World Wide Web?
2. Explain the process that takes place when a user types a URL into a browser
3. Describe the interaction which an HTTP client and server have once a TCP connection has been established.
4. What is the difference between client and server side scripts?
5. Where would you go to find an online copy of the HTML standard?
6. If a bookshop launches a website to sell mail order copies of books, how would the functionality required is divided between the client-side and the server-side? Would any software beyond the web browser and web server be involved?
7. In the HTTP examples in the notes, identify the different kinds of headers in the requests and responses.

1-5.2 Web Application Architecture

Applications running over the web can generally be broken into three parts:

- User interface
- Business logic
- Data store

The user interface (or front end) is the user's web browser, the business logic (the middleware) is implemented by server-side programming of some kind and the data store (back end) is most often implemented using a relation database system (RDBMS). This is illustrated below:



This, in several variants, is currently a popular division of functionality for applications in general, and is also known as the 3-tier architecture.

A related architecture is the Model-View-Controller (MVC) architecture - again, the functionality is split more-or-less into three parts. In the MVC architecture, the split is a little different. An underlying data store (the model) can be changed by the user through the controller, while any changes in the model force the model to notify any components registered as a view. While there are some parallels between MVC and the 3-tier model, they are not the same.

The holy grail of web development, especially in large applications, is to separate the presentation details from the business logic and data store, so that the web designers, concerned mainly with the user interface and user experience, can work more or less separately from the web developers - those who are writing the business logic or designing the data store.

5.3 Web Design, Publishing and Serving

Overview

There are three main steps between having an idea for a web site and that site being realized as it transferred to users' browsers. In this unit we will examine each of these. Firstly, we will examine some of the basic tenets of web design and testing, including means of production of content and testing methods. Secondly, we will briefly look at how this content is uploaded the web server from where it will be available. Finally, we will describe some of the features which web server software might have, and what they are used for, as well as examining some of the most popular web server software.

c Design Principles

re setting finger to keyboard, or mouse, and starting to create web pages, it is probably l to go over some of the issues involved in designing web content. There is a temptation to o and make pages full of links to everywhere, full of flashing text, large full-colour images, and ground sounds. Before you do any of these things, though, bear in mind the following:

ontent is king

Long term, web users will visit and revisit your page for its content, not because they particularly like the font you use, or because your puce background happens to be one of their favourite colours.

bility is important

Like any software system, the interface you present to your users can make your site a pleasure to use (and therefore often-visited) or a pain to look at and navigate, relegating it to a quiet backwater of the web. If there are another ten websites providing similar content to yours, it's likely to be the usability which is the key deciding factor of your traffic.

andwidth is limited

Not everyone sees your pages arrive at the same speed you do. Browsing pages off local disk, even a flopp may be substantially faster than the site as experienced by a user on the other end of a slow modem connection. Nobody will wait for your site to load if similar ones which load faster are only a click away.

e providing the con nt which will bring your visitor back is really down to you, and eeping the volume of data as low as possible is down to intelligent use of tools and data ats, usability covers a range of things, and here we will look at a number of issues affecting sability.

ember, on a commercial site, visitors and repeat visitors are where your revenue comes from you can't afford to ignore your customers, especially when your competition is metaphorically next door.

kob Nielsen's site, http://www.useit.com/ is an excellent source of usability studies and mmendations, for IT in general as well as for the web.

Readability

As in normal writing, it is important that web pages are both clear and concise, and that any important points are drawn to the attention of the reader. Unnecessary clutter should be avoided (use a clear layout) and write your HTML so that the logical structure of the content is obvious. A consistent layout over a set of pages also avoids distracting a reader away from the informational content of the pages. These are mainly points which any writer (not just for the web) should be aware of.

When designing web pages, however, other issues also have to be considered. For example, when using colours, ensure that combinations of foreground/background colours do not make the text difficult to read (check on a monochrome screen too), and avoid making too much use of emphasized text or too many different fonts, since overuse of these can clutter a page and distract the reader. Links within the text are to be encouraged, although again overuse can distract, since links are normally emphasized in some way.

Various studies have found that web users tend to scan for text, rather than fully read each page. Good (ie appropriate but limited) use of emphasis and links will make this easier for the user; bad use will make it more difficult.

Avoid splitting a single topic across several pages - having to jump between consecutive texts on the same topic is distracting; each page should stand on its own. Similarly, within a page, ensure that related information is grouped together visually.

Finally, nothing is more annoying on the web than pages full of poor grammar, misspelt words, and no means of correcting them. Check your pages for spelling mistakes, and proofread them, since spelling checkers will not find all errors, before publishing them. Finally, make sure every page has a link to the author, so that readers can let them know if they have problems with a particular page.

Navigability

Navigability refers to how easy it is for a reader of your WWW pages to navigate your pages while not getting `lost in hyperspace'. One way of improving the navigability of your site is to storyboard' it, by designing a link structure suitable to the audience and content. Some sites may e best implemented as a hierarchical structure, while others may be best implemented as a linear t of pages. Having a completely hyperlinked (fully meshed) web site means that a user can mp from any page to any other, but can be very confusing in anything but the most simple of es.

To avoid the problems of users getting lost in your website, you should not include too many links in a single page, and should always use links with some form of meaningful text; avoid at all costs using meaningless phrase like 'click here' as the text for a link, since this emphasised text will draw the reader's attention. Link menus (lists with each item being a hyperlink) are one way of providing easy navigability, as long as the number of links does not grow too large.

Links to 'next', 'previous' and 'up' can be useful in a hierarchical/linear website model, and some sites provide a visual cue letting users know in which part of a web site they are currently browsing. Avoid splitting your pages into too few or too many pages - either of these can result in confusion about what information is found by following what link - splitting well-defined topics into separate pages can help this. Try to avoid the words 'forward' and 'back' within your links, since most browsers have forward and back buttons which relate to the user's view of how they are browsing the web, which may be very different from your expectation of where forward and back are.

Finally, check your links. Firstly, check that all the links work - there are some automated tools to do this for larger sites. Check also, that the names of links and files you have used are not operating-system dependent - filenames to be used on the web, for example, should not contain spaces, and some operating systems consider files with the same names in upper- or lower-case to be different. This is a very common error. Secondly, perform usability testing - let a user who has not seen your website before try to navigate it, note any problems they have, and attempt to rectify them. These checks should be done after the site has been uploaded to its eventual destination, since O/S dependencies will become apparent only then.

There are a variety of ways to enhance the navigability of a web site, but the one of the most helpful is to have some graphical or textual representation of the structure of the site. Keep such a representation clear, however. It need not be an extremely complicated image, and even a hierarchical list of the separate sections within the site will help.

Accessibility

Accessibility refers to making your web pages accessible to as broad a section of people as possible. There are wide ranges of ways in which web pages can be inaccessible to groups of users. Very large files or a large number of images, for example, can be virtually inaccessible for those working over slow or unreliable modem links.

Images in general are not accessible for users of text-only browsers (many web surfers use these for speed), blind users, or (perhaps the most important visitors to your site) search engine webcrawlers, which operate only on text. Providing a text alternative to such information is very useful, as for video or audio information, for the same reasons.

Don't make display assumptions - some users may browse with a 'portrait' rather than a 'landscape'-shaped browser, for example, or may be using a monochrome screen. Assuming that just because you browse one way everybody does can render your page almost unreadable to some users. Watch out for colour assumptions too - not all users use a colour display, and a non-negligible proportion of those who do are colour blind.

Also don't make browser-specific assumptions - your pages should be accessible to users of all browsers, not 'Best viewed with BrowserX 6.2'. Some browsers don't support images, some don't support style sheets, some don't support frames, many don't support scripting, and your content should be accessible from any of these. Checking your pages when viewed with different browsers should identify some of these problems. Avoid browser-specific HTML extensions for the same reasons. HTML validators are available which can pick up errors like this.

In today's world, more and more users are accessing the web from devices other than a PC - for example handheld computing platforms, mobile phones, set-top boxes. Accessibility solutions should keep these kinds of user in mind as well as (for example) users with disabilities.

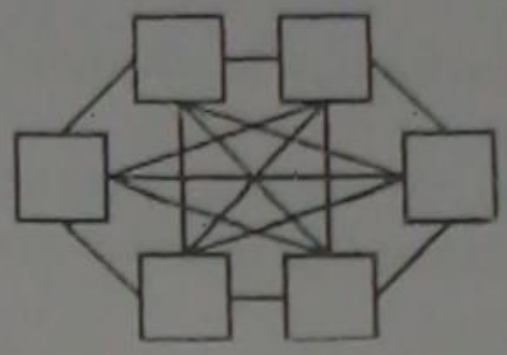
It is sometimes useful to provide a non-hypertext version of some information, especially if it is of the type which people may wish to print out. If an informative section of your website can be fitted into a 10-page PDF or PostScript document, rather than twenty separate HTML pages, provide the single document too.

Perform accessibility checking of your web page. The W3C's Web Accessibility Initiative (at http://www.w3.org/WAI/) provides guidelines and a checklist for HTML authors wishing to make their pages fully accessible. In addition, they provide a accessibility checker, called Bobby, validator, and many other HTML validators will pick up on accessibility issues. Also, check http://www.anybrowser.org/ for the 'Campaign for a Non Browser-Specific WWW', who include further pointers.

Finally, if your website is directed towards an international audience, consider providing translations of parts or all of it, for those who do not speak English. While it may seem to many that the WWW is purely Anglophone, searching for words like 'et' (French), 'und' (German) or even 'kaj' (Esperanto) using AltaVista shows that there is already a vast quantity of pages (and audience) using other languages.

Storyboarding - Web Site Structures

The most common navigation structures used within a web site are fully-linked, linear, and hierarchical.



While this allows quick navigation from one page to another, it can be confusing for inexperienced (or even experienced, in extreme situations) web users.

At the opposite end of the complexity spectrum, we have a linear structure, mirroring exactly that found in traditional media, such as books and articles. Each page is linked to the one before it, and the one after. Navigation is extremely easy, but loses in terms of flexibility.

A more often used structure is a hierarchical structure. This uses the human tendency to classify things as an aid to navigating the web site structure, and extends well to more than a single level.

SESSION 2-5 HTML INTRODUCTION

2-5.1 What is HTML?

