Web Basics

What is the web?

The World-Wide Web is a collection of documents and services, distributed across the Internet and linked together by hypertext links. The web is therefore a s*ubset* of the Internet, not the same thing.

Much of what is available on the web consists of web documents, which are often (somewhat misleadingly) called "*home pages*."

A *home page* appears to be a single entity, but is actually made up of a number of separate and distinct files, which may incorporate one or more of the following, in different configurations:

**text*
**graphics*
**animation*
**audio*
**video*
**hyperlinks (text or graphics which lead you from document to document)*
**interactive element, including:*
specially embedded programs such as:
** plug-ins (downloadable sets of software that enable the user to use part of a web document.)*

The base document of a home page is a file which is mostly text, containing commands ("*markup")* which determines how the above elements are configured. The rules governing this markup form a simple programming language called "*HTML*".

HTML stands for "Hypertext Markup Language".

As mentioned above, the Web also provides access to services. Some of these services are assembled "on the fly" by a computer program that accesses information available in some other form, and presents the information in the same format as any other web page.

The Web is therefore a user interface providing access to many types of information available on the Internet.

How Does the Web Work?

When you are looking at a web page, and you click on a link to go to a new page, what really happens? What's going on behind the scenes?

To begin with, you should realize that you only see a part of the link (the visible text). Associated with this, and available to the browser, is the address, or URL, of the page you will "go to" if you follow the link. Following the link is no different than typing in the URL yourself.

<< **Links**>>

It is possible, from the term "link," to get the mistaken impression that the two documents are connected or "tied together" in some way. This isn't really the way it works at all. A better description might be "pointer," because links are all one-way. A link is very much like a "see also" notation in a book, which refers you to another place to get related information. The document containing the link contains information about the document being linked to (it contains its address), but the reverse is not true. The document being linked to is not affected in any way by the link. It does not contain any information about the linking document, or even about the fact that any link exists. (If the second document does happen to contain a link back to the first, this is a completely separate link, having nothing to do with the first one.)

That fact that you can (in most browsers) click on a "back" button after following a link and return to the original page has nothing to do with the link itself. It is possible simply because the browser maintains a "history" list: in other words, it "remembers" which pages you have been to, and in what order.

When you click on the link, the browser takes the URL, and checks to see if you have been to that page recently (if you have, and the browser has saved a copy of it that is still available, it may simply retrieve its own local copy of the document--this is a process known as caching).

Assuming you have not, it next makes a request of the appropriate type to the indicated server. If we look at the process of following a link from one web page to another as an example, the browser connects to the web server and sends a request message to that server for the new page indicated in the URL of the link.
The browser's request message is very short, primarily containing the resource identifier portion of the URL, but it may contain other information as well (such as the type of browser you are using).

Then the server sends back a response message, and immediately closes the connection. The response message normally contains two sections: a header, with information about the server and about the requested document, and then the document itself. The information about the document includes its format (i.e., what type of document it is), its size, and how recently it has changed.

But we're not done. The browser must next take this document and go through the process of displaying it, a process known as "rendering" the page. For a web page, this means separating the text in the document from the structure information that says what to do with that text (the HTML), and then drawing the text in the appropriate location and font. As part of this process, the browser may discover that a particular HTML instruction calls for some object, such as an image, to be embedded in the page.

In this case, since such objects are not considered part of the document itself, but as separate entities in their own right (each one has its own URL that is different from the URL for the page), the browser must take the URL of the object and begin the process again, locating the appropriate server, connecting, and making a request for the needed object. This process gets repeated for every separate object (every individual image) on a web page. Only after each object has been retrieved, positioned, and displayed within the page is the process of loading the page complete.

In many browsers, if you watch the area where status messages are displayed, and if the process does not happen too quickly, you can actually see messages for each of these stages as the page loads. For instance, the description just given explains why on some pages you may see a message like "97% of 11K", followed shortly afterwards by "15% of 11K". Is the browser going backwards? No, these are progress indicators on the downloading of two separate objects (probably images), each of which is 11K bytes (roughly 11,000 characters) in size.

What's So Special About the Web?

The functioning of the Web is dependent upon many things, but perhaps the three most important are: first, that the web clients (browsers) are multiprotocol clients, capable of interacting with several different kinds of servers; second, that there is a common addressing scheme that makes it possible to unambiguously identify what you want and where to find it, anywhere on the Web; and finally, the fact that web browsers are extensible and therefore capable of handling a virtually unlimited variety of resource types.

Multiprotocol Clients

Most client-server systems are based on the notion that two programs are going to cooperate to achieve a common goal. Therefore, these two programs, the client and the server, are designed in tandem, sharing a single methodology for communicating (a single "protocol"). They are designed to work with each other, and only each other. During the growth and development of the Internet, most of the Internet services, such as FTP, were designed in this way. In the case of FTP, it was intended that there would be FTP servers running on various machines on the network, to provide information, and it was expected that one would use an FTP client to retrieve that information. Most of the other services (especially the earlier ones) were conceived in a similar way.

This meant that you not only had to know where the information you wanted was, but also what kind of a server had charge of it, so that you could use the right tool to retrieve it. In some cases, the same kind of information (such as a simple block of text) might be retrievable in any of numerous different ways, depending upon how the holder of that information chose to distribute it (for example, via FTP, e-mail, news, a telnet service, a "plan" file retrieved as part of a "finger" request, and many more possibilities). This made collecting information from various places on the Internet something of a chore.

The Web manages, for the most part, to do away with this restriction. Because a web browser has built-in support for many different protocols (in other words, it can talk to many different kinds of server), you no longer need to switch tools to retrieve resources that happen to be distributed in different ways. There *are* "web" servers (they are actually called HTTP servers), and web browsers *are* clients which can communicate with them, but a web browser can *also* be an FTP client, or a news client, or a gopher client, or any of several other kinds of client as needed.

Universal Addressing Scheme

In order to be able to tie together all of the different kinds of resources and services that are available on the Web, it was necessary to develop a common way of identifying each individual resource, document, and service. The solution to this problem (as far as the Web is concerned) is a universal address called a URL, or Uniform Resource Locator.

The goal of the URL is to describe, as concisely as possible, what you want, where it is, and how to get it. By embedding this information in the address of the resource itself, the person trying to retrieve the resource no longer needs to be concerned with the details of *how* to accomplish the retrieval. The browser handles the process of connecting to the appropriate server in the appropriate way, asking it to locate the desired item, and making the retrieval automatically. So, just as web browsers are multiprotocol clients, URLs are multiprotocol addresses.

