CHAPTER 26

Lab Assignments for Chapter 26

We have created six lab assignments for this chapter: Lab26-1 to Lab26-6. We have also included six lab-report sheets, which means that each lab assignment needs to be reported in a separate sheet. The six lab assignments are related to six application-layer protocols we discussed in this chapter. The lab assignment for SSH protocol has been moved to Chapter 32 because it involves security issues. It is strongly recommended that the student carefully study and digest the corresponding protocols before working on the related lab assignment. It is assumed that you have done the lab assignment for Chapter 1, which told you how to install the Wireshark software and how to use it.

26.1 LAB26-1: HTTP

HTTP is used to handle web pages. In this lab, we retrieve a web page and then, using Wireshark, capture HTTP packets. We learn about the two most common HTTP messages (GET and response). We also examine the role of browser caching.

26.1.1 Assignment

The assignment for this lab has three parts.

Part I: General

- Start your web browser and clear the browser's cache memory, but do not access any website yet.
- Open Wireshark and start capturing.
- Go back to your web browser and retrieve any web page that contains embedded objects (pictures, logos, etc.).
- Since the browser's cache memory has been cleared, the web page is retrieved from the original destination. Type **http** (lowercase) in the filter field of the Wireshark and click **Apply** so that only HTTP messages are displayed.
- After enough packets have been captured, select the **Capture** from the pull-down menu and select **Stop** to stop capturing. The packet list pane of the Wireshark should now display many HTTP packets.

Questions

Using the captured information, answer the following question in your lab-report sheet.

1. What is the source IP address of the first GET message?
2. What is the destination IP address of the first GET message?
3. What is the source IP address of the first response message?
4. What is the destination IP address of the first response message?
5. How the source and destination addresses in the first response message are related to those in the first GET message?
6. Using the time stamps of a GET message and that of the corresponding response message, determine how long it took from the time the GET message was sent until the response message was received. By default, the value of the time column is the amount of time in seconds since Wireshark tracing began.
7. From one of the messages, determine the HTTP version.
8. From the first GET message, determine the URL of the website.
9. From the first GET message, determine the *user agent*.
10. Using the first GET message, determine the *medium format*, the *language*, the *encoding*, and the *character set* that the client can accept.
11. What are the *status codes* for the first response message? Check the status code table to see the descriptions of this code.
12. Record the *etag* (identity tag) of the first response message. What is the application of *etag* in conditional request in HTTP.
13. What is the value of the *content-length* field of the first response message?

Part II: Embedded Objects

Most web pages contain pictures, logos, and so on, in the form of embedded objects. When you open any of these pages, embedded objects are retrieved from the same web-site or a different website. In this part, we extract information about these embedded objects in the captured file.

Questions

Using the captured file in Part I of the assignment, answer the following questions in your lab-report sheet.

1. Checking your browser, how many embedded objects are in the page?
2. How many GET messages sent by the browser to retrieve the embedded objects?
3. What is the URL of each embedded object?
4. Has the HTTP used persistent or non-persistent connection? Explain your answer.

Part III: Browser's Cache Memory

To reduce the response time and internet traffic, most bowers keep the recently retrieved HTTP objects in their cache memory. When the browser receives a request to retrieve a web file, it first checks its cache memory. If it has the file, it sends a condi-

tional GET (IF-Modified-Since) request. The server sends the file if it is modified; otherwise, it sends a "Not Modified" response.

Open the Wireshark and start capturing. Go to your browser and retrieve the same web page again by clicking reload or refresh bottom on your browser. This time the page is retrieved from the cache memory. Type **http** (lowercase) in the filter field of the Wireshark and click **Apply** so that only HTTP messages are displayed. Stop the Wireshark and save the captured file.

Questions

Using the captured file, answer the following question in your lab report.

1. What is the value of the content-length field of the response message?
2. Explain the answer to the first question.

26.1.2 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-1 report sheet that contains answered questions.
2. A printout of the supporting captured information.

