

Lab 1.2

Internet Basics

This lab covers fundamental concepts of network organization, focusing on the client-server model for network resources such as web pages and file storage. The procedure includes uploading files via ftp to a web folder but does not cover any HTML, which is left for more focused coverage in Lab 1.3. This lab assumes basic experience using a web browser and e-mail.

Vocabulary

All key vocabulary used in this lab are listed below, with closely related words listed together:

- host, IP address, hostname, domain
- hierarchy
- server, client
- web server, web browser (client)
- URL
- HTTP, FTP
- user name, password
- local, remote
- upload, download
- file properties, permissions

Post-lab Questions

Write your answers after completing the lab, but read them carefully now and keep them in mind during the lab.

1. Describe at least one way in which IP addresses and phone numbers are similar.

HTTP and FTP are both standard ways of sending/receiving files through a network. How do they compare with respect to privacy? How do they compare with respect to convenience?

Is it correct to say that folders/directories directly contain data? Why or why not?

Recall *pathname*s from the last lab. Discuss at least one way in

which pathnames and URLs are similar.

How can you recognize whether something is a pathname or a URL? (I.e., what is visibly different about them?)

Discussion and Procedure

Part 1. Domains, Hostnames and IP Addresses

Hostnames instead of IPs. In Chapter 3, we saw how every computer on the Internet has a unique numeric address called an *IP address*. An IP address is actually composed of four numbers, each in the range 0 to 255, separated by periods which we read as, “dot.” IP addresses are hard for humans to remember, so we usually identify networked computers using *hostnames*, instead. (Networked computers are often called *hosts*, hence, “hostnames.”) Hostnames are automatically translated into IP addresses by a special system called DNS (short for Domain Name System).

Hostnames and e-mail. You might not realize it, but you have been using hostnames ever since you started using e-mail. Every e-mail address has a hostname after the @ (“at”) character. In a world without DNS, you would have to put an IP address, instead of a hostname, in every e-mail address.

The fact that hostnames are composed of words rather than numbers is only part of the reason why they are easier for us to remember. Hostnames are organized hierarchically, the same way folders on disks can be organized. Let’s look at an example hostname and an example file location to compare these two hierarchical organization schemes:

example hostname: `www.cs.washington.edu`

example filename: `c:\personal\finances\taxes\1040.pdf`

Hostnames and domains. Let’s begin by taking apart the hostname. Notice that the hostname is split into parts separated by periods, which, like with IP addresses, are read as “dot.” The parts of a hostname are ordered from more specific to less specific, which is the opposite of how parts of the filename are ordered. Note that with the filename, directory names are separated by backslashes, and the path begins with the most general categorization (the drive letter, C) and gets more specific as you read on. On the other hand, the most general categorization in the hostname is the `.edu` part. All hosts in educational institutions have hostnames ending with `.edu`, and we call this category of hostnames the *edu domain*.

With over seven million hosts in the edu domain (as of early 2001), subcategorization is necessary. The next rightmost part is sometimes called the second-level domain and specifies which educational institution within the edu domain this host belongs to—in this case, `washington` corresponds to the University of Washington (UW). Every college and university with computers on the Internet has a special name that identifies

its group of hosts within the edu domain (e.g., `umich` for the University of Michigan, `ucla` for the University of California at Los Angeles).

A typical educational institution has so many hosts that yet another level of subcategorization is often used, usually based on academic department. The next part of the hostname, `cs`, indicates that the host is in the Computer Science department. Other departments at UW and their corresponding third-level domain names are Music (`music`), Physics (`phys`), Astronomy (`astro`) and School of Nursing (`son`).

Finally, the `www` part of the hostname specifies a particular computer within the domain `cs.washington.edu`. In this case, `www` is the name of the computer that serves web pages about the Department of Computer Science and Engineering at UW. It is typical for computers that serve web pages to be named `www` within their domains. We will learn more about what a web server does later in this lab.

All of these levels of subcategorization should make it easier to remember `www.cs.washington.edu` than to remember the corresponding IP address, `128.95.4.112`. We will start with a few small experiments with hostnames and IP addresses, to learn more about how they correspond to each other.