The structure of a URL is fairly simple. There are three basic parts. The "scheme", or protocol, comes first. This identifies the type of server that has charge of the resource, and indicates how the browser should talk to it (what protocol to use). Next, there is a "server" portion, which gives the name (or IP address) of the computer on which the server program is runnning (and port number, userid, and password, if necessary). Finally, there is the "resource identifier" itself. This indicates which of the many resources that server has that you are interested in. The form of the resource identifier varies, depending upon the type of server. Frequently, it is nothing more than a file name and a path (a description of the location of the file within a hierarchy of directories), although it can be more complicated than this.

You could think of a URL, then, as an abbreviated answer to the questions "Who?" "What?" "Where?" and "How?" in relation to some resource you want to obtain access to. The server part answers the "Who?" question: "Which entity has control over the resource I want? From whom do I get it?" The scheme part answers the "How?" question: "What method do I use to retrieve this item? How do I get it?" And the resource identifier part answers the "What?" and "Where?" questions: "What item do I want, and where is it within the collection of stuff at this site?"

Extensibility

Needing to know the type of server that was used to make some item of information available is not the only, nor even the biggest, obstacle to accessing it. Another factor is the format of the information itself. Depending upon what the information represents, there may be many different ways of organizing or arranging it: there are dozens of different formats for still images, several for sounds and moving images, many different (and often proprietary) formats for word processing documents, spreadsheets, presentations, and databases, and even variations on methods for storing plain text. Not only does this not exhaust the list of possibilities, but new kinds of information, and corresponding new formats, are being invented all the time.

Three features of the Web offer solutions to help cope with this problem. First, the structure of HTML makes it possible to mix media types within a collection of related material, either by embedding certain types of objects, such as images, within a document, or by linking to these and other types of objects as separate documents unto themselves. Second, the Web supports a scheme for classifying document types (this scheme is known as Multimedia Internet Mail Extensions, or MIME) that allows for the creation and classification of new formats. Finally, and in many ways, most importantly, the browsers themselves are extensible: when a browser encounters a format it does not itself know how to unravel and present to the user, it can pass off the job of doing so to another program that *does* know how to do it. Such programs are known as "helper applications." A specialized form of helper application that can use the browser window to display in, rather than creating a separate window, is known as a "plugin."

Together, these features allow the Web to incorporate new formats and new media types with relative ease. A web document can link to the new document using HTML; a MIME type indicates to the browser what format the document is in, and thereby designates which helper application to pass it to; and the helper application displays the document just as if such documents had been part of the Web from the very beginning.

What is the Internet?

"The" Internet (capital "I" this time) is the global information system consisting of computers on various interconnected networks using the TCP/IP family of protocols. Notice that "the" Internet is also "an" internet, according to our earlier definition, but that the reverse is not true. There are many other large networks (such as those operated by America Online and Compuserve, among others), which, whether they connect to the Internet or not, are not part of it (because they are based upon protocols other than TCP/IP). There are also private TCP/IP-based networks that are not connected to the Internet.

TCP/IP Protocol Family

The layers used in the TCP/IP family of protocols are shown in the table below. In fact, the family takes its name from two protocols at different levels: the transport-level protocol TCP, or "Transmission Control Protocol" and the network-level protocol IP, or "Internet Protocol." The protocols most users deal with directly, naturally, are the application-level protocols. Occasionally, however, it is useful to be aware of the other layers.

Application
  Telnet, FTP, SMTP, etc.
Transport
  TCP, UDP
Network
  IP
Link
  Ethernet, Token-Ring, PPP

Starting from the bottom and working upward...

Link-layer protocols deal with physically interfacing with the medium (such as coaxial cable) used to interconnect the computers into a network. This includes the operating system device driver software and the network interface card. Notice that different portions of the Internet may actually be built upon different physical network types.

Network-layer protocols deal with assigning addresses to all of the computers on the Internet, and with the movement of packets between source and destination. Routing of packets takes place at this level. (There are some other protocols at this level, such as ICMP, that are used for managing routers.)

Transport-layer protocols deal with data flow between computers. Dividing messages into packets, reassembling messages from packets, acknowledging receipt of packets or arranging for retransmission of missing or damaged packets, all take place here. TCP, in particular, uses the IP packet system to create reliable connections. (UDP provides faster, but not necessarily completely reliable, connections for specialized purposes like real-time audio or video, where occasional packet loss would not be disastrous, but the slowdown due to the overhead necessary to absolutely guarantee reliability would be.)

Application-layer protocols deal with the specifics of a particular application, such as electronic mail. Each application corresponds to a particular type of service supported by the Internet.

Client-Server Computing

An important aspect of most Internet applications is the fact that they are all based on a client-server model. This is basically a divide-and-conquer strategy for managing information and communication resources. In this division of labor, one program, known as "the server" is seen as holding, or controlling access to, some resource (or alternatively as providing some service). Another program (usually, but not always, on another computer) known as "the client" is seen as making a request for, and becoming the recipient of, this resource or service. Notice that the complete application involves both the client and the server; neither alone is sufficient.

Typically, the user interacts directly with the client program on his or her local computer, and uses it to retrieve information from some remote computer that is running the server program. Often, the server program is little more than a robot: it is continuously running, and it sits and waits for incoming requests, which it then attempts to fulfill. The "conversation" between client and server is a series of requests and responses: the client asks for something, and the server either satisfies the request (by providing whatever-it-is that was asked-for) or returns an error message indicating why it can't.

<< **Domain Name Services**>>

One of the most fundamental of the underlying Internet services is one most users take for granted; they usually aren't aware they are using a service at all. This is DNS, the Domain Name Services system. DNS is what makes it possible to specify a computer by name, for example "cs1.cc.lehigh.edu", instead of by number (128.180.1.27, in the case of cs1).

In the discussion above, we mention that addresses are assigned to individual computers throughout the Internet at the network layer. These addresses are known as IP addresses, and all routing of packets from one computer to another on the Internet uses them to specify source and destination. Whenever you specify a particular computer by name, this name must first be translated (the technical term is "resolved") into an IP address. So, for example, if you try to connect to a remote computer by name using Telnet, you are actually using two services: Telnet and DNS. The Telnet application program requests DNS resolution to obtain an IP address it can use to find the remote computer it is supposed to connect to.

There is no centralized listing of all of the computers on the Internet, either by name or in any other form. DNS is a distributed database of names. DNS servers at many sites around the globe contain information about the computers at that site, plus information about other nearby DNS servers which can be queried about computers outside the site (Lehigh maintains two official DNS servers as part of its campus network). DNS servers also "cache" (remember) addresses for computers that have been requested recently, in case they are needed again. Requests for distant computers are passed along until an address is found, a server is found that can definitely assert that no such machine exists, or a specified timeout period expires (this last case means that sometimes the system will fail to find a distant computer which does exist--in such instances, because of caching, trying again may solve the problem).