26.2 LAB26-2: FTP

File Transfer Protocol (FTP) is a standard client-server protocol for copying files from one host to another over the Internet. You can run the FTP program using command-line.

ftp [*options*] [*hostname*] **// You may want to use the -v option.**

After the connection is established to the remote site (*hostname*), you can use one of the available commands. The list is long, but Table 26.1 gives a partial list that you may need for doing this assignment. Note that the arguments in brackets are optional.

Table 26.1 *Some FTP Commands*

Command	Argument	Description
bye		Terminate **ftp** session and exit.
cd	[*remote-dir*]	Change the working directory to *remote-directory*
close		Terminate **ftp** session and return to command interpreter.
delete	[*remote-file*]	Delete *remote-file*.
dir	[*remote-dir*] [*file*]	Print a listing of the contents of the directory and place result in *local-file*.
get	remote-file [local-file]	Retrieve the *remote-file* and store it on the local machine as the *local-file*.
help	[command]	Print help information for the *command*.
ls	[*remote*] [*local*]	Display the contents of the *remote* directory and store in the *local* file.
mkdir	*directory*	Make a *directory* on the remote machine

Table 26.1 *Some FTP Commands*

put	*local-file* [*remote-file*]	Store a *local-file* on the remote machine.
pwd		Display the working directory
quit		Same as **bye**.
rename	[*old*] [*new*]	Change the file name (*old* to *new*).
rmdir	[*dir-name*]	Delete a directory on the remote machine
size	*filename*	Return the size of *file-name* on remote machine.

Anther solution is to use a graphical user interface. There are several GUI for FTP. One that we mention here is FileZilla. The client FileZilla can be downloaded (free) from the following website:

http://filezilla-project.org/download.php

Figure 26.1 shows an example of FileZilla interface window.

The window is divided into two panes. The left shows the local site; the right shows the remote sites. You can graphically execute FTP commands by applying familiar GUI actions (for example, you can hold the mouse on a file or a folder, right click, and select download to retrieve it from the remote server) or you can execute them by entering custom command which you can find under the Server or Transfer dropdown menu.

26.2.3 Assignment

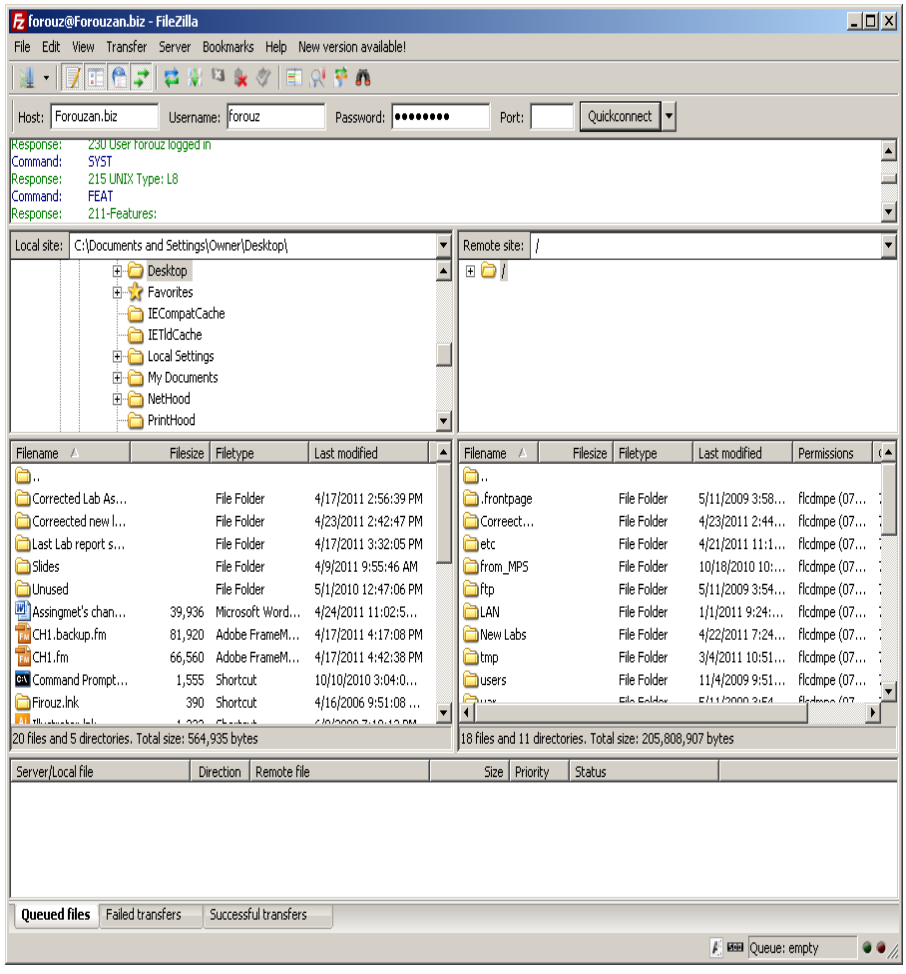
- Open Wireshark and start capturing.
- Open the FTP client from the command-line utility, or open FileZilla. Connect to the remote site and execute several FTP commands.
- After quitting the FTP program, go back to the Wireshark, select the Capture from the pull down menu and click **Stop**. Select the File menu and save the captured file. You will notice several FTP packets and FTP/DAT packets as well as many TCP packets that are used to establish and terminate FTP connection and acknowledge the transfer of FTP data.

