
LIS 534 LAB 1: 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 much of HTML, which is left for more focused coverage in future labs. This lab assumes basic experience using a web browser and e-mail.

This lab covers topics that are not exactly part of this course, but is important to prepare you for working with the technical side of web design that is covered in this course. So, feel free to ask a friend to help you do this lab. The goal of this lab is to help you understand the client-server structure of the Web so that you know how FTP works and how to publish web pages so that anyone with an Internet connection and a browser can access them.

Vocabulary

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

- host, IP address, hostname
- 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. 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?
2. Is it correct to say that folders/directories directly contain data? Why or why not?
3. 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. Domain, Hostnames and IP Addresses

Hostnames instead of IPs. 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 millions of hosts in the edu domain, 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 external 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://www.asis.org`, the home page of the Nature Conservancy.
2. *Find out the IP address corresponding to a hostname using an “nslookup gateway.”* An nslookup gateway provides something like directory assistance for IP addresses, and can translate a hostname into its corresponding IP address(es). Google “nslookup gateway” and pick any website from the search result to find out what the IP address of `nature.org` is and write it down.

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 `www.asis.org`. Does the page appear to be the same as or different from the page at `http://www.asis.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 www.ualberta.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 lab, you will use FTP to put your first files 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).

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 create a simple web page and put your first file on the web. You will try to understand the web page creation and improve your page in later labs. The focus here is more on the process of getting your web pages published onto the web.

4. *Create a HTML file to be posted on the web.*

Open Notepad and copy and paste the following into the file:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My first web page</title>
  </head>
  <body>