<<**Local Area Networks**>>

One common source of confusion about networking concerns the way Local Area Networks (LANs) fit into the picture. It's not as hard as it seems. There are several areas of difference, and they are much more significant than the areas of overlap. LANs, per se, are not considered to be part of the Internet, even though the same physical hardware (Ethernet cable and network interface cards, mostly) may be used to access both.

For one thing, LANs are typically used to share different kinds of resources than are made available by Internet applications: LANs are often used to provide access to shared applications programs (which are run directly from the LAN as if they were on the user's hard drive), shared disk space for common storage, and shared peripheral devices (such as printers and scanners). LANs (as the name implies) are typically used for creating smaller networks, usually within a single organization (or department). They can be also be used to provide internal electronic mail and messaging services, but Lehigh does not use them in this way, preferring to use Internet services for this.

More importantly, a LAN uses different protocols for moving information around the network than do Internet applications. LANs are also packet-based, and may use the same link-layer protocols (such as Ethernet), along with the same underlying hardware, but all higher-level functions are provided using different protocols. In Lehigh's case, where our LANs are based on Novell Netware, the LAN protocol is called IPX. Both types of packets (IP and IPX) circulate through the network simultaneously; so a given portion of the network may be both a subnet of the Internet and a LAN at the same time (in fact, a single user may be actively using both types of functions at the same time--for example, loading and running an application program, such as Telnet, from the LAN in order to access an Internet service), but that doesn't mean that these two things are the same thing.

What is a Network?

A network is a set of computers that have been interconnected so that they can communicate with one another, and thereby share information and resources.

Physical Infrastructure

Networks come in various sizes. In a small network, sometimes known as a "Local Area Network" or LAN, all of the computers are connected to a single "hub" that allows them to communicate with other computers on the same network. Larger networks can be created by connecting two or more smaller networks together. Such interconnected networks are known as "internets" (small "i"). In such cases, each smaller network is referred to as a "subnet." Special devices known as "routers" connect these subnets to a high-speed backbone, or to external long-haul networks. Notice that this means that a particular computer can participate in both local and wide-area networking at the same time, using the same hardware.

Information Flow

To properly understand modern computer networks, it is important to recognize some things about the way information moves within the network. Unlike the early telephone system, which used special "relays" (switches) to create a dedicated circuit between a sender and a receiver (this is called a "circuit-switched network"), computer networks use a method called "packet switching." What this means is that every communication is broken down into small units, called packets, that are independently routed to their destinations. (Most modern telephone networks are now actually packet-switched, even though they still seem to behave--from the user's point-of-view--as if they were circuit-switched.)

In a computer network, every computer is not directly-connected to every other computer; instead, information must often pass through a series of routers and/or other computers in order to reach its destination, specified in terms of an address. Each computer on the network (each "node") has a globally-unique address; no other computer anywhere in the world has this same address.

In some cases, there may be more than one path to a given destination. Different packets, even from within the same message, may take different paths to reach their destinations (and may not necessarily arrive in the same order they were originally sent in). Some packets may be lost or become corrupted in transit. The network is responsible for sorting all of this out and making sure that every packet arrives intact (including being sent again, if necessary) and is returned to its proper order within the original message.

> *<<Analogy>>*
>
> This is a little bit like taking a large cargo, loading it onto trucks, and sending it across country via the highway system. A good dispatcher can compensate for slow roads and places where bridges are out by routing traffic around these. Rather than clogging one small road, the trucks can be divided among several parallel routes, and all can thus reach their destinations faster. The end result is a system that is robust and highly reliable, because of its flexibility.

Protocols and Layering

One term that is frequently mentioned in connection with discussions about networks is "protocol."

> *<<Analogy>>*

> In diplomatic circles, this term refers to formalized customs or procedures governing conversations, debates, meetings, and other sorts of diplomatic exchange. This includes proper forms of address, ritual greetings ("Good Morning, Your Highness" and so on), who gets to speak first, when one is entitled to make a rebuttal, what sort of terminology is appropriate, and other similar sorts of things. Well, with respect to computers, the term still means pretty much the same thing.

Protocol describes formalized rules or accepted customary usage governing communications between computers. The exact sequence of messages used to request, establish, maintain, or break off communication, and the format of their contents, is specified by a protocol. This is what makes computer-based communication possible. If both the sending and receiving computer programs didn't agree on how they were going to communicate, each would see the other's messages as nothing more than gibberish.

But the rules aren't specified all at once. Instead, the various functions that the network must perform are divided conceptually into separate "levels" of abstraction.

> *<<Analogy>>*
>
> Just as a manager might delegate certain details to subordinates, for example specifying that a package be delivered, but leaving the decision of whether to use Federal Express or UPS or some other shipper up to an assistant, computer protocols are divided into several layers. This keeps the problem of "what" you are trying to accomplish from getting unnecessarily tangled up in the details of "how" each part of the task will get done.

Lower-level protocols specify details more closely related to the workings of the hardware, while higher-level protocols specify broad functions more closely related to behavior that a typical user would see. All of the protocols at the various levels need to work together, however, and so these related protocols are seen as belonging to a family: higher-level protocols built upon services provided by lower-level protocols.

Internet Services

Overview

The Internet is much more than simply a bunch of computers that happen to be connected to each other. The physical aspects (hardware) of the various networks that make up the Internet are what make it possible for its various computers to communicate with one another, but it is the set of services (provided by software, and built upon the underlying TCP/IP protocol) that really defines the Internet.

Each service represents a different kind of communication: one specific way of passing information from computer to computer in order to meet a particular need. Each service utilizes its own special application-level protocol within the TCP/IP "family" of protocols.

Some of the most important of the Internet services are summarized below. Note the type of
target address typically given when using the service, along with the protocol used by the service (in many cases, the client application the user invokes to use the service, and perhaps even the service itself, are commonly referred to by the name of the protocol--for example, Telnet and FTP).