Questions

Examining the above-mentioned packets and consulting Tables 26.4 and 26.5 in the textbook, answer the following questions in your lab-report sheet.

1. Identify the three TCP packets used for connection establishment.
2. What is the first FTP packet? Is it a request or response packet?
3. How many packets are exchanged for login? What is the exchanged information?
4. Is the user name and password encrypted? What can you say about security of FTP?
5. What request command is used in any of the following activities? What response code is used in any of them?
   a. storing a file,

Figure 26.1 *FileZilla Window*



b. retrieving a file,

c. getting the list of files and folders in a folder,

d. deleting a file,

e. renaming a file, and

f. quitting

6. What well-known port number does the server used for control connection?

7. In active mode, the FTP data connection is opened by the server from its well-known port 20 to a negotiated (ephemeral) client port. In passive mode, the data connection is opened by the client from an arbitrary (ephemeral) port to a negoti-

ated (ephemeral) server port. What is the server port number for the data-transfer phase (FTP-DAT)? Is the data connection opened by the server or by the client?

8. Identify the TCP packets used to terminate the connection.

26.2.4 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-2 report sheet that contains answered questions.
2. A printout of the supporting captured information.

26.3 LAB26-3: TELNET

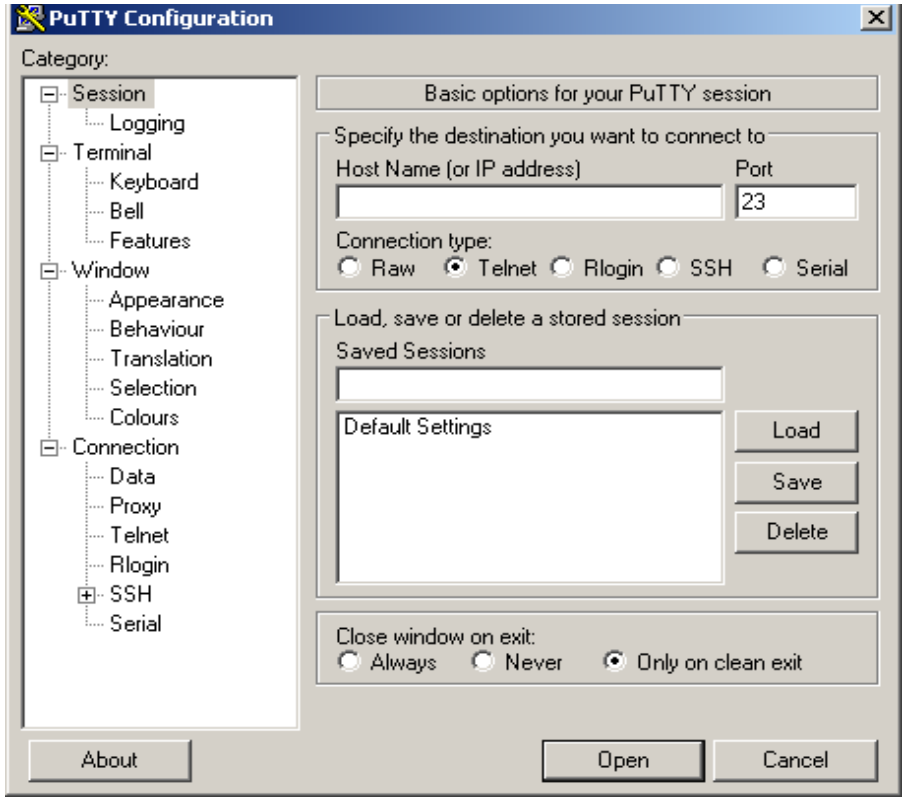
Using a virtual terminal connection, TELNET provides remote logging, an interactive text-oriented communications between a user and a remote server, over TCP. The best way to capture packets exchanged between your computer and a remote host is to use a client GUI TELNET program. We recommend to use PuTTY, which is a free software and can be downloaded from

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Save the *putty.exe* file on your computer (no installation needed). Figure 26.2 shows a sample of PuTTY configuration window.