  </body>
</html>
```

Save the file on the Desktop. Select **File \ Save As...**, and make sure that you choose “all files” as the file type, and put the extension .html at the end of the file name. You might want to create a separate folder called “myWeb” to keep this file separate from other files.

Add whatever content you wish to the page (between <body> and </body>) to post to the web. You might consider writing a list of your favorite bands or a short autobiography. Remember to save your file.

5. *Open an FTP connection to your FTP server.*

At the University of Alberta, you will be using a secure FTP client to access your GPU (General Purpose Unix) account - **WinSCP**. For Mac users, **Fetch** is one of the programs to use for this purpose. Both are free for education purposes.

You have a CCID (Campus Computing ID) and password, which also means you have 1 GB server space on the University’s GPU account. Your GPU account is used for not only posting documents to your Web space, allotted to you by the University, but is a convenient way of transferring files from home to school or between computers at school.

Think of it like an external storage place for your files, accessible from any computer that has a secure FTP client.

The secure FTP tool that the University of Alberta recommends for accessing your GPU account is WinSCP. If you are using the computers on campus, WinSCP is already installed. You can also download this program and install it on your home computer.

When you open WinSCP you will automatically be shown the “Stored Sessions.” Click on “New” in the upper right of the WinSCP window. This will bring you to a window in which you need to enter some information.

The host name for accessing GPU accounts at UofA is: `gpu.srv.ualberta.ca`. (Without a period at the end). Your login and password are the same as your CCID. All other fields like the port ID and protocol remain as the defaults. You then click on “Login” and a different window divided into a right and left side appears.

The right side of that window shows you the files on your GPU account while the left side of that window shows you the files on the computer you are currently using. To move any other documents (like Word documents, etc.) to your GPU space so you can access them from another computer later, just drag them into the space on the right side of the window. To move files from your GPU account to the computer you are using, you can simply drag the files out of the right side of the window to the desktop or you can drag them into the left side of the screen to an appropriate place on that computer’s hard drive.

To post a document to the Web so that anyone with an Internet connection and a Web browser can view it, you need a “web folder” with proper access permissions. In the UofA system like in many other systems, this folder is named as “public_html”. You should already have this folder created and proper permissions set if you attended the IT orientation session as part of the new students orientation each year. For those who do not have this folder yet, follow the instructions found on the IST website to create it and to set the permissions. Feel free to ask a friend to create it for you if you cannot figure it out yourself, as this is not part of this course.

Once you have this folder on your GPU space, to publish a document to the Web, you just drag that document into your public_html folder. To view the document from any computer on the Internet, one just needs to point the browser at **`http://www.ualberta.ca/~YourCCID/DocName`**. (e.g. `http://www.ualberta.ca/~dzhao/dzPublications.html`)

You can find another explanation of using **WinSCP** to access your GPU account from a campus computer, complete with screen shots, on the University of Alberta Information Services and Technology website.

For Mac users, **Fetch** is one of the secure FTP clients you can use. Fetch is a full-featured FTP client for Mac OS X made by Fetch Softworks. In addition to basic FTP functionality, “Fetch includes such features as editing files without having to download them and re-upload them.”

To install Fetch on your computer, follow the steps below.

- a) Download fetch from <http://fetchsoftworks.com/fetch>.
- b) Double-click the **Fetch5.7.dmg** file.
- c) Double-click the **Fetch folder** and then drag the icon to the Applications folder (or any folder of your own desire).

To connect to the University of Alberta Server, follow the steps below.

- a) Open Fetch.
- b) From the File menu, select New Connection.
- c) Enter the Hostname (`gpu.srv.ualberta.ca`), Username and Password are your CCID username and password.

- d) Select SFTP from the Connect using drop-down list. **Note:** Once SFTP is selected as the security type, the "Enable encryption" checkbox is activated. By selecting SFTP, you are ensuring that your session is secure.
- e) Click connect.

Once you are logged into your GPU space, browse to the public_html folder. You can upload files by dragging from Finder or your desktop into the folder trees of Fetch window, or you can use the "Put" feature to browse for files in your local machine.

- a) Click Put on the Fetch window.
- b) The "Choose a File" dialog box is displayed. Select the file or folder that you want to transfer and click Put.
- c) The selected file is transferred. Continue to select and transfer files or select Fetch > Quit Fetch to close the connection and exit the program.

Downloading files from the server is similar to uploading files. You can either drag the file you want to download from the server (public_html folder in the fetch window) to your desktop (or any other folder in your computer), or use the "Get" feature. Click on the Path button to choose which folder you want to go to, and the New Folder button to create a new subfolder in the current folder.

6. *Explore the other features of WinSCP or Fetch.* Just as with Windows Explorer or Finder, you can delete and rename files on the ftp server using WinSCP or Fetch. Most of these tasks can be done by right-clicking the file you want to work with.

Try renaming your 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 WinSCP or Fetch, create a folder called "LIS534" (no space) in your web folder. Copy your HTML file into this new folder by dragging it to the Desktop and then from there to this folder. Verify that a copy of the file is in the "LIS534" folder by opening the folder to see what files are inside.

What is the URL for the file in the "LIS534" folder? Verify the URL using a web browser.

Assignment

Create a webpage to organize your assignments for this LIS 534 course.

1. Go to <http://www.ualberta.ca/~dzhao/courses/534/studentsAssignments.html>, and download the webpage to your desktop. Rename it as index.html.
2. Open the index.html with a text or HTML editor (e.g., NotePad). Change its content to something that shows clearly that this webpage is yours for organizing your assignments for this LIS 534 course. Save your file.
3. Upload the index.html file into the LIS534 folder that you created on your U of A web space earlier in this lab.

4. View it in a browser (what is its URL?). If you want to make changes to it, change the copy on your local computer, then go back to step 3 above.

Optional Additional Activities

- Using WinSCP, 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 WinSCP, 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 WinSCP, select the file and select Operation \ Download..., or simply drag it to the Desktop. Then, choose a local folder to store the downloaded file in and click Download to start the transfer.

Acknowledgements

This lab is a modified version of Lab 1.2 of the following book:

Snyder, L. (2007). *Fluency with Information Technology: Skills, Concepts, and Capabilities*. Boston: Addison Wesley.

I would like to thank my graduate assistants over the years for their help improving the labs in various ways.