Electronic Mail

- A one-way relay of a message to one or more specified recipients.
- A mailbox (usually a userid or account) on a particular host (a computer, usually specified by name, but may be specified by IP address).
- SMTP (Simple Mail Transfer Protocol)

The oldest, and still by far the most popular, of the Internet services. Modeled closely after its non-electronic counterpart, it is an essentially one-way noninteractive message delivery service. Yes, you can correspond with another user, but you cannot interact in real time. Outgoing messages are queued up by the mail server (known as a "message transfer agent," MTA) and passed along in store-and-forward relay fashion from MTA to MTA until they eventually reach their destinations (this can take anywhere from seconds to days), where they are once again queued in a "spool file" until the client application (known as a "user agent") is ready to retrieve them (the READ command on Lehigh's Network Server invokes a user agent, and the various POP-Mail clients such as Pegasus, Eudora, and Netscape Mail are also user agents).

Like regular postal mail (sometimes sneeringly referred to as "snail mail"), electronic mail consists of a message wrapped in an envelope which carries information about where the mail has been and where it is going (this information is known as "header" information, and it is divided into fields such as "To:", "From:", "Date:", "Subject:", "Reply-to:", "Received-by:", "Organization:", "CC:", and so on). The address of a mail message usually looks like the following, although there are rare instances when it can be more complex (usually when the origin or destination of the message lies beyond the boundaries of the Internet proper):

user@host ***e.g.****:* ***philt@sparky.billing.megacorp.com***

In many instances (Lehigh is a good example), the name given for the host is simply an alias for the system responsible for relaying mail. The name "lehigh.edu" is not the real name of any computer; it is simply a pointer to the system which handles mail for Lehigh. Which system this is has changed several times over the last few years, without most users being aware of the fact (or, fortunately, needing to be). At the moment, lehigh.edu points to a computer whose real name is nss2.cc.lehigh.edu.

Chat

- Multiple two-way, sustained connections enabling real-time interactive discussion among several simultaneous users.
- An IRC Channel (named discussion topic).
- IRC (Internet Relay Chat)

News

- A distributed message exchange; serves as a worldwide public bulletin board system.
- A discussion group (a topic name).
- NNTP (Network News Transfer Protocol)

Like electronic mail, the fundamental unit of information upon which this service is based is the message. However, unlike mail, the message does not have a specific recipient. Instead, the message is put into a public area (called a "newsgroup") that is segregated into different topics. Anyone who is interested in the topic can access the newsgroup and read the message. This makes this service similar in function to a public bulletin board.

(A historical note: news is a descendant of an earlier service called Usenet, which was based on UNIX and used Unix-to-Unix-Copy or UUCP to provide store-and-forward movement of messages around the netowrk. Most people do not distinguish between the two.)

The topics are arranged hierarchically. The top-level groupings normally include (there may be others):

alt: Alternative topics
comp: Computer-related topics
misc: Miscellaneous topics
news: Topics relating to News itself
rec: Recreational topics
sci: Sciences and technical topics
soc: Social, political, and cultural topics
talk: General discussion and debate

Each grouping is further subdivided into specific topic areas, topics, and subtopics, with no set number of levels. For example, the rec grouping includes the subgroups

arts, food, games, humor, pets, puzzles, sport, and travel

(among many others). At the moment (groupings do change over time, as new topics develop, and interest in some topics wanes), the puzzles topic is not further subdivided. On the other hand, other topics may be repeatedly subdivided; for example, the arts topic contains a subtopic sf (science fiction), which in turn contains another subtopic tv (science fiction television programs), which in turn contains a subtopic babylon5 (a particular tv series), which at last glance was further subdivided into two final subtopics info (general information about the show) and moderated (controlled discussion). The complete name of any particular group is just the concatenation, in order from broadest grouping to most specific, of the names of the topics which contain that group; for example, the full name of the puzzles group in the first example above would be simply "rec.puzzles", while the full name of the info group in the last example would be: "rec.arts.sf.tv.babylon5.info". So newsgroup names look like the following:

area.topic.subtopic   ***e.g. rec.music.classical.guitar***

News is a distributed service. Messages are sent around the Internet, not to specific users, but to centralized points of collection (news servers), where a large number of users can use a local client (a newsreader program) to read the messages. This distribution process is called a news feed. A given news server does not necessarily accept all messages, nor keep copies of messages forever; Lehigh's news server, for example, does not accept messages from newsgroups within the alt hierarchy, and it limits the time it retains messages for most groups to about three days.

FTP   (**File Transfer)**

- A two-way, authenticated (requires userid and password) sustained connection enabling file upload and download between a local host (computer) and a remote host (computer).
- A remote host (computer).
- FTP (File Transfer Protocol)

FTP (File Transfer Protocol) is used to provide access to the file system of a remote computer: with FTP, you can browse through directories, list files, and transfer files between the local computer and the remote one. Using FTP creates a session, which is a sustained connection in real time between the two computers (in contrast to electronic mail, for example, where there is no direct connection between the sender's computer and the recipient's, and the message is not passed in real time, but may be delayed). There is an explicit "login" to begin the session, and an explicit "logout" to end it. The login process is "authenticated", which means that it requires a userid and password.

FTP is another client-server system. As is typical of such systems, the FTP client is the program that the user interacts with directly. The FTP server is the program that is running on the remote computer whose filesystem is to be examined. (If the remote computer is not running an FTP server, then you cannot connnect to it to transfer files. UNIX computers and other computers whose operating systems are designed to support multiple users or multiple processes at the same time often run many servers, or "daemons", at all times from the moment they boot up. If the computer is connected to a TCP/IP network, this frequently includes an FTP server. Most single-user computers, such as PCs, typically do not run any type of server unless the user of the system explicitly starts one. This is one reason why you generally can't connect to your office PC from home to retrieve a file via FTP; you would have had to have left an FTP server running on the PC when you left your office.)

To use FTP, you must specify the hostname (or the IP address) of the remote computer on which the FTP server is running. This looks like:

host e.g. : **lab1.physics.stanford.edu**

Once the session has been established, you can perform as many transactions (file transfers, directory listings, etc.) as you like.

Some organizations (and a few individuals) have created collections of files that are of interest to a broad audience, and they use FTP to make these files generally available. They do this by allowing users to connect to their system with a "guest" account that does not require a password. Such quasi-public facilities are known as archive sites. In most cases, the userid to permit access to such sites is the word "anonymous". For this reason, these sites are also called anonymous FTP sites. By custom, users supply their e-mail address as the password, to help those managing the site to track who is using their services.

