Name: ____

Experiment 9.2

Build A DNS Client Program

Purpose

To learn how to build a client for the Domain Name System.

Background Reading And Preparation

For an overview of the Domain Name System (DNS), read Chapter 31 in *Computer Networks And Internets*. The exact specifications of the protocol can be found in RFCs 1034 and 1035, which have been updated by RFC 2181. Although the specifications are both long and tedious, it will be necessary to consult them for details. The file *domain.h* that defines the format of DNS messages using C declarations can be found in directory:

ftp.cs.purdue.edu/pub/comer/labbook/lab9.2

A packet capture tool will also be useful in debugging. Later experiments ask you to construct such a tool. If you have not built a tool yourself, look at Experiment 3.5, which specifies downloading a shareware tool.

Overview

Build a DNS client that looks up a domain name and displays the corresponding IP address.

Procedure And Details (checkmark as each is completed)

______ 1. Build a client program that uses UDP to send a query message containing one question to a DNS server on port 53. The message consists of the ten items listed below. All integer values in the message must be represented in network byte order (use function *htons* or an equivalent).

Field	Type	Value
ID	16-bit integer	any value
Parameter	16-bit integer	0x100
Question count	16-bit integer	1
Answer count	16-bit integer	0
Authority count	16-bit integer	0
Add. info count	16-bit integer	0
Domain name	L bytes	Sequence of labels in name
Lookup type	16-bit integer	1
Lookup class	16-bit integer	1

A domain name is represented in a message as a sequence of labels, where each label consists of an 8-bit count field followed by a sequence of ASCII characters. The sequence terminates with a zero count. For example, the name *example.com* has two labels of length seven and three, so the name is encoded in thirteen octets:

```
07 ’e’ ’x’ ’a’ ’m’ ’p’ ’l’ ’e’ 03 ’c’ ’o’ ’m’ 00
```

______ 2. To test the client, look up the name *www.netbook.cs.purdue.edu* and dump the response in hexadecimal.

______ 3. Augment the program to print the IP address from the response in dotted decimal.

______ 4. Test the client by looking up a set of well-known domain names and verifying that the server returns the correct IP address for each.

Optional Extensions (checkmark options as they are completed)

______ 5. Extend the DNS client to handle DNS mail exchanger lookups (type *MX* records).

______ 6. Read about using DNS pointer queries (i.e., looking up an IP address to obtain a domain name), and extend the DNS client to provide pointer query lookup.