To access a remote site (which you need to have account and password), type the host name or the IP address of the remote host in the corresponding box and choose **Telnet** as the connection type (using the corresponding radio button). Then click the **Open** button to connect to the site. After the connection has been established, type your username when prompted for it. Then type your password when prompted. When you see the prompt for the Unix or Linux (for example, $), the remote host is ready to

Figure 26.2 *PuTTY Window*



accept your commands. There is a long list of Unix or Linux commands, but we give only a few of them in Table 26.2 if you are not familiar with them.

Table 26.2 *Some Unix or Linux Commands*

Command	Argument	Description
cal	[*month*] [*year*]	Display a calendar for the given *month* and *year.*
cat	[*files*]	Display the contents of *files*.
clear		Clear the terminal display.
cp	[*file1*] [*file2*]	Copy *file1* to *file2*.
cp	[*files*] [*directory*]	Copy *files* to a *directory.*
date		Gives the date and time.
head	[*files*]	Displays the first ten lines of *files*.
ls		Give the list of directories.
mkdir	[*directories*]	Create new *directories*.
mv	[*source*] [target]	Move a file from *source* to *target*.

Table 26.2 *Some Unix or Linux Commands*

pwd		Gives the name of the current directory.
rm	[*files*]	Delete (remove) *files*.
tail	[*files*]	Display the last ten lines of *files*.

26.3.5 Assignment

- Open Wireshark and start capturing.
- Start a TELNET client session from PuTTY. Connect to the remote site, login, and execute a few Unix/Linux commands.
- Close the TELNET connection (quit) and return to Wireshark. Stop capturing and save the captured file.

Part I: General

Examining the captured file, you will notice several TELNET packets as well as many TCP packets that are used to establish and terminate TELNET connection.

Questions

Using the captured information, answer the following question in your lab-report sheet.

1. Identify the three TCP packets used for connection establishment.
2. Identify the TCP packets used for connection termination?

Part II: Negotiation

Before (or during) a session the TENET client and server negotiate some options and suboptions. Some common options are Binary, Echo, Terminal type, Terminal speed, and Line mode (Check the Internet to learn about more TELNET options and suboptions). TELNET allows each party (sender or receiver) to enable or disable an option.