Command-line Services

- A two-way, authenticated (requires userid and password) sustained connection enabling the user to issue commands to (run programs on), and receive responses from, a remote host.
- A remote host (computer) and (optionally) a port number.
- Telnet, Rlogin (Remote Login, primarily a UNIX service)

Telnet

This service is designed to allow a user to issue commands to a remote computer. Like FTP, telnet is session-oriented. There is a login process, which normally requires authentication (a userid and password), and a continuous connection is maintained between the local computer and the remote one, until the user explicitly performs the logout process. Virtually any command that could be issued at the console of the computer (i.e., while "sitting in front of the machine") can be given in a telnet session. You can run programs, browse through the filesystem, create, rename, modify, or delete files, and more (assuming, of course, that the privileges assigned to your userid on the remote system permit you to perform these tasks). Regardless of the environment or the operating system on the local computer (for example, a PC running Windows 95), the commands that are available and the syntax that must be used (the rules you must follow in constructing a command; the grammar of the command language) are completely dependent upon the operating system and software on the remote computer.

Before networking, remote access to services on a particular computer required you to go to a place where you could use a terminal device (a screen or a printer and a keyboard, usually) that was physically connected to the computer you wanted to access. By performing terminal emulation, telnet makes possible "virtual terminals": devices which act like terminals, even though they are not physically and permanently connected to any one remote computer. (In essence, your local computer--a PC, for example--pretends that it is some specific type of terminal device, such as a DEC vt100.) Telnet is responsible for translating between the capabilities of your actual screen and keyboard, and those expected by the remote computer.

Like FTP, telnet expects you to specify the host name (or IP address) of the computer you wish to connect to. In some cases, you must also specify a port number. This is usually given when a single computer can be connected to in more than one fashion. For example, connecting to the computer madlab.sprl.umich.edu on port 3000 gives you access to a special service at the University of Michigan, called the "Weather Underground" (this service gives National Weather Service forecast notices). This is a public service, so no userid and password are required. Connecting to the same host name, but on the standard port (omitting the port number) would allow you to log in to that computer to give ordinary commands (but only if you had a valid userid and password).

A similar service, rlogin, performs remote connection and terminal emulation between two UNIX computers. Because it is specific to UNIX, it does a slightly better job of handling certain characters that have special significance to UNIX programs.

Gopher

- A hierarchical, menu-based system for accessing resources (mostly documents and search services) on multiple hosts (computers) using intermittent connections.
- Resource identifier code and Gopher selector string.

This service was designed to be a file system browser. Unlike FTP, however, the goal was to embed as much information as possible within the system itself, freeing the user from having to remember unnecessary details. It allows the user to traverse from one system to another nearly transparently. In addition to several different types of files, Gopher provides access to search engines, directory services, and can link into telnet and tn3270 sessions. In many ways, it is the precursor to the World-Wide Web.

The interface to Gopher is based upon a series of menus that mimic a directory structure. As such, information is arranged in a definite hierarchy. Each menu item appears as a line of descriptive text; this can lead to another menu, or to a file or a service. The user simply navigates up and down through this series of menus. If a subsequent item or menu is hosted by a different computer than the current menu, the user need not be aware of this, or do anything special--setting up the proper connection is the responsibility of the person who maintains the Gopher server. Similarly, while the user can specify a starting point, by giving the hostname (and, if necessary port number) of the Gopher server to connect to initially, the configuration of the client usually specifies a default.

Other Services

Ping, NSLookup, Traceroute
Finger, Whois, Ph, LDAP
Archie, Veronica, WAIS
Listserv

World-Wide Web

- A multimedia hypertext-based system for accessing resources (including compound-documents, many different types of files, and various types of services) on multiple hosts (computers) using intermittent connections.
- Uniform Resource Locator (URL).
- HTTP (Hypertext Transfer Protocol)

Links and Navigation

To allow users to navigate or maneuver between web documents, links are embedded within those documents. Navigation on the WWW is predominantly handled by pointing and clicking. When you move the mouse over a link the cursor changes from a pointer arrow to a pointer finger. Clicking on a link such as hypertext or a graphic (picture) will display a new web document, that resides at a unique web address (or url).

Even though the link might point to a new web document, if that web document does
not exist at that specific url, a "File not Found" error message will be displayed.

Graphical links can be contained in either the whole graphical image or as a part of a graphic in an image map.

Hypertext

Links can be represented textually as hypertext or graphically (as pictures). Hypertext is typically displayed as underlined words or a string of associated words such as a phrase or sentence. Clicking on hypertext, the user can traverses to other web documents, download special files (executables, zipped), or open email. Hypertext is typically found embedded in web document narratives or in lists. Moreover, hypertext is an example of branched navigation, where the user can browse topics non-sequentially.

Image Map

An image with defined clickable areas that link to predefined url's is an image map. Image maps, then, allows branched navigation by the number of

Browsers

"Browser" is the generic word for the piece of software that allows you to view web pages. Different browsers (and versions of the same browser) have different capabilities, however, they all have some features in common.

Some examples of these common features include:
A place to enter the URL (uniform resource locator), or address, of the web page you wish to view.

A way in which to move from the current page being viewed to the previous page or next page (typically using forward and back buttons)

The ability to "save" the location of web pages that you find useful (this is called a bookmark).

Two of the most commonly used browsers are Netscape Navigator and Microsoft's Internet Explorer. Lehigh has chosen to standardize on Netscape Navigator and this lesson will focus on how to use Navigator to "surf the web"

Netscape Navigator Fundamentals

This section describes what you see in the main browser window of Netscape Navigator. It is important to note that this information refers to version 4.05 of Navigator. Earlier versions, while similar, may not have the same features or visual look.

The main browser window is what you see when you start up Navigator. Most of the tools and text fields that help you to navigate the web are visible, but you do have the option of hiding some tools to give you more viewing area (which is helpful on some web pages).

MAIN WINDOW

The title bar of the window shows the title of the currently loaded page. The menu bar is located below the title bar and accesses various pull-down menus.

TOOLBARS

NAVIGATION TOOLBAR

Below the menu bar is the navigation toolbar. The buttons located on the navigation toolbar activate the features you'll use most often while in Navigator. Clicking on these buttons allow you to re-visit pages, re-load pages, go to the page selected as your "home" page (the page that is automatically loaded when Navigator starts up), search for information and guide you to interesting sites on the web, print pages, view security information about the page you are currently viewing, and stop transfers of information in progress. This toolbar may be minimized by clicking on the vertical rectangular button found on the left edge of this toolbar.

To the right of the navigation toolbar is the status indicator. When a web page transfer is in progress you will see what appears to be comets streaking across the sky. If you click on this logo you will be transferred to Netscape's home page.

LOCATION TOOLBAR

The location toolbar is found below the navigation toolbar. This toolbar allows you to save and manipulate your bookmarks, and enter the location, or URL, of the page you wish to view. This toolbar may be minimized by clicking on the vertical rectangular button found on the left edge of this toolbar.