DNS to the rescue! DNS makes life on the Internet more than just convenient. Because DNS translates hostnames to IP addresses, the IP address corresponding to a particular hostname can be changed and updated in DNS, allowing users to continue accessing the host by hostname without even being aware of the address change. In late July 2001, network security experts did exactly this to head off a massive, carefully coordinated virus attack on the White House web site by a virus called “Code Red.” A *virus* is a program that secretly copies itself onto a computer (usually via files transferred over a network or floppy disk) and performs unintended, often malicious, actions. Code Red was designed to rapidly spread across the Internet and wait until 5:00 pm Pacific on 19 July, at which time, every infected host (est. over 225,000 worldwide) would deluge the web server at IP address `198.137.240.91` (`www.whitehouse.gov`’s IP address at the time) with data, effectively preventing anyone else from accessing it (a “denial of service” attack). White House network administrators acted fast, though, and before the coordinated attack began, they switched the web server’s address to `198.137.240.92`, outsmarting the virus by “moving the target.”

1. *Open a web page by hostname.* Start a web browser and open the URL `http://nature.org`, the home page of the Nature Conservancy.
2. *Find out the IP address corresponding to a hostname.* Your instructor will provide you with the URL of a web page called an “nslookup gateway,” which provides something like directory assistance for IP addresses. An nslookup gateway can translate a hostname into its corresponding IP address(es). Open a new browser window with the nslookup gateway URL. Use this page to find out what the IP address of `nature.org` is and write it down below.

What do you expect will happen if you point your browser at `http://x.x.x.x`, replacing the `x.x.x.x` with the IP address you wrote down above?

3. *Open a web page by IP address.* Check your guess by opening a third browser window with the IP address URL for `nature.org`. Does the page appear to be the same as or different from the page at `http://nature.org`?

Normally, when you are using the Internet, you only have to remember hostnames and can forget the IP addresses they correspond to. It is still important, however, to understand that IP addresses are being used “behind the scenes” to interpret DNS-related error messages you might encounter as you use network software.

Part 2. Servers and Clients on the Web

You have probably already heard the terms “web server” and “e-mail client,” but you might not realize that “server” and “client” are general terms describing roles that computers can play on a network, depending on what software they are running. In general, a *server* is a computer that provides some kind of data (e.g., web pages, database entries) or service (e.g., e-mail, printing), and a *client* is a computer that requests and receives the data or service. To illustrate this difference, we will begin by discussing an example with web pages.

When you view a web page on your computer, your computer is acting as a *client* to a *web server* somewhere on the Internet, the computer where the web pages are stored. Each time you click on a link, your web browser sends a web page request through the network to the web server, and the server responds by sending a copy of the requested page back to your computer, where the browser displays it for you.

Where to find a web resource. At the heart of each of these requests is a *URL* (short for Uniform Resource Locator), which is a standard way of specifying a file or resource on a particular computer on the network. Although URLs can be used to specify many kinds of network requests, since they are most commonly used for web pages, URLs are also called “web addresses” or “links.” URLs seem to appear everywhere now, from advertisements to local television news broadcasts and even boxes of breakfast cereal, as companies and other organizations set up web sites to accompany traditional media materials.

Every URL includes three important pieces of information: what kind of request it is (usually for a web page), the hostname of the server the request is going to, and the location and name of the file being requested. Suppose you are trying to view the web page at this URL on your computer:

```
http://www.pcwebopedia.com/TERM/s/server.html
```

The first part of the URL is before the `://` and, in this case, `http` indicates that this is a request for a web page. *HTTP* stands for Hypertext Transfer Protocol, where “hypertext” is a technical term for text that includes links to other documents, and is the standard method of transferring web files through the Internet.

One useful way of thinking about the last two parts of the URL is to interpret them together as a pathname for a web page file. One important difference between URLs and pathnames is that a URL must specify which computer the file is on, something which is just assumed in the case of a pathname. The hostname is specified between the `://` and the next `/`, and the remainder of the URL is just like a pathname—it specifies the location of the file on the given host, with slashes separating folder names. In some cases, the filename at the end of the URL can be omitted (e.g., `http://nature.org`), and the web server assumes that a file with a standard name like `index.html` or `home.html` is being requested.