- **Offer to Enable**. The sender can use *WILL* command to offer enabling an option. The receiver can agree with *DO* response or disagree with *DONT* response.
- **Request to Enable**. The sender can use *DO* command to request the receiver to enable an option. The receiver can agree with *WILL* response or disagree with *WONT* response.
- **Offer to Disable** The sender can use *WONT* command to offer disabling an option. The receiver must agree with the *DONT* command.
- **Request to Disable** The sender can use *DONT* command to request disabling an option. The receiver must agree with the *WONT* command.

Some options negotiation require additional information that are called suboptions negotiation. For example, to define the Terminal type or speed, the negotiation must include a string or a number.

Questions

Using the captured information, answer the following question in your lab-report sheet.

1. Check packets that have option negotiation. What options are negotiated?
2. Is the option offered or requested by the sender?
3. What is the response of the receiver to option negotiation?

Part III: Session

After negotiation, the TELNET session starts.

Questions

Using the captured information, answer the following question in your lab-report sheet.

1. How many packets are exchanged for remote login? What is the exchanged information?
2. Is the user name or password encrypted? What can you say about the security of TELNET?
3. What packet is used to terminate the TELNET session?
4. Select **Analyze** from the drop down menu and then click **Follow TCP stream**. What are the contents of the Follow TCP stream window?

26.3.6 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-3 report sheet that contains answered questions.
2. A printout of the captured information.

26.4 LAB2-4: SMTP

SMTP (Simple Mail Transfer Protocol) is a TCP/IP protocol for transmitting e-mail messages across the Internet. SMTP, however, is a push protocol; it can deliver e-mails up to the recipient server mail box. A pull protocol, either POP3 or IMAP4, is needed by the e-mail recipient, for the last stage of the e-mail delivery.

In this lab we use Wireshark to examine packets exchanged by SMTP protocol. Open your e-mail user agent (Microsoft Outlook, Eudora, …) and set the e-mail securities options off. Make sure that the outgoing e-mail does not require an encrypted connection. After capturing the SMTP packets, you can set the security options back if you want.

26.4.7 Assignment

- Compose an e-mail and address it to yourself, but do not send it yet.
- Open the Wireshark and start capturing.
- Go to your e-mail user agent and send the e-mail.
- In the Wireshark window, type **smtp** (lowercase) in the filter field and click **Apply** so that only SMTP packets are displayed.
- Stop capturing and save the captured file.

Questions

Using the captured information, answer the following question in your lab report.

1. All SMTP packets have the same two IP addresses. Which one is the IP address of your computer?
2. Which host does the other IP address represent?
3. All SMTP packets have the same two port numbers. Which one is port number of the SMTP client process?
4. In which range is the client port number (see Figure 23.5 in the textbook)?
5. What is the port number of the SMTP server process?
6. Examine the SMTP commands or SMTP response codes in each SMTP packet to determine its significance. (You can find description of most of these request or response commands in Tables 26.6 and Table 26.7 in the text. You may guess the meaning of the ones that are not in these tables or you can look them up in the Internet.
7. There is an IMF packet that is encapsulated inside a SMTP packet. What is the content of this packet? Do an Internet search to find the scope of IMF.

26.4.8 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-4 report sheet that contains answered questions.
2. A printout of the captured information.

26.5 LAB2-5: POP3

In this lab we use Wireshark to examine packets exchanged by Post Office Protocol, version 3 (POP3). Open your e-mail user agent (Microsoft Outlook, Eudora, …) and set the e-mail securities options off. Make sure that the incoming e-mail does not require an encrypted connection. After capturing the POP3 packets, you can set the security options back if you want.

26.5.9 Assignment

- Compose an e-mail and address it to yourself, but do not send it yet.
- Open the Wireshark and start capturing.
- Go to your e-mail user agent and send the e-mail.
- In the Wireshark window, type **pop** (lowercase) in the filter field and click **Apply** so that only POP3 packets are displayed.
- Stop capturing and save the captured file.