PERSONAL TOOLBAR

Below the location toolbar is the personal toolbar. The personal toolbar is available for you to customize your own "shortcut" buttons to particular sites. This toolbar may be minimized by clicking on the vertical rectangular button found on the left edge of the toolbar.

CONTENT AREA

Below the toolbar area is the content area. This is where the "useful stuff" is located and why you want to get to the web, after all!

Links are present on most web pages. The function of a link is to help you navigate around a web site or to point you to other useful sites with related information. Links are usually underlined and/or in a different color (highlighted) than the rest of the text. Clicking on the highlighted text will move you to the new page or location indicated. Also notice that as the mouse pointer passes over such highlighted text, it changes from an arrow to a pointing hand. Some images (pictures) are links as well; most of these (but not all) will be bordered in the same highlight color as the link text. The mouse pointer will change as it passes over these, too. Some links will change color after you have visited them, this is to help you remember what links you have already visited.

Some web pages use a technique called "frames" which appear to be a patchwork of pages together in the content area. These pages contain rectangular frames with each frame presenting its own information. This technique is used frequently to allow people to move around large web sites, typically one of the frames will be a table of contents or outline.

Error messages result from a variety of situations, often originating from the server providing the page you wish to see. Navigator tries to evaluate any problem you encounter and present information to help you solve or circumvent it. A common error message is "Page not found". Often this occurs because the server issuing the page is temporarily shut down or too busy with connections to handle your request. Occasionally the page is no longer available at the specified URL.

Beneath the content area to the left is an icon of a lock. This icon may appear in either the locked or unlocked positions. When locked, this indicates that the site is secured. This is typically seen in commercial sites where credit card or other personal or sensitive information is being transmitted.

To the right of the lock is the progress bar. The progress bar animates to show the progress of the current transfer. The progress bar shows the percentage done of the document being loaded and displays the word "Done" when the entire web page has been loaded for your viewing. If a web page is taking an extremely long time to load and you wish to discontinue the loading of that page, press the Stop button located on the navigation toolbar. Also when you move your mouse pointer over a link in the content area, you will see that the URL that the link points to is displayed in the progress bar.

To the right of the progress bar are icons that refer to other Netscape tools in the Communicator package. Other than the first icon of the captain's wheel, they are unrelated to Navigator and will not be discussed further in this course. The captain's wheel icon allows you to open up another Netscape Navigator window. You may open multiple windows, be aware that you may impact the performance of your pc if you have too many Navigator windows open at one time.

HOW DO I?

Print A Page:

For most web pages, printing is as simple as clicking on the Print icon located on the navigation toolbar. A pop-up window will appear prompting you for printer information, similar to other applications. When you are satisfied with the settings in this pop-up window, click on the OK button to send the information to the printer. If the print button is not available, you can also try selecting the text with your mouse, and using the Edit/Copy functions that are available from the menu bar at the top of the screen. This will copy the text to the Windows clipboard, from which you can then print.
(do we need to talk about printing white on black pages???)

Bookmark A Site:

When you find a particularly useful site you may want to save the URL for future use. In order to do this (while on the page you want to save) click on the Bookmarks button (on the location toolbar). A window will pop up which will allow you to add the bookmark, file the bookmark, or edit your bookmark file. Bookmarks are stored on your pc's hard disk as a file. When you choose the "Add Bookmark" option, the bookmark is apended to your bookmark file. As this list grows, you may want to organize your bookmarks into folders (by using the "Edit Bookmark" option). The "File Bookmark" option allows you to file a new bookmark into a particular folder once you have organized your bookmark file.

See **Where You've Been (View History):**

Navigator keeps track of the web sites that you have visited. To see where you've been (perhaps to return to a recent site whose URL you don't remember) you will select the Communicator option from the menu bar at the top of the screen and select History from that pull down menu. A shortcut for this is to simply press the Control-H keys. The History file can be cleared out as well. To clear your history file, select Edit/Preferences from the Menu bar, click on the Navigator category on the left side of the screen, and press the Clear History button displayed on the bottom right of the screen.

Fill Out a Web Form:

Web forms are used for many things, but typically to request information from somewhere or order something. One of the trickiest things about web forms is forgetting what you already know. On most computer-type forms you expect to press the Return key to move from area to area. It doesn't work that way on a web form. When you encounter one of these (and it won't take long till you do, believe me), you want to remember to use your mouse and click on the box to be filled out. On some forms you can use your tab key to jump from box to box, but sometimes the order of the boxes that the tab key moves you to is inconvenient. Be safe, use your mouse and click on the next box to be entered.

MAILTO: Links

On many web pages you will see a link that allows you to contact via email some person or organization. This is called a mailto link. When you click on this link, Netscape's mail package (called Communicator) will pop up on your screen. The "TO:" field will already be filled out with the email address of the person or organization you wish to contact. All you need to do is fill in the Subject line and body of the email message and click on the send button.

Cache and reloading

When you get a document 'off the web', your browser actually gets and holds a copy of that document in order to display it. This copy is kept in temporary storage, the 'cache', for a period of time, so that the browser does not need to get a new copy if you want to go back to that page.

This is much like the secretary's strategem, where documents called for by the boss are not refiled immediately, in case they are wanted again.

However, while the copy of the page is sitting in your computer, changes may have been made of the original on the web. Or the page may be 'interactive' so that it changes based on your input.

Sometimes you may have to get the browser to load a fresh copy of the page. Most of the time, this can be accomplished by clicking the 'Reload' button or choosing 'Reload' from the View menu.

However, periodically you may want to get rid of all the temporary 'cache' files your browser has, or you may need to, because occasionally 'Reload' doesn't work. This is a standard thing to try when a site that used to work just suddenly doesn't. To do this, you clear the cache.

Clearing the cache in Netscape 4:
- From the 'Edit' menu, choose 'Preferences'.
- In the "Category" box, double-click on 'Advanced' so that a submenu opens up.
- Click on 'Cache' under 'Advanced'.
- Click on 'Clear Memory Cache' button. Click on 'OK' in the message box.
- Click on 'Clear Disk Cache' button. Click on 'OK' in the message box.

Searching

I like to start w/a quote on the screen:

> *"The maddening thing about the virtual library is*
> *that it's not a digitization of the real library at all,*
> *although the instances of overlap are growing.*
> *Rather, it's a second, parallel library that's*
> *growing according to its own rules."*
>
> Paul Gilster
> Digital Literacy, 1997

The quote articulates the difference between "real" library content and the virtual library (which I interpret as the Internet). This quote becomes a springboard for enumerating the differences, which are...