HTML is a language for describing web pages

- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- HTML uses **markup tags** to describe web pages

HTML Tags

HTML markup tags are usually called HTML tags

- HTML tags are keywords surrounded by **angle brackets** like <html>
- HTML tags normally **come in pairs** like and
- The first tag in a pair is the **start tag**, the second tag is the **end tag**
- Start and end tags are also called **opening tags** and **closing tags**

HTML Documents = Web Pages

- HTML documents **describe web pages**
- HTML documents **contain HTML tags** and plain text
- HTML documents are also **called web pages**

The purpose of a web browser (like Internet Explorer or Firefox) is to read HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page: Example below

```
<html>
<body>

<h1>My first Heading</h1>

<p>My first paragraph</p>

</body>
</html>
```

Example Explained

- The text between <html> and </html> describes the web page
- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between and is displayed as a paragraph

What You Need

You do need a **text editor** to learn HTML. However,

- You don't need a web server
- You don't need a web site

Editing HTML

In our lessons we use a plain text editor (like Notepad) to edit HTML. We believe this is the best way to learn HTML. However, professional web developers often prefer HTML editors like FrontPage or Dreamweaver, instead of writing plain text.

HTM or HTML *Extension*?

When you save an HTML file, you can use either the .htm or the .html extension. We use **.html** in our lessons.

2-5.2 HTML Basic

HTML *Headings*

HTML headings are defined with the <h1> to <h6> tags.

Example

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

HTML *Paragraphs*

HTML paragraphs are defined with the <p> tag.

Example

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

HTML Links

HTML links are defined with the <a> tag.

Example

```
<a href="http://www.w3schools.com">This is a link</a>
```

Note: The link address is provided as an attribute.

HTML Images

HTML images are defined with the <img> tag.

Example

```
<img src="w3schools.jpg" width="104" height="142" />
```

Note: The name and the size of the image are provided as attributes.

HTML Elements

HTML documents are defined by HTML elements. An HTML element is everything from the start tag to the end tag:

Start tag *	Element content	End tag *
	This is a paragraph	
	This is a link	

* The start tag is often called the **opening tag**. The end tag is often called the **closing tag**.

HTML Element Syntax

- An HTML element starts with a **start tag / opening tag**
- An HTML element ends with an **end tag / closing tag**
- The **element content** is everything between the start and the end tag
- Some HTML elements have **empty content**
- Empty elements are closed **in the start tag**
- Most HTML elements can have **attributes**

Nested HTML Elements

Most HTML elements can be nested (can contain other HTML elements).

HTML documents consist of nested HTML elements.

HTML Document Example

```
<html>

<body>
<p>This is my first paragraph</p>
</body>

</html>
```

This example above contains 3 HTML elements.

Example Explained

The element:

```
<p>This is my first paragraph</p>
```

The element defines a paragraph in the HTML document.
The element has a start tag and an end tag
The element content is: This is my first paragraph

The <body> element:

```
<body>
<p>This is my first paragraph</p>
</body>
```

The <body> element defines the body of the HTML document
The element has a start tag <body> and an end tag </body>
The element content is another HTML element (a paragraph)

The <html> element:

```
<html>

<body>
<p>This is my first paragraph</p>
</body>

</html>
```

The <html> element defines the whole HTML document.
The element has a start tag <html> and an end tag </html>
The element content is another HTML element (the body)

Don't Forget the End Tag

Forgetting the end tag can produce unexpected results or errors.

Empty HTML Elements

HTML elements without content are called empty elements. Empty elements can be closed in the start tag.

 is an empty element without a closing tag (it defines a line break).

In XHTML and future versions of HTML, all elements must be closed.

Adding a slash to the start tag, like
, is the proper way of closing empty elements, accepted by HTML and XHTML.

Even if
 works in all browsers, writing
 instead is further proof.

HTML Tip: Use Lowercase Tags

The World Wide Web Consortium (W3C) **recommends** lowercase in HTML 4, and **demands** lowercase tags in future versions of (X)HTML.

HTML Attributes

Attributes provide additional information about HTML elements.

- HTML elements can have **attributes**
- Attributes provide **additional information** about the element
- Attributes are always specified in **the start/opening tag**
- Attributes come in name/value pairs like: **name="value"**

Attribute Example

HTML links are defined with the <a> tag. The link address is provided as an attribute:

Example

```
<a href="http://www.w3schools.com">This is a link</a>
```

Always Quote Attribute Values

Attribute values should always be enclosed in quotes. Double style(" ") quotes are the most common.

HTML Note: Use Lowercase Attributes

Attribute names and attribute values are case-insensitive.

However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation

Newer versions of (X)HTML will demand lowercase attributes.

HTML Attributes Reference

- Below is a list of some attributes that are standard for most HTML elements:

Attribute	Value	Description
class	*class_rule* or *style_rule*	The class of the element
id	*id_name*	A unique id for the element
style	*style_definition*	An inline style definition
title	*tooltip_text*	A text to display in a tool tip

HTML Headings

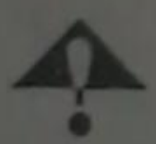
Headings are important in HTML documents.

Headings are defined with the <h1> to <h6> tags.

<h1> defines the largest heading. <h6> defines the smallest heading.

Example

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```



Note: Browsers automatically add an empty line before and after headings.

Headings Are Important

Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.
Search engines use your headings to index the structure and content of your web pages.
Since users may skim your pages by its headings, it is important to use headings to show the document structure.

H1 headings should be used as main headings, followed by H2 headings, then less important H3 headings, and so on.

HTML Rules (Lines)

The <hr /> tag is used to create an horizontal rule (line).

Example

```
<p>This is a paragraph</p>
<hr />
<p>This is a paragraph</p>
<hr />
<p>This is a paragraph</p>
```

HTML Comments

Comments can be inserted in the HTML code to make it more readable and understandable. Comments are ignored by the browser and are not displayed.

Comments are written like this:

Example

```
<!-- This is a comment -->
```

Note: There is an exclamation point after the opening bracket, but not before the closing bracket.

HTML Tip - How to View HTML Source

Have you ever seen a Web page and wondered "Hey! How did they do that?"

To find out, right-click in the page and select "View Source" (IE) or "View Page Source" (Firefox) or similar for other browsers. This will open a window that shows you the HTML code of the page.

HTML Tag Reference

- Additional information about these tags and their attributes.

Tag	Description
<html>	Defines an HTML document
<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
<hr />	Defines a horizontal rule
<!-->	Defines a comment

HTML Paragraphs

HTML documents are divided into paragraphs.

Paragraphs are defined with the <p> tag.

Example

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Note: Don't forget the end tag

HTML Line Breaks

Use the
 tag if you want a line break (a new line) without starting a new paragraph:

Example

```
<p>This is<br />a para<br />graph with line breaks</p>
```

The
 element is an empty HTML element. It has no end tag.

TML Tag Reference

Additional information about these tags and their attributes.

Tag	Description
<p>	Defines a paragraph
 	Inserts a single line break

-5.3 HTML Text Formatting

HTML Text Formatting

HTML Text Formatting

This text is bold

This text is big

This text is italic

This is computer output

This is subscript and superscript

HTML Formatting Tags

HTML uses tags like <b> and <i> for formatting output, like **bold** or *italic* text.

These HTML tags are called formatting tags.

Refer to the next page for a complete reference.

	Defines strong text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<ins>	Defines inserted text
<del>	Defines deleted text
<s>	Deprecated. Use <del> instead
<strike>	Deprecated. Use <del> instead
<u>	Deprecated. Use styles instead

"Computer Output" Tags

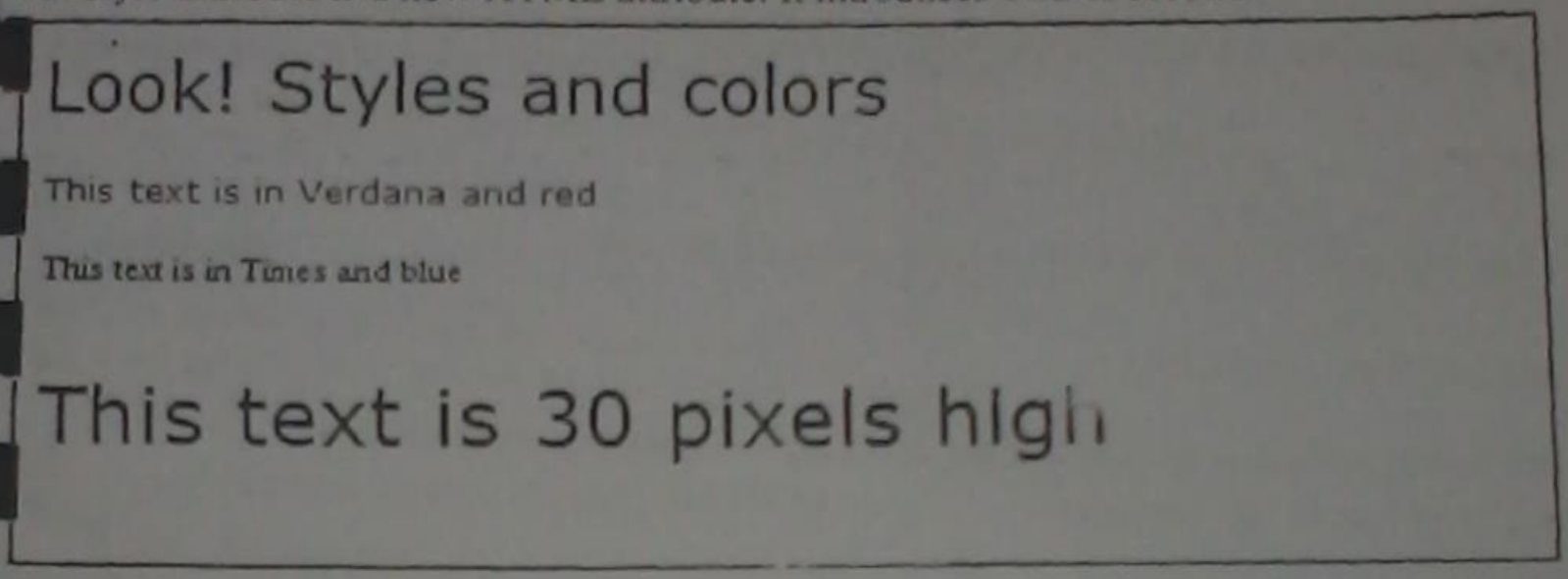
Tag	Description
<code>	Defines computer code text
<kbd>	Defines keyboard text
<samp>	Defines sample computer code
<tt>	Defines teletype text
<var>	Defines a variable
<pre>	Defines preformatted text
<listing>	Deprecated. Use <pre> instead
<plaintext>	Deprecated. Use <pre> instead
<xmp>	Deprecated. Use <pre> instead

Citations, Quotations, and Definition Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<bdo>	Defines the text direction
<blockquote>	Defines a long quotation
<q>	Defines a short quotation
<cite>	Defines a citation
<dfn>	Defines a definition term

HTML Styles

The style attribute is a new HTML attribute. It introduces CSS to HTML.