Questions

Using the captured information, answer the following question in your lab report.

1. All POP packets have the same two IP addresses. Which one is the IP address of your host?
2. Which host does the other IP address represent?
3. All POP3 packets have the same two port numbers. Which one is the port number of the POP3 client process?
4. In what range is this port number?
5. What is the port number of the POP3 server process?
6. Examine the content of each of the POP3 packets to determine its significance.

26.5.10 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-5 report sheet that contains answered questions.
2. A printout of the captured information.

26.6 LAB26-6: DNS

There are several network administration tools for Microsoft Windows and UNIX-like operating systems that are useful for network troubleshooting as well as for educational purposes. Among these tools are the following:

- **dig** (Domain Information Groper) is used for querying DNS servers. This utility replaces older tools such as *nslookup*.
- **ipconfig** (Internet Protocol Configuration) for Windows or **ifconfig** (Interface Configuration) for UNIX-like operating systems is used to configure, control, and query TCP/IP network interface parameters.

26.6.11 Assignment

This lab is made of three parts. In Part I, we use the dig utility. In Part II, we use ipconfig utility. Finally, in Part III, we use Wireshark to find more information about the packets exchanged by the DNS protocol.

Part I: Using dig

The **dig** utility is used for querying DNS name servers for any DNS records. If you do not have this utility in your system, you can download the command line interface (CLI) version of this utility from the following website.

http://members.shaw.ca/nicholas.fong/dig/

The installation and setup instruction is on the same website. You can find the dig manual page on the website below:

https://www.isc.org/software/bind/documentation/arm95#man.dig

Alternatively you can run *dig* (and many other command line interface utility tools) from many websites who offer this service for free. For this lab, we run *dig* from the following website:

http://www.kloth.net

You can access the above site and select **Services** and then **dig.**

```
; <<>> DiG 9.3.2 <<>> www.riohondo.edu A

;; global options:  printcmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64295

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,

ADDITIONAL: 0

;; QUESTION SECTION:

;www.riohondo.edu.             IN        A

;; ANSWER SECTION:

www.riohondo.edu.6884         IN        A     156.3.125.16

;; Query time: 0 msec

;; SERVER: 127.0.0.1#53(127.0.0.1)

;; WHEN: Sat Dec 18 23:00:33 2010

;; MSG SIZE  rcvd: 50
```

The first line tells us about dig version (9.2.3) and the query. The second line shows the global options (*printcmd*). The "**Got answer**" section shows some technical details about the answer received from the DNS server. In the line after the **QUESTION SECTION**, we see the query again followed by IN (Internet) and A (IP address query). The **ANSWER SECTION** starts with TTL number (6884 seconds in this example) and finishes with the IP address (156.3.125.16) of the www.riohondo.edu website. The final output section contains statistics about the query.

If we make the same query with NS (name server) option, the result looks as shown in the box below:

```
...
;www.riohondo.edu.          IN    NS
;; AUTHORITY SECTION:
riohondo.edu.         6535  IN    SOA     nccunix.lacoe.edu.
art.mssmtp.lacoe.edu. 2010120801 10800 1800 3600000 7200
...
```

The **AUTHRITY SECTION** in this case gives the host names of two authoritative DNS for riohondo.edu (nccunix.lacoe.edu. and art.mssmtp.lacoe.edu.). We can

now repeat the query but instead of the local server, we can ask the root server nccunix.lacoe.edu. to perform the quarry. The result is shown in the next box.

```
...
;; QUESTION SECTION:
;www.riohondo.edu.          IN      A
;; ANSWER SECTION:
www.riohondo.edu.    7200    IN      A       156.3.125.16
;; AUTHORITY SECTION:
riohondo.edu.        7200    IN      NS      ns3.csu.net.
riohondo.edu.        7200    IN      NS      nccunix.lacoe.edu.
riohondo.edu.        7200    IN      NS      netmgr.lacoe.edu.
riohondo.edu.        7200    IN      NS      pws.lacoe.edu.
;; ADDITIONAL SECTION:
pws.lacoe.edu.       14400   IN      A       156.3.1.1
netmgr.lacoe.edu.    14400   IN      A       156.3.1.3
nccunix.lacoe.edu.   14400   IN      A       156.3.1.8
...
```

This time in the **AUTHORITY SECTION** we get the name of four DNS servers who can provide an authoritative answer to our query. The **ADDITION SECTION** provides the IP addresses of the DNS servers listed in the **AUTHORITY SECTION**.