To view the page at the URL given above, your web browser sends a web page (HTTP) request to the host `www.pcwebopedia.com`. If this computer is properly set up as a web server, it is running software that listens on the network for these requests and will respond by sending back the appropriate web page—in this case, `server.html` in the folder `TERM/s/`.

A single computer can act as more than one kind of server by running more than one kind of server software at once. In fact, it is common for most hosts to be playing the role of at least a few different servers, e.g., web, mail, printing, and file storage. In the next part of the lab, we will see how another kind of server can be used to copy files between computers over the network.

Is the server the computer or the software? The term “web server” can be confusing, because it is often used in two different ways. The term commonly refers to a computer which is running software that enables it to send web pages on request, as in, “My home page is on the web server `students.washington.edu`.” However, some people also use the term “web server” to refer to this software, rather than the computer, as in, “If you’re running a Microsoft web server on your computer, you should regularly check for security problems.” Context usually disambiguates, but not in some common cases where it is not. In this lab, we use the term in the sense of a computer or a role that it plays, rather than the software, which we call “server software.”

Part 3. Copying Files Across the Network with FTP

FTP (short for File Transfer Protocol) is a standard method of sending files between computers through a network. The primary difference between FTP and HTTP is that using FTP requires that you identify yourself with a *user name* (also known as a *login*) and a *password*. (Imagine if HTTP were like this. You would have to type in a user name and password every time you clicked a link!) In this part of the lab, you will use

FTP to put your first file on the web, i.e., use FTP to make a file publicly available via HTTP.

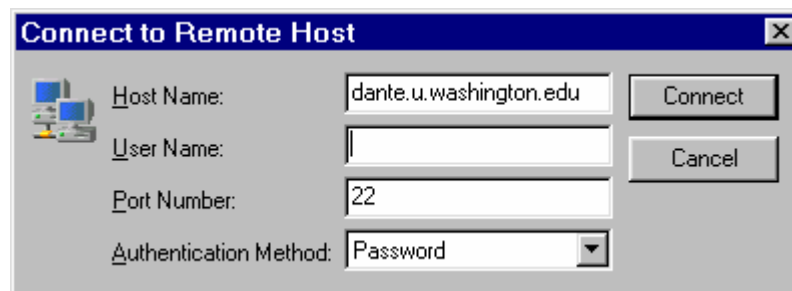
An FTP server is a lot like a bank for files, i.e., a computer on the network with lots of hard disk space that offers individual users access to private storage. Just like with a hard disk on your own computer, you can copy, delete and rename files on an FTP server, as well as maintain folders to keep your files organized. You use FTP client software to connect with an FTP server, just as you use a web browser (web client software) to connect with an HTTP server (web server).

We learned that web pages are stored and delivered from web servers, but how do web pages get to a web server in the first place? Sometimes, web authors create and edit them on the web server directly, but this is not the usual case. Typically, a web author edits a page on their own computer, makes sure the page looks right, then transfers a copy to the web server. As soon as a file is transferred to a web server, it generally becomes publicly available, so working on their own computer helps ensure that there are no mistakes in the pages that actually go on the web, or “go live,” as some people say, borrowing the expression from broadcast media. In this part of the lab, you will follow the same process to put your first file on the web. Writing even a simple web page is worth an entire lab in itself (Lab 1.3, in fact), so you will start by posting a plain text file in this lab.

4. *Create a text file to be posted on the web.* Just as you did in Lab 1.1, use Notepad to create a text file with whatever content you wish to post to the web. You might consider writing a list of your favorite bands or a short autobiography. First, save your file on your floppy disk. Insert your floppy disk in the drive, select **File \ Save As...**, and switch to your floppy disk (usually drive A). You might want to create a separate folder called “web” to keep this file separate from your other work.
5. *Open an FTP connection to your FTP server.* Start **SSH Secure File Transfer**, an ftp client program. If there is no icon for it on the Desktop, your instructor will tell you where you can find it in the Start menu.



Open a new connection to your FTP server. Your instructor will tell you the server’s hostname.