The HTML Style Attribute

The purpose of the style attribute is:

To provide a common way to style all HTML elements.

Styles were introduced with HTML 4, as the new and preferred way to style HTML elemer

With HTML styles. styles can be added to HTML elements directly by using the style attribu or indirectly in separate style sheets (CSS files).

HTML Style Examples

style="background-color:yellow"
style="font-size:10px"
style="font-family:Times"
style="text-align:center"
Style Examples:

Background Color

<body style="background-color:yellow">

The style attribute defines a style for the <body> element.

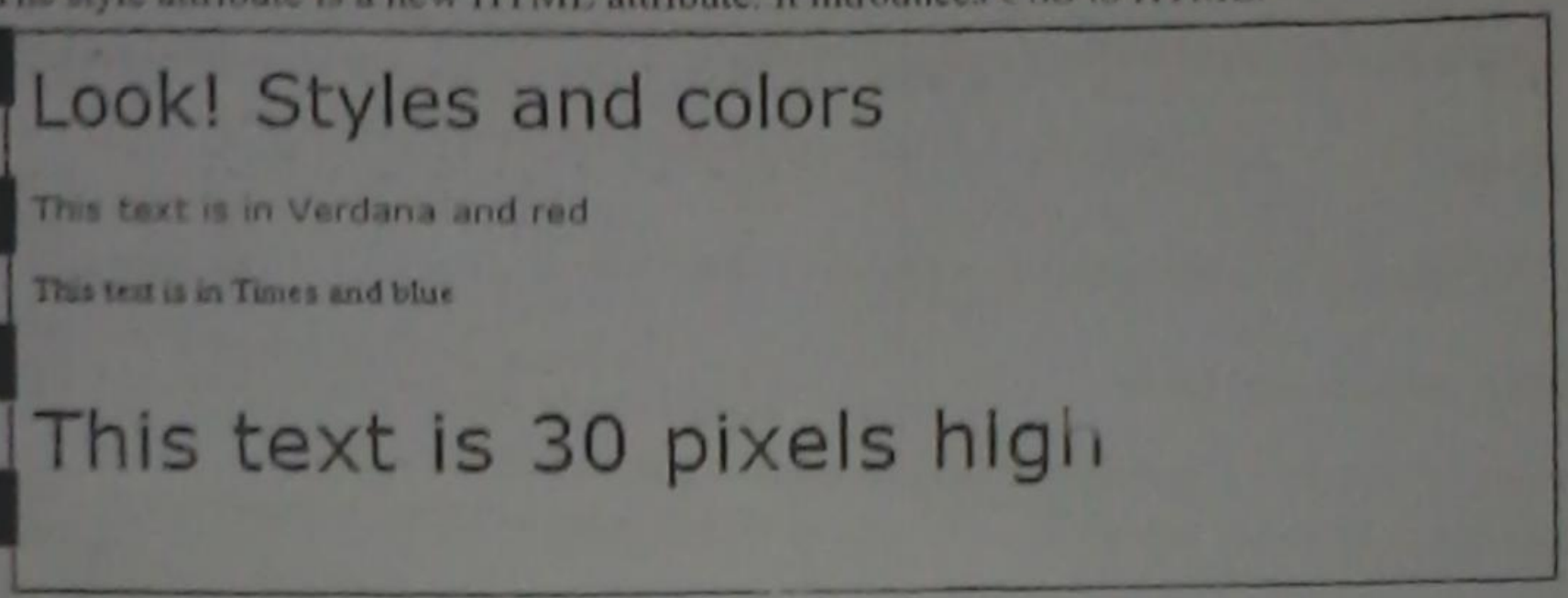
Font Family, Color and Size

<p style="font-family:courier new; color:red; font-size:20px">

The style attribute defines a style for the <p> element.

TML Styles

The style attribute is a new HTML attribute. It introduces CSS to HTML.



The HTML Style Attribute

The purpose of the style attribute is:

To provide a common way to style all HTML elements.

Styles were introduced with HTML 4, as the new and preferred way to style HTML elements. With HTML styles, styles can be added to HTML elements directly by using the style attribute or indirectly in separate style sheets (CSS files).

HTML Style Examples

style="background-color:yellow"
style="font-size:10px"
style="font-family:Times"
style="text-align:center"
Style Examples:

Background Color

<body style="background-color:yellow">

The style attribute defines a style for the <body> element.

Font Family, Color and Size

<p style="font-family:courier new; color:red; font-size:20px">

The style attribute defines a style for the <p> element.

Text Alignment

<h1 style="text-align:center">

The style attribute defines a style for the <h1> element.

HTML Links

A link is the "address" to a document (or a resource) on the web.

Hyperlinks, Anchors, and Links

In web terms, a hyperlink is a reference (an address) to a resource on the web.
Hyperlinks can point to any resource on the web: an HTML page, an image, a sound file, a movie, etc.

An anchor is a term used to define a hyperlink destination inside a document.

The HTML anchor element <a>, is used to define both hyperlinks and anchors.

We will use the term HTML link when the <a> element points to a resource, and the term HTML anchor when the <a> elements defines an address inside a document.

An HTML Link

Link syntax:

```
<a href="url">Link text</a>
```

The start tag contains attributes about the link.
The element content (Link text) defines the part to be displayed.

Note: The element content doesn't have to be text. You can link from an image or any other HTML element.

The href Attribute

The **href** attribute defines the link "address".

The target Attribute

The **target attribute** defines **where** the linked document will be opened.

The code below will open the document in a new browser window:

Example

```
<a href="http://www.w3schools.com/"
target="_blank">Visit W3Schools!</a>
```

The name Attribute

When the **name attribute** is used, the <a> element defines a named anchor inside a HTML document.

Named anchor are not displayed in any special way. They are invisible to the reader.

Named anchor syntax:

```
<a name="label">Any content</a>
```

The link syntax to a named anchor:

```
<a href="#label">Any content</a>
```

The # in the href attribute defines a link to a named anchor.

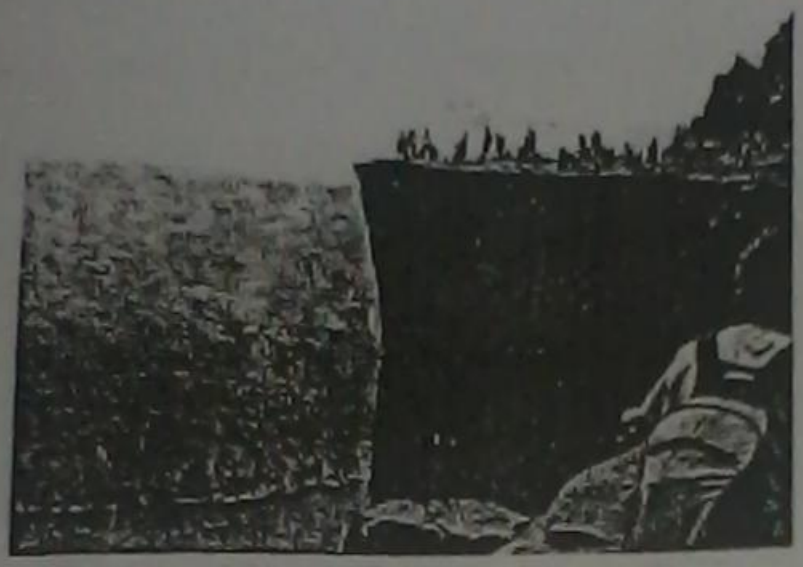
Link Tags

Tag	Description
<a>	Defines an anchor

2-5.4 HTML Images

Example

Norwegian Mountain Trip



In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only and it has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page.

The syntax of defining an image:

```
<img src="url' '>
```

The URL points to the location where the image is stored. An image named "logo.gif" located in the directory "images" on "www.knust.edu.gh" has the URL:

http://www.knust.edu.gh/images/logo.gif.

alternate text - alt.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

The Alt Attribute

The alt attribute is used to define an "alternate text" for an image. The value of the alt attribute is an author-defined text:

```
<img src="boat.gif" alt="Big Boat" />
```

The "alt" attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the "alt" attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers.

Image Tags

Tag	Description
<img>	Defines an image
<map>	Defines an image map
<area>	Defines a clickable area inside an image map

HTML tables

HTML Tables

Apples	44%
Bananas	23%
Oranges	13%
Other	10%

Tables

Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

Tables and the Border Attribute

If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

To display a table with borders, you will have to use the border attribute:

```
<table border="1">
<tr>
<td>Row 1, ce   1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

Headings in a Table

Headings in a table are defined with the <th> tag.

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

Heading	Another Heading
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

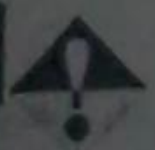
Empty Cells in a Table

Table cells with no content are not displayed very well in most browsers.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</t
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	



Note that the borders around the empty table cell are missing (NB! Mozilla Firefox displays the border).

To avoid this, add a non-breaking space ( ) to empty data cells, to make the borders visible:

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td> </td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, c 2
row 2, cell 1	

Basic Notes - Useful Tips

The <thead>,<tbody> and <tfoot> elements are seldom used, because of bad browser supp
Expect this to change in future versions of XHTML.

Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table
<thead>	Defines a table head
<tbody>	Defines a table body
<tfoot>	Defines a table footer

2-5.5 HTML Lists

HTML supports ordered, unordered and definition lists.

HTML Lists

- This is the first
- This is the second
- This is the third

Unordered Lists

An unordered list is a list of items. The list items are marked with bullets (typically small black circles). An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

Here is how it looks in a browser:

- Coffee
- Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

```
<ol>
<li>Coffee</li>
<li>Milk</li>
</ol>
```

Here is how it looks in a browser:

1. Coffee
2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

- ***Definition Lists***

A definition list is not a list of single items. It is a list of items (terms), with a description of each item (term).
A definition list starts with a <dl> tag (definition list).
Each term starts with a <dt> tag (definition term).

Each description starts with a <dd> tag (definition description).