Some other query types includes MS (mail exchange), SOA (start of query), and ANY (any type). One interesting feature of dig query is the trace option. To do the query with trace option, select the trace box. The result is shown in the next box.

```
; <<>> DiG 9.3.2 <<>> @localhost www.riohondo.edu A +trace
; (2 servers found)
;; global options:  printcmd
.                  192837      IN    NS      j.root-servers.net.
...
;; Received 228 bytes from 127.0.0.1#53(127.0.0.1) in 1 ms
edu.               172800      IN    NS      g.edu-servers.net.
...
;; Receive190 bytes from 156.3.1.8#53(nccunix.lacoe.edu) in
177 ms
```

This time *dig* shows you the root servers ".", followed by the servers responsible for "edu." domains, and finally followed by the name servers responsible for "riohondo.edu." domains.

Questions

Do the following query and answer the following question in your lab report.

1. Obtain the IP address of your campus.
2. Find the name and IP addresses of DNS servers who can provide an authoritative answer to the above query.
3. Run the Question 1 with trace option and interpret the result.
4. Run a DNS query to obtain IP of your mail server (MS option).

Part II: Using ipconfig

Select **Run** from the **Start** Menu of your computer, type **cmd** and press OK. In the command screen window type **ipconfig/all** and press enter. The current TCP/IP settings of your network are displayed.

Questions

Using the result of running *ipconfig*, answer the following question in your lab report.

1. What is the host name?
2. What is the connection-specific DNS suffix?
3. What is the physical (data-link) address?
4. What is the IP address?
5. What is the IP address of the default gateway? This address is the IP address of the host on the local subnet that provides the physical connection to remote networks.

Part III: Using Wireshark to Capture DNS Packets

To make sure that the packets we are going to captured are exchanged between the DNS servers and the host and are not saved in the host cache memory, start this section by emptying the DNS records from the host memory and the browser cache memory as shown below:

First, in the command screen window type **ipconfig/flushdns** and press enter to clear DNS record from the cache memory of your computer. Next, clear your browser's cache memory.

Open the Wireshark and start capturing. In your browser type the web address of your campus and press enter. Wireshark starts to capture packets. Type **dns** (lowercase) in the filter field and press **Apply** so that only DNS messages are displayed. Stop capturing and save the captured file.

The packet list pane of the Wireshark displays several DNS packets. In the packet details pane, make sure that the **Internet Protocol** box and all the boxes above it are collapsed (have plus sign). Expand Domain Name System and all the subsequent boxes below It.

Questions

Using the captured information, answer the following question in your lab report.

1. Do the DNS messages use the service of UDP or TCP?
2. What are the source and destination port numbers for the query DNS message?
3. What are the source and destination port numbers for the response DNS message?
4. To what IP address and what network the query message is sent?
5. What is the query message ID number? What is the response message ID number? What is the purpose of this field?
6. How many bits are in the flag field of a DNS message?
7. Which bit in the flag field determines whether the message is a query or a response?

8. Which bits are only used in the response message? What is the function of these bits in the response message?
9. What are the number of questions records, answer records, authority records, and addition records in the query message?
10. Interpret the information in the questions-and-answer sections of the packets.

26.6.12 Documents to Turn in

Turn in the following documents:

1. A copy of the Lab26-6 report sheet that contains answered questions.
2. A printout of the supporting captured information.