1. (Unevenness of) Coverage: By far the most content on the WWW is popular/consumer oriented. Give supporting examples, stats (SEK can provide). Note that substantive content from govt, ed, thinktanks, research centers, etc is increasingly rich.

2. (Lack of) historical coverage. WWW has extremely poor coverage of historical information, no collective memory beyond its own history (early 90s). Great for current info, anything since early 90s.

3. Authority of content
-- convey the idea that anyone can publish; no one administers/controls content
-- hints for assessing credibility of content (handout?):

Evaluating Search Results

1.Where does the information come from?
can you determine origin from the URL?
do you recognize the organization?
is there any identifying information?

2.Who created the site?
what is their authority?
are they published or recognized experts?
what are their credentials?

3.How is the information selected?
what is the coverage?
what topics are included?
who decides what to include?

4.Who is the intended audience?
are you the intended audience?
is it stated somewhere on the page?

5.Is the information presented accurate?
are the facts documented?
are the facts attributed?

6.Is the information biased?
are the arguments emotional?
is the wording inflammatory?
is bias acknowledged?

7.Is the information current?
when was the page last updated?
what information was changed?

8.Is the site/information stable?
is it reliably accessible?

-- copyright ("why can't i read WAITING TO EXHALE on the WWW")
-- licensing ("why do I have to type my password to search LEXIS?")
-- scholarship ("why don't professors publish research on the WWW?")
tenure, peer review, dissemination channels for scholarly info

4. No permanence to content/ephemeral nature of WWW

II. Searching the WWW

Make a general statement that tools differ in how selective they are, in the number of sites/resources covered, and in the method/criteria for selection. To condense this for WebBasics, I would describe and give an example of each the types of tools listed below and summarize most of the rest in a separate handout (or leave it out).

1. Search engines
A search engine is a searchable index or

database of web pages and other
information available on the Internet

The database is created by using robots/crawlers
that harvest www pages systematically, automatically,
as comprehensively as possible

Comprehensive, inclusive, full text, powerful search
features. Different engines cover different pages but
praticaly speaking each covers a large proportion of the
WWW; most comprehensive tool.

Use When
your search is complex
you need comprehensive coverage
your topic is obscure
you're searching for a phrase

Do Not Use When
your topic is very broad and/or
vague
you want only a few good sources

EXAMPLES include Altavista, Infoseek,Lycos, Hotbot

2. Directories

Web directories allow you to browse hierarchical categories of information arranged by subject.

Most involve human selection of the sites/sources, more
quality control, more ordered, but not as comprehensive
Most cover fewer than 500,000 sites

Use when:
you have a broad topic
you have limited time
you do not need comprehensive
coverage
you are looking current events or
newsy topics
as a complement to other forms of
research

you're searching for an organization,
event, place, or current event

When not to use:
for specialized/complex searches

Selected Directories/Indexes:
Yahoo
Librarian's Index to the Internet
INFOMINE - Scholarly Internet
Resource Collections
My Virtual Reference Desk
Essential Links
Galaxy Guide to Information
LibWeb
The Mother-of-All BBS
Yahoo's Collection of Web
Directories/Indexes

3. Metasearch Engines

A metasearch engine conducts searches across a variety of search engines on your behalf. The metasearch engine interacts with other engines and compiles/presents results.

Collectively cover most of the (indexable) WWW

Use When
your search is straightforward
you want to find the best engine for
your search
your topic is obscure

Do Not Use When
your search is complex
you need special search capabilities

Metasearch Engines
All In One Search Page
Dogpile

Inference Find
Internet Sleuth
Metacrawler
Profusion

4. Review Services

A Review Service features a select set of resources that meet criteria for inclusion in the service's listings.

Review services are highly selective and tend to provide summary and/or evaluative info about sites.
Very selective, may cover <100,000 sites

Use When
you are looking for a broad topic
you want to find the best of the Web
your time is limited

Do Not Use When
your search is highly complex or obscure
you want EVERYTHING there is on a topic

Review Services Examples
Argus Clearinghouse
Excite Review Service
Scout Report
Top 5% from Lycos

6. TRENDS IN SEARCH TOOLS: more specialized services geared to specific user groups or content focus (medieval history search engine, aquatic sciences search engine, newsgroup search engines, image archives)

Cookies and Privacy

What Is A Cookie?

A cookie is a small piece of information that a server can ask a client (your web browser) to store for it. This information is in the form of a simple variable; in other words, an object with a NAME and a VALUE. Both the name and the value are just text strings, like "SESSIONJID" or "HQVXJ5282#AA324". Notice that there is no requirement that either you or your browser will have any idea what this variable represents, or what its value means. As long as it means something to the server the next time it sees it (when the browser returns the cookie to the server), the cookie has done its job. Of course, there's nothing that says the cookie has to be mysterious, either. It could, for example, be "USERJNAME=JohnJDoe", which is reasonably selfJevident.

What Does A Cookie Do?

The specifications for cookies describe them as a "state management mechanism". What this means is that, basically, a cookie is just a memory aid for a web server. Cookies are a tool that is intended to provide the solution to the problem that the protocol used for web transactions (HTTP) is "stateless".

Statelessness simply refers to the fact that, when you request a resource via the web, there is no inherent connection between the complete historical sequence of what you have seen previously and the resulting cumulative consequences of all of these steps (your "state") and the contents of the page you have requested. In other words, if the server is using just the HTTP protocol and HTML, with no other enhancements, the page you see is the same no matter who you are, or when you request it, or whatever else you may have done. Because of this, even though your travels through a particular web site may seem to you to blend smoothly into one continuous series of interactions, the web server you are visiting doesn't see you that way at all. Each page you view produces a separate request to the server that is unrelated to any other page you may have seen. (In fact, every image on the page also produces a separate request that is not related to the other images on the page or even to the page that it is on!) And your requests are mixed in along with everyone elses'.

This means that the server doesn't intrinsically know what other pages you may have already seen, (although it could try to create a database to keep track), or whether you will ever make any additional requests (since you didn't "log on" to that server, you don't have to log off, either, and so the server never has any way of being sure you are finished with it). In computer terms, you don't have a "session". What cookies are intended to do, basically, is to help the server create the illusion of continuity, so that your interaction with a web site behaves more the way you intuitively expect it to.

So What Are Cookies Used For?

Cookies can be used in a lot of different ways. Because they are relatively new, there are probably uses that no one has even thought of yet. Some examples of typical uses include: Website Personalization, Online Ordering, Targeted Marketing, and Site Tracking.