Shortly after you click **Connect**, you will need to enter your password. (You might first see a dialog titled, “Host Identification.” Double-check that you have the hostname right, then click **Yes**.) The SSH ftp window should look a lot like Explorer, with the left pane showing a folder hierarchy and right pane showing files in the currently selected folder. Unlike Explorer, however, you are now looking at folders and files on the ftp server, not on the computer you are working on. The sample screenshot below shows the window for a connection to host `dante.u.washington.edu` as user `yasuhara`.



local vs. *remote* – “Local” is used to describe anything that is on the computer you are currently working on, physically. In contrast, “remote” describes a resource that is physically elsewhere but accessible to you via the network. That explains why the dialog shown above has the title “Connect to *Remote* Host.”

6. *Open the web folder on the ftp server.* Your space on the ftp server is set up with a folder with a special name whose contents can be made accessible via the web. This “web folder” is usually named `public_html`, but your instructor will tell you what the actual name is for your campus network setup. You should see this folder in your ftp space as shown in the sample screenshot above. Select the web folder by clicking it in the left pane or double-clicking it in the right pane to open it.
7. *Upload your text file into the web folder.* The process of sending a copy of a file to a remote server is called *uploading*. (*Downloading* is the process of retrieving a copy of a file from a remote server to your local disk.) To start an upload, select **Operation \ Upload...** and select the text file you created in Step 4. When the transfer is complete, you should see the file in the right pane of the SSH ftp window.

ALTERNATIVES: You can start an upload by pressing **Ctrl-U** or clicking the toolbar button with an up arrow on it.

8. *Use a web browser to verify that the file is available on the web.* Your instructor will give you an example URL like the one below. Based on this example, you should be able to form a URL for the file you just uploaded.

```
http://students.washington.edu/username/
```

Your URL will be different from the example above (at least the hostname will be different from `students.washington.edu`), but your user name will come somewhere after the hostname. Any files that you place in your web folder will be accessible by adding the filename to this URL. Extending the example above, if user `yasuhara` uploaded a file `autobio.txt` to his web folder, the file could be accessed via the web at this URL:

```
http://students.washington.edu/yasuhara/autobio.txt
```

Leave SSH ftp open for now, and open a web browser window with the URL of your newly uploaded file. You should see the file's contents in your browser window. Write down the URL for your text file below:

TROUBLESHOOTING: If you are not able to access your text file using your web browser, do not panic. Start by identifying what kind of problem the server is having. Check the title bar of your browser window for an error code number and a brief description of the error. Here are some tips for dealing with the more common errors:

“404 Not Found” – Your URL specifies a file that does not exist on the server. Double-check all parts of the URL, especially the hostname, your user name, and the filename itself.

“Permission Denied” – The server has located the file, so your URL is correct, but the file's properties on the server are set such that it is not publicly readable. See the next step for how to fix this.

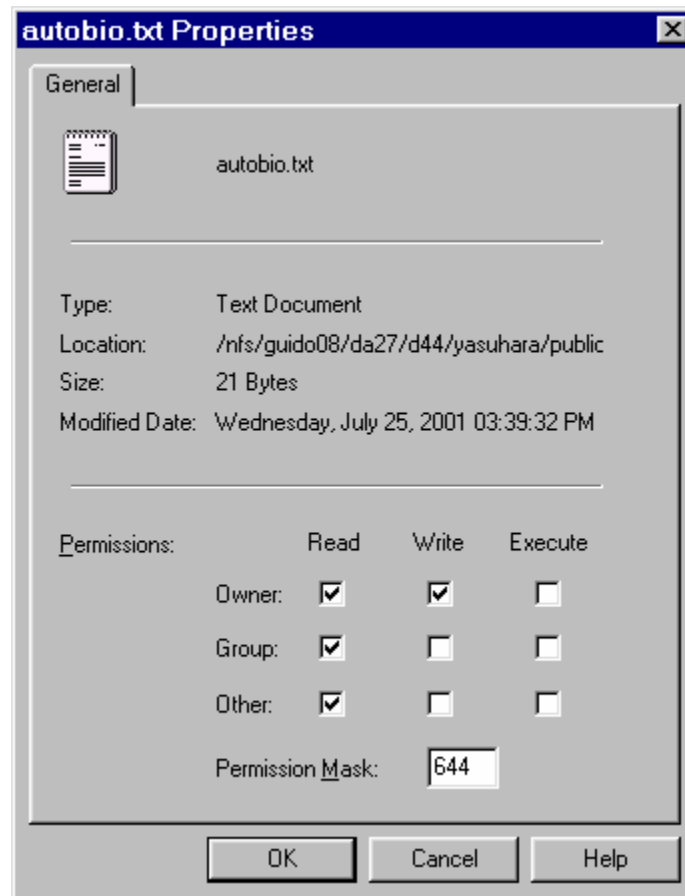
“Cannot Find Server or DNS Error” – The hostname part of the URL is probably mistyped. Recall that DNS is the system by which hostnames are converted into IP addresses. If you request a host whose name is not registered with DNS, you will get this error.