```
<dl>
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Here is how it looks in a browser:

Coffee
    Black hot drink
Milk
    White cold drink

Inside the <dd> tag you can put paragraphs, line breaks, images, links, other lists, etc.

List Tags

Tag	Description
	Defines an ordered list
	Defines an unordered list
	Defines a list item
<dl>	Defines a definition list
<dt>	Defines a term (an item) in a definition list
<dd>	Defines a description of a term in a defin...on list
<dir>	Deprecated. Use instead
<menu>	Deprecated. Use instead

2.4.5 HTML Forms and Input

HTML forms are used to select different kinds of user input.

Forms

A form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the <form> tag.

```
<form>
.
input elements
.
</form>
```

Input

The most used form tag is the <input> tag. The type of input is specified with the type attribute. The most commonly used input types are explained below.

Text Fields

Text fields are used when you want the user to type letters, numbers, etc. in a form.

```
<form>
First name:
<input type="text" name="firstname" />
<br />
Last name:
<input type="text" name="lastname" />
</form>
```

How it looks in a browser:

First name:
Last name:

Note that the form itself is not visible. Also note that in most browsers, the width of the text field is 20 characters by default.

Radio Buttons

Radio Buttons are used when you want the user to select one of a limited number of choices.

```
<form>
<input type="radio" name="sex" value="male" /> Male
<br />
<input type="radio" name="sex" value="female" /> Female
</form>
```

How it looks in a browser:

Male
Female

Note that only one option can be chosen.

Checkboxes

Checkboxes are used when you want the user to select one or more options of a limited number of choices.

```
<form>
I have a bik
<input type="checkbox" name="vehicle" value="Bike" />
<br />
I have a car:
<input type="checkbox" name="vehicle" value="Car" />
<br />
I have an airplane:
<input type="checkbox" name="vehicle" value="Airplane" />
</form>
```

How it looks in a browser:

I have a bike:
I have a car:
I have an airplane:

The Form's Action Attribute and the Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to the server. The form's action attribute defines the name of the file to send the content to. The file defined in the action attribute usually does something with the received input.

```
<form name="input" action="html_form_submit.asp" method="get">
Username:
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

How it looks in a browser:

Username: Submit

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html_form_submit.asp". The page will show you the received input.

Form Tags

Tag	Description
<form>	Defines a form for user input
<input>	Defines an input field
<textarea>	Defines a text-area (a multi-line text input control)
<label>	Defines a label to a control
<fieldset>	Defines a fieldset
<legend>	Defines a caption for a fieldset
<select>	Defines a selectable list (a drop-down box)
<optgroup>	Defines an option group
<option>	Defines an option in the drop-down box
<button>	Defines a push button
<isindex>	Deprecated. Use <input> instead

Self Assessment 2-5

Find out your IP address when you are connected to the Internet

Visit the WWW Consortium web site, and find the latest HTML and CSS standards.

Use the telnet application to connect to port 80 of a directly accessible web server. Over this TCP connection, use HTTP to send a request to the server, and examine the response. Break the response into its component parts. You may need to turn on local echo in your telnet application to see what you are typing, and most web servers expect, at a minimum, the Host: HTTP header.

Also, some web server's close down the TCP connection fairly quickly - you may have to type and work quickly to save any information you see.

Use Notepad, or a similar text-only editor, to type in the HTML example in the notes. Save this file (as example.html for instance) and open it in a web browser. If you have more than one browser available, open it in both browsers and note any differences.

Using a URL for a page you are visiting, try changing the spelling of various sections of it. What error messages, if any, result in each case? Are these error messages being generated by the browser or the server?.

U

FEATURES AND TOOLS ASSOCIATED
INTE

Introduction

This unit is on tools and features associated with the internet. You will be taken thro
handle and use some of these tools.

Learning Objectives

After reading this unit you should be able to:

1. Identify all the tools and features associated with Inter
2. Handle and use some of these Tools

Unit content

SESSION 1-2: FEATURES AND TOOLS ASSOCIATED WITH THE INTERN

- 1-2.1 E-mail
- 1-2.2 Usenet news groups
- 1-2.3 File Transfer
- 1-2.4 Telnet

SESSION 2-2: FEATURES AND TOOLS CONT.

- 2-2.1 Gopher (Gopherspace)
- 2-2.2 WAIS (Wide Area Information Servers)
- 2-2.3 World Wide Web
- 2-2.4 Web browser

SECTION 1-2: SOME FEATURES AND TOOLS ON THE INTERNET

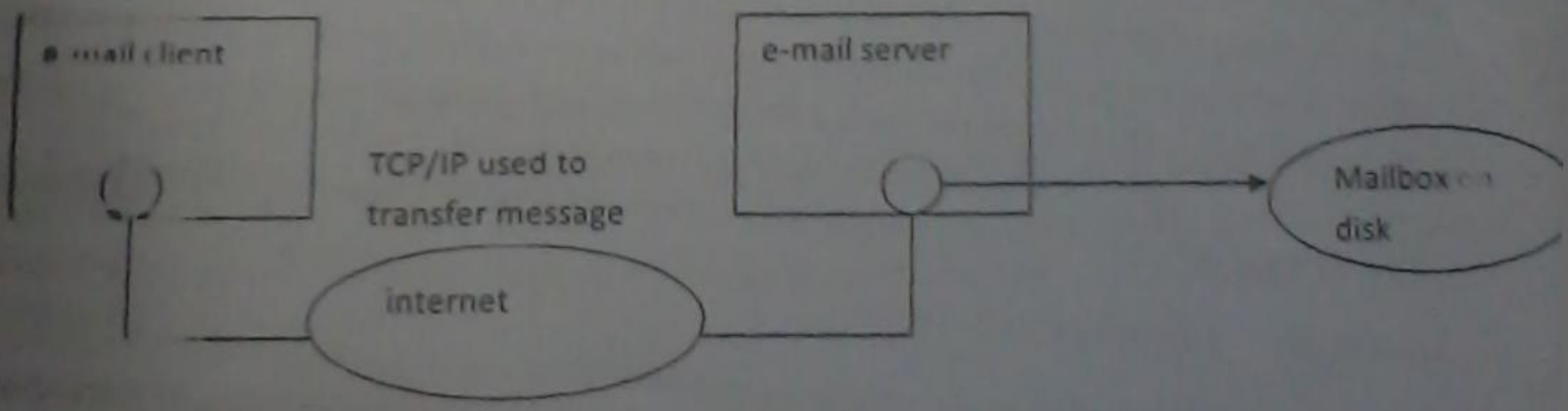
The principal features of the internet are e-mail, discussion groups, file transfer, remote access, and information searches (Gopher, WAIS, and WWW)

1.1 E-Mail

The World Wide Web is getting all the headlines, but for many people the main attraction of the internet is the electronic mail (e-mail). There are over 400 million users of e-mail in the world today. Out of this figure, about 60% are found in educational institutions. This e-mail is a service that permits one to send a memo to another person or a group. Electronic mail also permits one to reply to a memo.

How E-mail works

The computer communication always involves interaction between two programs called a client and a server. E-mail systems follow the client-server approach: two programs cooperate to transfer an e-mail message from the sender's computer to the recipient's mailbox (i.e. transfer requires two programs because an application running on one computer cannot store data directly in a mailbox on another computer's disk). When a user sends an e-mail message, a program on the sender's computer becomes the client. It contacts an e-mail server program on the recipient's computer and transfers a copy of the message. The server stores the message in the recipient's mailbox.



A client's software starts automatically as soon as a user finishes composing an e-mail message. The client uses the recipient's e-mail address to determine which remote computer to contact. The client uses TCP to send a copy of the e-mail message across the internet to the server. When the server receives a message, it stores the message in the recipient's mailbox and informs the recipient that e-mail has arrived.

1-2.2 Usonet news groups – electronic discussion groups.

One of the internet's most interesting features goes under the misleading name new -group although they don't have much to do with news. Actually, newsgroups are electronic discussio held by groups of people focusing on specific topics e.g. AOL's forums.

Usenet is a Public Access Network (PAN) of dispersed newsgroups that exchanges e-mail a messages. That is users post questions, answers, general information and FAQ files (Frequen Asked Questions) on Usenet. This FAQ is a document that contains questions and answ regarding a specific topic, technology or internet service

1-2.3 File Transfer

It is the use of File Transfer Protocol (FTP) for copying all the files one wants. Many Net us enjoy "FTPing" i.e. crushing the system and checking into some FTP sites for free interesti files to download (copying). FTPing is a method where you can connect to a remote compu and transfer publicly available files to one's own pc

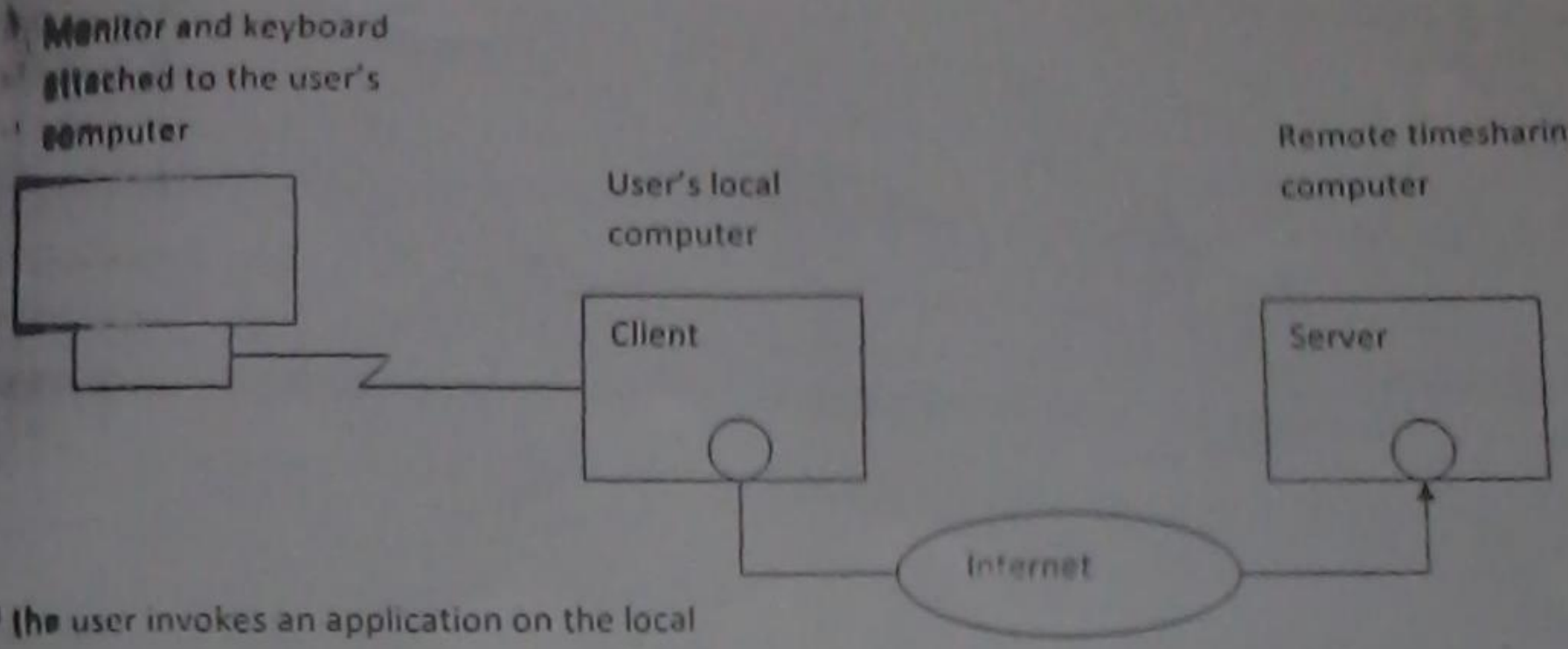
1-2.4 Telnet

It is a cooperative system that allows you to connect (log on) to remote computers. This feat enables you to tap into internet computers and public access files as though you are connec directly or l ally. It is especially useful for perusing large databases or library card catalogs. in short the TELNET is an internet standard remote log in service protocol. The TELN protocol specifies exactly how a remote login client and a remote login server interact.

How Remo Login (Telnet) works

Like other internet applications, remote login follows the client server paradigm. hen a user a local computer decides to log into a remote system, the user invokes a local applicat program for the remote login service and enters the name of the remote computer to contact. application becomes a client that uses TCP/IP to connect across the internet to a server on remote computer.

The server sends exactly the login prompt used for conventional terminals.



- The user invokes an application on the local computer. The local application connects the user's keyboard and displays the remote timesharing system

Once a connection has been established between the client and the server, the software allows the user to interact directly with the remote computer. When the user presses a key on the keyboard or moves a mouse, the client application sends the data across the connection to the remote computer. When the application program on the remote computer produces output, the server sends it back to the client.

1-2.5 Gopher (Gopherspace)

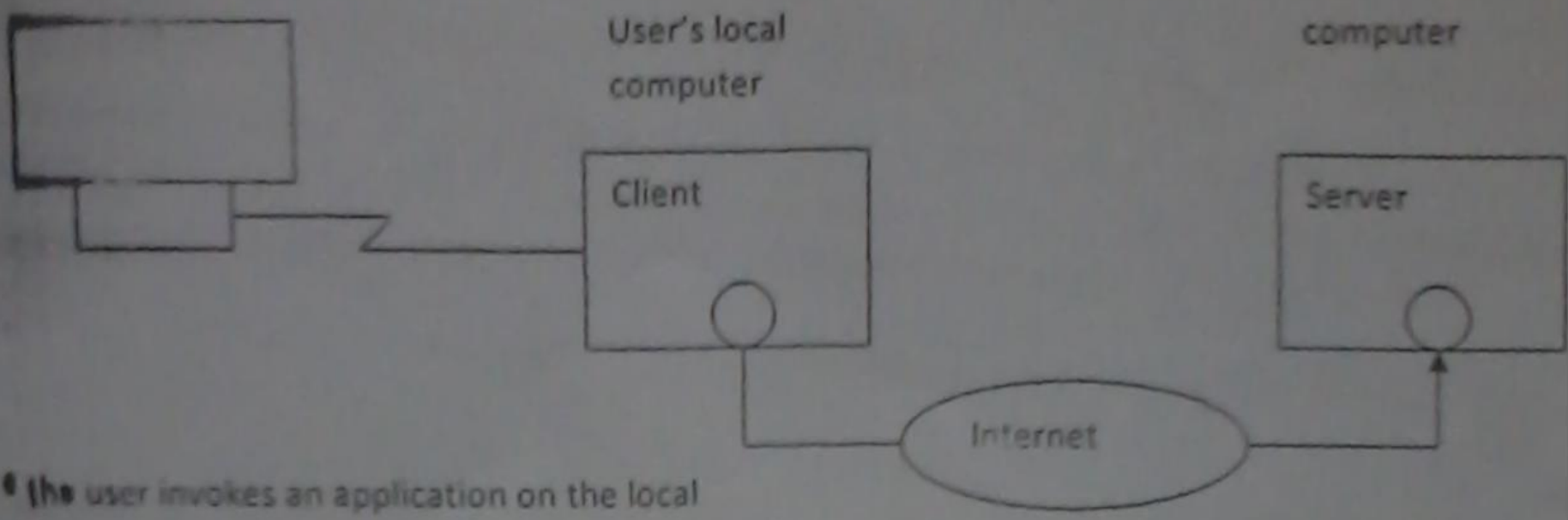
It is a uniform system of menus, or series of lists that allow users to easily browse through and retrieve files stored on different computers. It is called gopher because the first Gopher was developed at the home of Golden Gophers – University of Minnesota.

The Gopher also includes
VERONICA – Very Easy Rodent-Oriented Net-wide Index Computerized Archives.

1-2.6 WAIS (Wide Area Information Server)

It is a system for searching internet databases by subject, using specific words or phrases rather than sorting through a hierarchy of menus. WAIS is offered only by certain information sites (servers) and can be applied to only a limited number of files. WAIS – which was the only early automated search service that permits one to locate documents that contain key words or phrases, has been replaced by other search engines.

monitor and keyboard attached to the user's computer



- the user invokes an application on the local computer. The local application connects the user's keyboard and displays the remote timesharing system

Once a connection has been established between the client and the server, the software allows t user to interact directly with the remote computer. When the user presses a key on the keyboa or moves a mouse, the client application sends the data across the connection to the remo computer. When the application program on the remote computer produces output, the serv sends it back to the client.

1-2.6 Gopher (Gopherspace)

It is a uniform system of menus, or series of lists that allow users to easily browse through a retrieve files stored on fferent computers. It is called gopher because the first Gopher v developed at the home of Golden Gophers – University of Minnesota.

The Gopher also includes
VERONICA – Very Ea Rodent-Oriented Net-wide Index Computerized Archives.

1-2.6 WAIS (Wide Area Information Server)

It is a system for searching internet databases by subject, using specific words or phrases r than sorting through a hierarchy of menus. WAIS is offered only by certain information (servers) and can be applied to only a limited number of files. WAIS – which was the only automated search service that permits one to locate documents that contain key words or phr has been replaced by other search engines.

1-2.7 World Wide Web