WEBSITE PERSONALIZATION:

Many websites strive to provide useful information to their visitors, but anticipating their audience's needs is sometimes difficult. By allowing users to identify themselves and indicate their preferences for what type of information is most interesting or important to them, sites such as My Yahoo! and GIST can customize their presentation of information in a way that maximizes its usefulness to each individual vistor. Cookies can be used to store the user's identity (username) or even the preferences themselves.

ONLINE ORDERING:

Gathering the information you need in order to make a decision to buy something often happens in stages. You may browse a product catalog, adding things to a virtual "shopping cart," which you may eventually decide to purchase (at which point you will need to enter billing and or shipping information). Cookies are often used as a way of creating such "virtual shopping carts."

TARGETED MARKETING:

This is probably the area that causes the greatest concern. It includes a number of relatively innocuous types of activity, ranging from making sure that you do not see the same ads over and over again, to helping advertisers gather statistics on how effective their advertising campaigns are (by revealing details such as how many people, who, after having seen the ad, bothered to follow the link to find out more). It potentially includes (under certain specific conditions, which require your assistance) unsolicited mailings of advertising material direct to individual users. The key to keeping this in perspective is to remember that the server can only ask the browser to help it remember something it already knows, and that the browser is not required to comply. (More about this below.)

SITE TRACKING:

Even if they aren't actually trying to sell something, web site designers still want to know whether or not they have been successful. How popular is the site? Do visitors stop at the front page, and then leave the site altogether, or do they stay and browse through it? Are they bookmarking pages? Are some pages less useful than others, indicating that they may need to be redesigned? Because of the fact that visitors don't have sessions, keeping track of how people actually use their sites can be very difficult for web designers. Cookies can help.

Why Would Anyone Want To Be Kept Track Of?

Suppose you are a person who loves to read, and you happen to have a neighborhood bookstore nearby that you visit frequently. As the sales clerk gets to know you, he can do small things, like greeting you by name, that make you feel welcome, and make your visits just a bit more pleasant. If you buy a particular outJofJtown newpaper every Friday, he can anticipate this, and have it ready (or even set one aside, so they don't run out). If he knows you have been reading several books in a series, he can let you know when the next sequel is due to be published. Based on the books you've bought, and on what you've commented about, he may even be able to suggest a book you might like. Basically, you get better service, because he knows you. Even if you happen to be the sort of person who prefers to browse without being bothered by sales clerks, and who would consider a clerk's presumption in making reading suggestions to be a rude invasion of privacy, the clerk has to know you in order to know that. After all, not everybody feels that way. If the clerk knows you don't like to be bothered, and he still does, then, of course, that is rude.

The point here is that it isn't the information itself, it's what gets done with it that matters. Web sites are not people; they are computer programs. Most people aren't comfortable dealing with something that's completely cold and impersonal, and because computers don't have the sort of everyday capabilities we take for granted in peopleJJlike common sense or intuitionJJit's hard for them to be anything else. Web sites are potentially worse than most computer interactions, because they are, in effect, constantly forgetting who you are from minute to minute. It's as if you were dealing with a clerk who had an advanced case of Alzheimer's. By keeping track of you, a wellJdesigned web site can do better than this, and cookies are one tool that can be used to help the computer keep track.

But Aren't Cookies A Big Risk?

Is there a risk that cookies will be used to keep track of things you'd prefer weren't tracked? Yes, there is. Wouldn't getting rid of cookies altogether prevent this? No, because cookies are just one tool that can be used to make certain kinds of tracking easier to do; they aren't the only one, and cookies alone aren't what makes tracking possible.

Cookies are merely a solution to a problem; if you get rid of cookies, the problem still remains to be solved, and some tool will be used to solve it. If not cookies, then something else. Besides, keeping track of you, per se, is not the issue. Despite how many of us may feel at times, advertising and marketing aren't inherently evil: after all, there are times when you have something you want to buy, and if you don't know about it, you can't buy it.

The problem is the constant buzz of commercials that are irrelevant to our needs, the flood of useless information in the form of junk mail, the intrusive nature of direct telephone marketing. Think about it: if the people who were trying to sell something actually knew exactly what people wanted in a product, knew exactly who wanted it, knew when they were ready to buy it, and used that information responsibly, things could be nearly ideal for both buyer and seller. You could have nearly perfect service and incredibly efficient commerce.

The rub, as they say, is that part about using the information responsibly. Marketing can be seen as more than just identifying the people who are ready to buy what you have to sell: it can also involve convincing the uncertain, or changing the minds of those who aren't interested. And that's where many people start to feel manipulated and intruded upon.

What Information Can Be Tracked?

Keep in mind that cookies are a tool for helping the web server remember things. Obviously, the server can only try to remember something it already knows. Notice that cookies have nothing to do with what the server knows, but only with keeping track of it.

There are a few things the server has to know, in order to communicate with your browser. Like the network address (called an IP number) of the computer it should send pages to, when the browser requests them (this may be a gateway, or proxy server, or the machine you are connected to at your Internet Service Provider, or it may be your PC, if that is on a network). Without this, you would get nothing.

There are other things it is useful, but not absolutely necessary, for the server to know. Your browser may tell the server some of these, without you realizing it. For example, most browsers inform the server which browser, and what version number of that browser, you are using, along with what type of computer you are running the browser on. The server could (but often doesn't) use this information to make sure you see a version of the page that uses features that are compatible with your setup. If you are following a link (as opposed to typing in a URL directly, or using a bookmark), many browsers will tell the server which page the link was on, so that it will know where you came from.

Then there is information that the server ASSIGNS to you. This is information created and used by the server for its own purposes; it isn't really personal information. There are lots of analogues to this in the real world: if you go bowling, and you check out a pair of shoes, there may be a number stenciled on themJJwhat does this number mean? It identifies which pair of shoes you are using. The web server can assign numbers like that, too.

Most other types of information are only available to the server if you explicitly provide it. This includes things like your real name, your social security number, your postal address, your telephone number, and so on. The browser doesn't know any of these and so can't tell the server without you knowing about it. So the only way the server can connect your activities on the web with any of these things is if you tell the server who you are (usually by filling out a form). Once the server knows who you are, it can keep track of you as long as you remain at that site (or, using cookies, if you return to that site before the cookie expires).

Finally, the server can collect separate pieces of information and examine them to look for patterns, in effect creating new information. If these patterns contain, or can be linked to, personal information, then the potential for abuse does indeed exist. However, it isn't cookies that makes it possible to do this; much of the information is also automatically collected in server log files, and the information from the forms you fill out could easily be collected in a database on the server.