9. (OPTIONAL) *Set access permissions on the file so that it is publicly accessible.* One more step might be necessary before the file can be accessed via the web. Depending on how your ftp server is set up, any file you upload to your server might have its properties set such that only you can read and modify the file. (These default settings are good for protecting your privacy. You get to decide what to make publicly accessible on a per-file basis.)

To view and adjust a file's properties using SSH ftp, select the file in the right pane and select **Operation \ Properties** or right-click the file and select

Properties. This will bring up a “Properties” dialog that looks similar to the Explorer “Properties” dialog you saw in Lab 1.1. Under the **Permissions** section, make sure **Read** is checked for **Owner**, **Group**, and **Other**, as shown in the example below. This sets permissions on the file such that all users are allowed to read (view) this file. However, as long as **Write** is checked for only **Owner**, only you have the ability to change the file. Click **OK** after verifying these settings.

Now, go back to your browser and click the **Refresh** button (or press **Ctrl-R**) to try the URL again.



10. *Explore the other features of SSH Secure File Transfer.* Just as with Windows Explorer, you can delete and rename files and create subdirectories on the ftp server using the SSH ftp. Most of these tasks can be done using the **Operation** menu or right-clicking the file you want to work with.

Try renaming your text file on the ftp server now. Switch back to your web browser window. What happens when you refresh your browser without entering a new URL?

What do you need to change the URL to in order to view your file?

Back in the SSH ftp client, create a folder called “text” in your web folder. (Make sure your web folder is currently open and select **Operation \ New Folder** or press **Ctrl-N**.)

Copy your text file into this new folder by dragging it to the folder. Verify that a copy of the file is in the “text” folder by opening it to see what files are inside.

What is the URL for the file in the “text” folder?

Verify the URL using a web browser.

Optional Additional Activities

- Using SSH ftp, upload an image file (e.g., a GIF or JPG file) to your web folder, figure out its URL, and view it using your web browser.
- Using SSH ftp, try downloading a file from your web folder. Suppose that sometime after you uploaded your text file, you accidentally lost it. You could always retrieve the copy that you put on the ftp server by downloading it. To download a file using SSH ftp, select the file and select **Operation \ Download...** or click the toolbar button with a down arrow on it. Then, choose a local folder to store the downloaded file in and click **Download** to start the transfer.

Further Reading

- A brief history of the Internet is on the web at this PSB web page:
<http://www.pbs.org/internet/timeline/timeline-txt.html>
- The Wikipedia, a publicly maintained, free, web-based encyclopedia, has a page about the Internet, too:
<http://www.wikipedia.org/w/wiki.phtml?search=internet&go=Go>
- Discovery Channel Canada features a timeline of computer virus events:
<http://exn.ca/Stories/2000/05/04/57.asp>
- Carnegie Mellon University’s Software Engineering Institute maintains an extensive web site that tracks security issues (including but not limited to viruses). Their CERT Coordination Center was the first computer security incident response team and now serves as an authoritative source for security bulletins.
<http://www.cert.org/>
- Chapter 12 (Using Computers in Polite Society) discusses viruses in more detail, including the story of another relatively recent and widespread virus, Melissa.