The WWW is an internet service that organizes information using hypermedia. Each docum can contain embedded reference to images, audio, or other documents. The WWW or simply "WEB" consists of an interconnected system of sites or places, all over the world that can st information in multimedia form – sound, photos, video as well as text.

The most popular internet browsing service uses hypermedia known as WWW, the service mechanism that links together information stored on many computers. In essence, www allo the references in a document on one computer to refer to text on another computer.

In addition to containing textual information, WW documents can contain sounds and graphi images.

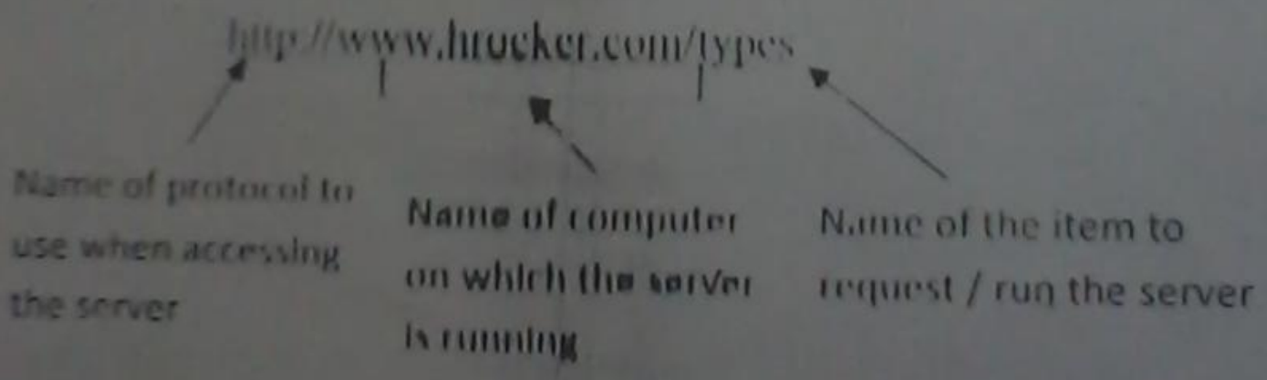
How the WWW works

The WWW uses client-server interaction. The browser program acts as a client that uses internet to contact a remote server for a copy of the requested page. The server on the rem system returns a copy of the page along with additional information – at this stage it tells browser two (2) important things

1. It describes how to display the information
2. It gives a URL, for each selectable item on the page.

When the browser receives a page from a remote server, it displays the page and then waits the user to select one of the highlighted items. Once the user makes a selection, the brow consults the hidden information that arrived with the page top find the URL (Uniform Resou Locator) that corresponds to the selection. The browser then uses the internet to obtain the ne selected page of information.

E.g. of URL



WAIS deal with text, the web provides information in multimedia form – it contains graph video and audio as well as text.

Whereas Gopher is a menu-based approach to accessing Net resources, the web uses a hypertext format. Hypertext is a system in which documents scattered across many internet sites are directly linked, so that a word or phrase in one document becomes a connection to an entirely different document. The format used on the web is called HTML (HyperText Markup Language) and swaps information using HTTP (Hyper Text Transfer Protocol).

SESSION 2-2 TERMS

Website: is a file stored on a computer (server or host computer)

Homepage: is the main page or first screen you see when you access a website. It is a page of information accessible through www that is the main page for an individual. A homepage usually includes references to other pages.

Web browser: Software that helps you get information you want by clicking your mouse pointer on words or pictures on the screen.

Examples of web browsers

- Netscape Navigator
- Internet webcrawler
- Microsoft Internet explorer
- NCSA Mosaic
- Mozilla Firefox

URL: The URL (Uniform Resource Locator) is an address that points to a specific resource on the web. To find a particular website (homepage) you need its URL. It is a short character string used by browsers to identify a particular page of information on the www.

Unit 1

INTRODUCTION TO INTERNET TECHNOLOGY

INTRODUCTION TO ALGORITHMS

This unit introduces you to the internet. You will be given the definition of the internet and shown how to connect to the internet.

Learning Objectives

After reading this unit you should be able to:

1. Explain what an Internet is and describe how to use and surf through the Internet
2. Explain the various terms associated with the Internet Technology.

Unit content

SECTION 1-1: INTERNET

1-1.1 Definition of Internet

1-1.2 Connecting to the Internet

SECTION 2-1: TERMINOLOGIES ASSOCIATED WITH THE INTERNET TECHNOLOGIES.

2-1.1 Internet Addresses

2-1.2 Domain Name

2-1.3 Domain Name Systems/Severs (DNS)

2-1.4 Name Resolution

SESSION 1-1: INTERNET

1-1.1 Definition of Internet

Is the collection of networks and routers that use the Transmission Control Protocol/Intern Protocol (TCP/IP) suite and function as a single, large network. The internet reache government, commercial and educational institutions around the world. The TCP/IP literally the name of protocols that specify how computers communicate on the internet. Informally, t name refers to the software that implements the protocols. All computers that use the intern need TCP/IP software.

Internet Is the world's biggest network which uses a protocol called TCP/IP to allo computers to communicate. Users can connect to the internet through direct connections, onl information services and internet service providers. It is therefore the "mother of all networks The Net is an international network connecting more than 36,000 smaller networks.

History

It was created by the UN department of Defense in 1969 under the then name (ARPAnet). ARPA was the department's Advanced Research Project Agency. It was built to serve tv purposes:

- The first was to share research among military, industry and university sources.
- The second was to provide a system for sustaining communication among military u in event of any nuclear attack.

With the many different kinds of computers being connected, engineers had to find a wa speak the same language. The solution developed was TCP/IP, the standard since 1983 and standard language of the internet.

TCP/IP, for Transmission Control Protocol/Internet Protocol, is the standardized set guidelines (protocols) that allow different computers on different networks to communicate each other efficiently, no matter how they gained access to the Net.

1-1.2 Connecting To the Internet

There are three ways to connect your PC with the internet.

- **Through Direct method**

Through school or public workplaces. Many universities, colleges and large businesses dedicated, high-speed phone lines that provide a direct connection to the internet

Advantage. – Cheaper

- **Through online information services**

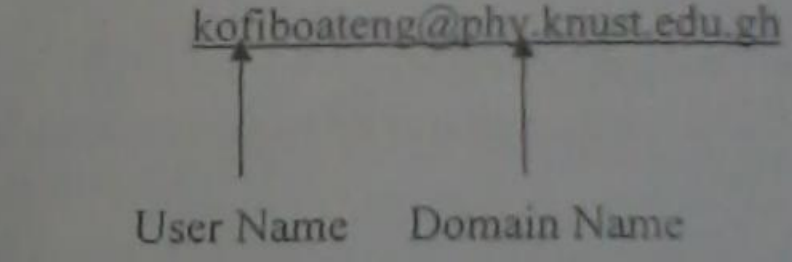
That is subscribing to a commercial online information service, which provides you with its own communication software, e.g. Africa On-Line, AOL, CompuServe. They provide an electronic "gateway" to the internet. Although very cheap, they do not always provide complete Internet Services.

- **Through Internet Service Providers (ISPs)**

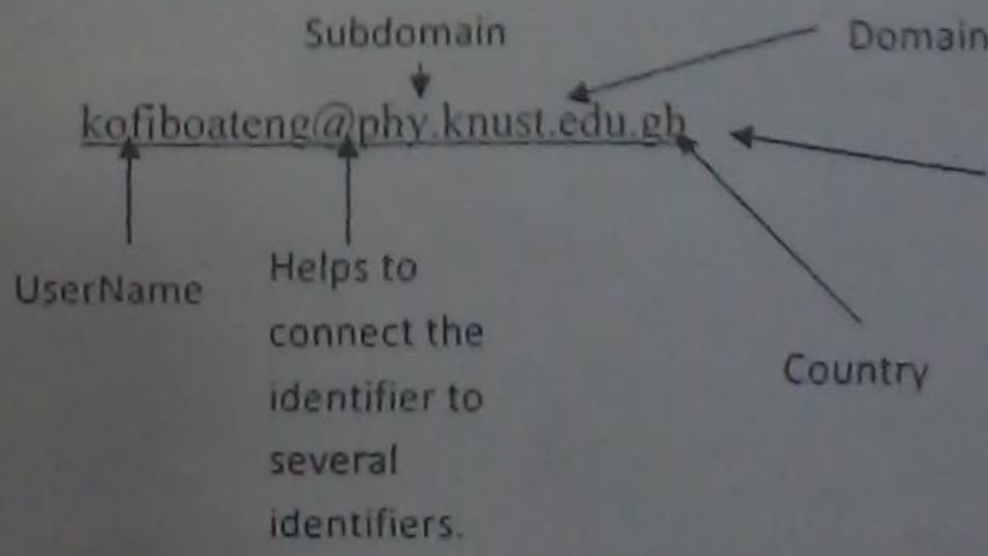
To obtain complete Internet Services through dial-up connection, you use an ISP. ISPs are local or national companies that provide unlimited public access to the internet and World Wide Web (www) for a flat rate. Using your computer and the ISP's communication software, you dial-up your ISP's phone number. The ISP's host computer uses SLIP (Serial Line Internet Protocol) or PPP (Point-to-Point Protocol) software to connect you to the internet. Most ISP's have small and limited coverage area

SESSION 2-1: TERMINOLOGIES ASSOCIATED WITH INTERNET TECHNOLOGIES.

2-1.1 Internet Addresses.



When you are on the Net, how do you get where you want to go? You need internet address. It is this address that helps to identify the intended recipient when it sends a message. They are normally unique. It is also known as IP Address.



2-1.2 Domain Name

Domain Name

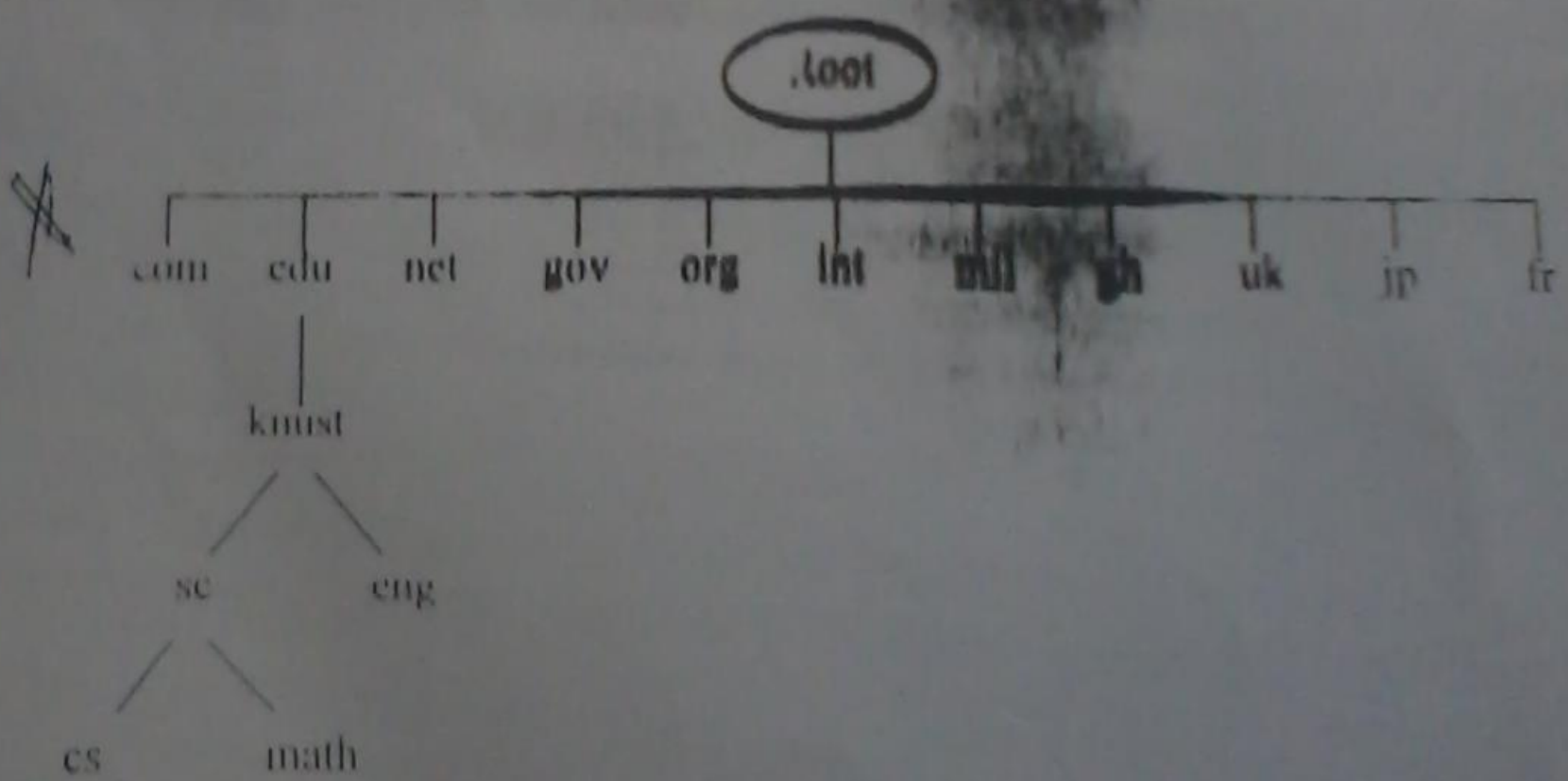
The name assigned to a computer on the Internet, a single computer's name can contain multip strings separated by periods (.) Domain names often end in

com,
gov,
edu,
.net, etc,

The internet service used to look up computer's name and find the computer's IP address by th help of DNS (Domain Name System)

The Domain name is a unique name used to identify and locate computers c nected to t internet. T : DNS is a collection of database" containing information about domain names, su domain names and corresponding IP addresses.

DNS



2-1.3 Domain Name Systems (DNS)/Servers

Domain Name Systems (DNS)/Servers

In small TCP/IP internetworks hosts are typically given simple names such as computer1 or system1. The mapping between these hosts' names and their associated IP addresses is then maintained as a "flat" database in a local file on each host. The resolver process on each host translate host name into IP addresses by a simple lookup procedure.

In a large network, the maintenance of the host files, which have to be identical for all hosts and continuously updated in order to reflect additions and changes, can become quite a tedious job. On internets with millions of names, this becomes impossible.

The DNS provides a network-wide directory service that maps host names against IP addresses. For most users this is a transparent process and not relevant whether the resolution takes place through a host file or through DNS.

When the IP address of a specific destination host has to be resolved, the DNS resolver on the source host contacts a DNS server somewhere on the network. There is usually more than one DNS server, and the database may be distributed among them. Where an individual DNS server does not have access to the entire database, the hosts name resolver may have to contact more than one DNS server or the DNS servers may exchange information amongst themselves in order to resolve the query.

Each DNS name server maintains a tree-structured directory database. The collective database stored on all the DNS servers forms a global namespace of all the hosts that can be referenced anywhere on the network.

Name Scheme hierarchical namespace

Originally internet namespace was "flat" i.e. it had no hierarchical structure. At that time it was administered by NIC – **Network Information Centre**. The task eventually became too large because of the rapidly increasing number of hosts and a hierarchical (tree-structure) namespace was adopted.

At present, the ultimate responsibility of this namespace is vested in the **Internet Assigned Names Authority (IANA).**

In a domain name, the most local domain is written first and the most global domain is written last. The domain name gardencity.edu might identify Garden City University. This domain name is registered against a specific IP address. The administrator of this domain name may now create subdomains such as cs.gardencity.edu for a computer science department at Garden City

University. The administrator of the computer science department in turn may assign a fu qualified domain name (FQDN) to an individual host as follows:

Computer1.cs.panbemcity.edu

If a user is referring to a specific host within a local network, a FQDN is not needed, as the DN resolver will automatically supply the missing high level domain name qualifier.

Standard Domain Names

The original namespace contained a set of standard top level domains without any reference to specific country. Since the original Internet was not envisaged to exist beyo borders, tl absence of any reference to a country implies an organization within USA.

The following are some of the common top level domains administered by IANA.

.edu educational sites
.gov government sites
.com commercial sites
.mil military sites
.int organizations established by international treaties
.org organizations other than the above
.net major network support centres

Country codes are used to identify international sites. A two letter abbreviation can be used for particular country such as "UK" for United Kingdom or GH Ghana

2-1.4 Name Resolution

The two methods of name resolution in DNS are *iterative resolution* and *recursive resolution*. iterative resolution, if a client sends a request to a name server that does not have the informatio the client needs, the server returns a pointer to a different name server and the client sends a ne request to that server. In recursive resolution, if a client sends a request to a server that doesn have the requested information, that server takes on the responsibility for sending requests other servers to find the neeessary records, and then returns them to the client. A server doin this takes on the role of client for its requests to other servers.

. **With recursive resolution**, the DNS client makes the initial request. The burden of t rocessing is then borne by the server, who may have to contact other servers before eventual assing the result back to the client. This is typical for smaller hosts such as pc and laptops.

. **With iterative recursion**, the resolver contacts a server that either provides the answer fers the esolver to another name server. This process is repeated until the resolution process

complete. The computational burden is shared between resolver and name servers - large computers and mainframes.

Conventional name resolution transforms a DNS name into an IP address. At the highest level, this process can be considered to have two phases. In the first phase, we locate a DNS name server that has the information we need: the address that goes with a particular name. In the second phase, we send that server a request containing the name we want to resolve, and it sends back the address required.

The Difficult Part of Name Resolution: Finding the Correct Server

Somewhat ironically, the second phase (the actual mapping of the name into an address) is fairly simple. It is the first phase—finding the right server—that is potentially difficult, and comprises most of the work in DNS name resolution. While perhaps surprising, this is a predictable result of how DNS is structured. Name information in DNS is not centralized, but rather distributed throughout a hierarchy of servers, each of which is responsible for one zone in the DNS name space. This means we have to follow a special sequence of steps to let us find the server that has the information we need.

The formal process of name resolution parallels the tree-like hierarchy of the DNS name space, authorities and servers. Resolution of a particular DNS name starts with the most general part of the name, and proceeds from it to the most specific part. Naturally, the most general part of every name is the *root* of the DNS tree, represented in a name as a trailing "dot", sometimes omitted. The next most-specific part is the top-level domain, then the second-level domain and so forth. The DNS name servers are "linked" in that the DNS server at one level knows the name of the servers that are responsible for subdomains in zones below it at the next level.

Suppose we start with the fully-qualified domain name (FQDN) "C.B.A.". Formally, every name resolution begins with the root of the tree—this is why the root name servers are so important. It's possible that the root name servers are authoritative for this name, but probably not; that's not what the root name servers are usually used for. What the root name server *does* know is the name of the server responsible for the top-level domain, "A.".

The name server for "A." in turn may have the information to resolve "C.B.A." It's still fairly high level, though, so "C.B.A" is probably not directly within its zone. In that case, it will not know the address we seek, but it will know the name of the server responsible for "B.A.". In turn, that name server may be authoritative for "C.B.A.", or it may just know the address of the server for "C.B.A.", which will have the information we need. As you can see, it is very possible that several different servers may be needed in a name resolution.

Key Concept: Since DNS name information is stored as a distributed database spread acros many servers, name resolution cannot usually be performed using a single request/respons communication. It is first necessary to find the correct server that has the information that th resolver requires. This usually requires a sequence of message exchanges, starting from a roo name server and proceeding down to the specific server containing the resource records that th client requires.

DNS Name Resolution Techniques

The DNS standards actually define two distinct ways of following this hierarchy of servers t discover the correct one. They both eventually lead to the right device, but they differ in ho they assign responsibility for resolution when it requires multiple steps.

Iterative Resolution

When a client send an iterative request to a name server, the server responds back wit either the answer to the request (for a regular resolution, the IP address we want) *or* the name another server that has the information or is closer to it. The original client must then *iterate* b sending a new request to this referred server, which again may either answer it or provi another server name. The process continues until the right server is found; the me...od illustrated in Figure below.

Recursive Resolution

When a client sends a recursive request to a name server, the server responds back with t answer if it has the information sought. If it doesn't, the server takes responsibility for finding th answer by becoming a client on behalf of the original client and sending new requests to oth servers. The original client only sends one request, and eventually gets the information it wan (or an error message if it is not available). This technique is shown in **Figure B**

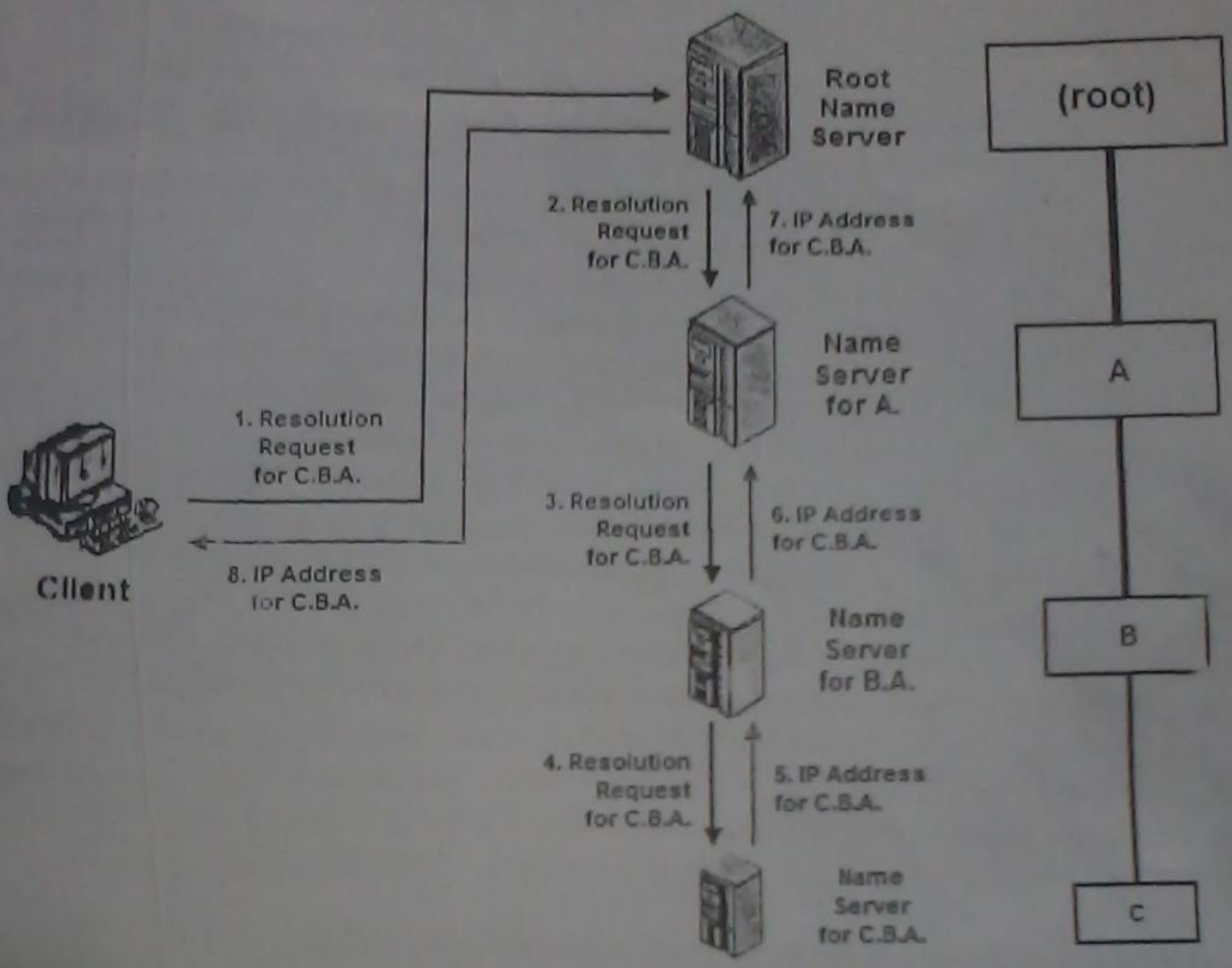


Figure B: Recursive DNS Name Resolution

This is the same theoretical DNS resolution that I showed in Figure A, but this time, the client asks for the name servers to perform recursive resolution and they agree to do so. As in the iterative case, the client sends its initial request to the root name server. That server doesn't have the address of "C.B.A.", but instead of merely returning to the client the address of the name server for "A.", it sends a request to that server itself. That name server sends a request to the server for "B.A.", which in turn sends a request to the server for "C.B.A.". The address of "C.B.A." is then carried back up the chain of requests, from the server of "C.B.A." to that of "B.A.", then "A.", then the root, and then finally, back to the client.

Contrasting Iterative and Recursive Resolution

To help explain the difference between these methods, let's take a side-trip to a real-world case. Suppose you are trying to find the phone number of your old friend Carol, with whom you

haven't spoken in years. You call your friend Joe. Joe doesn't have Carol's number, but he gi
you John's number, suppose you call him. So you dial up John; he doesn't have
information but he knows the number of Carol's best friend, Debbie, and gives that to you. Y
call Debbie and she gives you Carol's information. This is an example of an iterative process.
contrast, suppose you called Joe and Joe said "I don't know, but I think I know how to find ou
He called John and then Debbie and called you back with the phone number. That would be
recursive resolution.

So recursive iteration is doing the job yourself, while recursion is "passing the buck". Y
might think that everyone would always want to use recursion since it makes "the other guy"
the work. This is true, but "passing the buck" is not considered good form if it is not done w
permission. Not all name servers support recursion, especially servers near the top of
hierarchy. Obviously, we don't want to bog down the root name servers and the ones that han
.COM and other critical TLDs with doing recursion. It is for this reason that clients m
request that name servers perform recursion for them.

One place where recursion is often used is with the local name server on a network. Rather t
making client machine resolvers perform iterative resolution, it is common for the resolver
generate a recursive request to the local DNS server, which then generates iterative request
other servers as needed. As you can see, recursive and iterative requests can be combined i
simple resolution, providing significant flexibility to the process as a whole. This is demonstra
in a more realistic example in the topic detailing the DNS name resolution process.

Again, remember that for the purpose of understanding resolution, a DNS server can in fact
as a client. As soon as a DNS server accepts a recursive request for resolution on a nam
cannot resolve itself, it becomes a client in the process. I should also point out that it is comm
for resolvers to know the names of not one but two local DNS servers, so if a problem occ
reaching the first they can try the second.

Key Concept: The two methods of name resolution in DNS are *iterative resolution* a
recursive resolution. In iterative resolution, if a client sends a request to a name server that d
not have the information the client needs, the server returns a pointer to a different name ser
and the client sends a new request to that server. In recursive resolution, if a client send
request to a server that doesn't have the requested information, that server takes on
responsibility for sending requests to other servers to find the necessary records, and then ret
them to the client. A server doing this takes on the role of a client for its requests to other serve

The DNS name servers we explored in the preceding section are arguably the most impo
part of the system as a whole. After all, they store all the data on the system and actually pro

the addresses we need when names are given to them; without these servers, there would be no DNS at all. Of course, what use is a server if nobody is asking for service? The clients in the system, called *resolvers*, are also important, because they initiate the process of name resolution; resolvers are where the "rubber meets the road", so to speak.

The operation of DNS resolvers is explained in the two main DNS standards. RFC 1034 describes the functions performed by resolvers, and how they work in general terms. This includes a discussion of the algorithm used to conduct name resolution. RFC 1035 deals more with the implementation details of resolvers, and the fine points of *how* they do their jobs. Several subsequent standards have of course modified these base standards, changing some of the ways that resolvers work in different ways.

Name Resolution Services

Just as the main job of a DNS server is to store DNS name data and "serve" it when it receives requests, the main job of a DNS resolver is to, well, *resolve*. While most people only think of name resolution as the process of transforming a DNS name into an IP address, this is but one of several types of resolution services performed by DNS. A few of the most typical types of DNS resolution are:

- **Standard Name Resolution:** Taking a DNS name as input and determining its corresponding IP address.
- **Reverse Name Resolution:** Taking an IP address and determining what name is associated with it.
- **Electronic Mail Resolution:** Determining where to send electronic mail (e-mail) messages based on the e-mail address used in a message.

There are other types of resolution activities as well, though again, most name resolution requests are of the "standard" variety, making it the primary focus in our discussion.

Resolution Functions Performed By Name Resolvers

To accomplish its resolution duties, name resolvers perform a number of related functions:

- **Providing the User Interface:** In order for DNS to be of maximum value to TCP/IP users, it must be possible for names to be used interchangeably with addresses. This is usually done automatically by the resolver, which provides an interface to the user to allow names to be entered and used like addresses.

- **Forming and Sending Queries:** Given a name to resolve, the DNS resolver must creat… an appropriate query using the DNS messaging system, determine what type of resolutio… to perform, and send the query to the appropriate name server.

- **Processing Responses:** The resolver must accept back responses from the DNS server … which it sent its query, and decide what to do with the information within the reply. A… we'll see, it may be necessary for more than one server to be contacted for a particul… name resolution.

This seems fairly simple, and it is in some ways, but implementation can become rath… complicated. Bear in mind that the resolver may need to "juggle" several outstanding nam… resolutions simultaneously. It has to keep track of the different requests, queries and respons… and make sure everything is kept straight.

The user interface is a very important part of a name resolver's job. We want users to be able … just use a name and have their software automatically treat it like an address. For this reaso… normal name resolution usually doesn't involve explicitly running a piece of "resolver softwar… Consider again your Web browser. You don't have to say "please find the IP address … www.xyzindustries.com" and then say "please connect to this IP address for XYZ Industries … You just type in "www.xyzindustries.com" and the name resolution happens "magically".

There is no magic, of course. The resolver is just called *implicitly* instead of explicitly. The W… browser recognizes that a name has been entered instead of an IP address and feeds it to … resolver, saying "I need you to resolve this name, please". (Hey, it never hurts to be polite.) T… resolver then takes care of resolution and provides the IP address back to the Web brow… which connects to the site. Thus, the resolver is the interface between the user (the human … and the software user, the browser) and the DNS system.

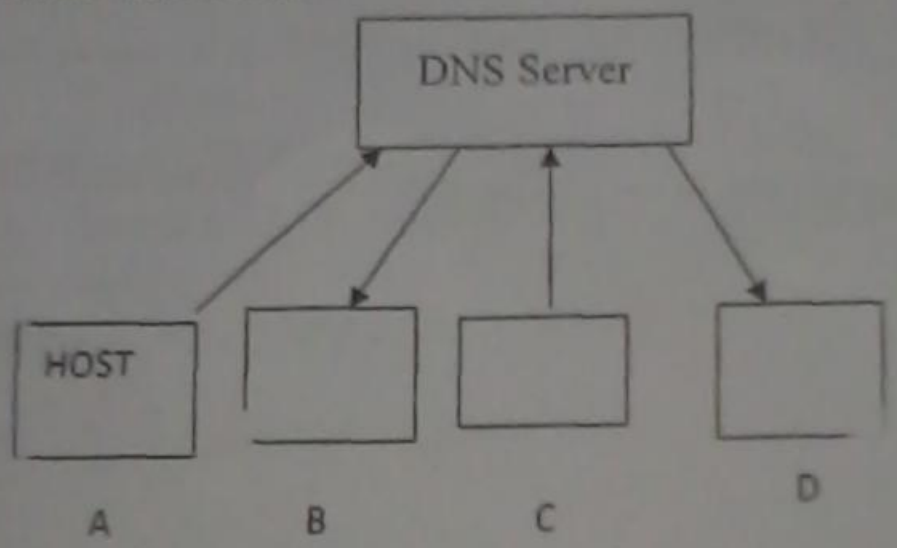
Key Concept: The primary clients in DNS are software modules called DNS *name resolve…* They are responsible for accepting names from client software, generating resolution reques… DNS servers, and processing and returning responses.

Other Functions Performed By Name Resolvers

Name resolvers don't have to perform nearly as many administrative jobs as name server… clients are usually simpler than servers in this regard. One important support function that m… name resolvers *do* perform, however, is caching. Like name servers, name resolvers can … the results of the name resolutions they perform to save time if the same resolution is requ… again (not all resolvers perform caching, however.)

I should point out that even though resolvers are the DNS components that are most associated with name resolution, name servers can also act as clients in certain types of name resolution. In fact, it is possible to set up a network so that the resolvers on each of the client machines do nothing more than hand resolution requests to a local DNS server and let the server take care of it. In this case, the client resolver becomes little more than a shell, sometimes called a *stub resolver*. This has the advantage of centralizing name resolution for the network, but a potential disadvantage of performance reduction.

DNS Name Resolution



Authoritative servers

DNS Frame Format

The message format for DNS messages is as follows.

0 15 16 17 20 21 27 28 31 Bits

ID	QR	OP code	Flags	RCODE
QDCOUNT	ANCOUNT			
NSCOUNT	ARCOUNT			
Question Section				
Answer Section				
Authority Section				
Additional Information Section				

ID Identifying Queries and Responds
QR flag that identifies the message as a query. QR – 0/– 1

OP Code
0 standard query
1 inverse
2 server status request

Flags used to describe the message further. They are from right to left.
AA Authoritative answer
TC Truncation
RA Recursion Desired
RA Recursion Available

RCode the last field in the long word is used for response codes with the following
0 no error
1 - format error
2 server error
3 name error
4 not used
5 refused

Four count fields indicate the length of the field to follow
QDCOUNT gives the no. of question entries
ANCOUNT gives the no. of resource records in the answer section
NSCOUNT refers to the no. of name server resource records in the authority section
ARCOUNT refers to the no. of resource records in the additional records section.

Question section

This contains queries in the format shown. A query consists of a query domain name containing the FQDN about which the type of information required, and a query which the is associated

Answer Section

This contains information returned in response to a query in a specific format. The re domain name, type, and class fields are from the original query. The time to live field sp how long this information can be used if it is cached at the local host. The format of the r data field depends on the type of information required

Authority Section

This identifies the server that actually provided the information if a name server has to contact another name server for a response. The format for this field is the same as for the answer section

Additional Query Information

This contains additional information related to the name in query; (e.g. the IP address of the host that is the mail exchanger, in responses to a MX (mail exchanger) query

The DNS message contains a query type field, since the name server database consists of many types of information. The following list shows some of the types.

A Host IP address

CNAME canonical domain name for an alias

MINFO information about a mail box or mail list.

MX name of a host that acts as mail exchanger

NS name of authoritative server for a domain

PTR domain name

SOA multiple fields that specifies which parts of the naming hierarchy a server implements.

Table of Contents