

Daniel Steele
Lab #8

Review the code below. What will happen if a response is not received to the initial message this client sends? What could we add to the client so that such a condition is not possible?

```
int main(int argc, char **argv) {
    int sockfd, portno, n;
    int serverlen;
    struct sockaddr_in serveraddr;
    struct hostent *server;
    char *hostname;
    char buf[BUFSIZE];

    /* check command line arguments */
    if (argc != 3) {
        fprintf(stderr, "usage: %s <hostname> <port>\n", argv[0]);
        exit(0);
    }
    hostname = argv[1];
    portno = atoi(argv[2]);

    /* socket: create the socket */
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");

    /* gethostbyname: get the server's DNS entry */
    server = gethostbyname(hostname);
    if (server == NULL) {
        fprintf(stderr, "ERROR, no such host as %s\n", hostname);
        exit(0);
    }

    /* build the server's Internet address */
    bzero((char *) &serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
        (char *)&serveraddr.sin_addr.s_addr, server->h_length);
    serveraddr.sin_port = htons(portno);

    /* get a message from the user */
    bzero(buf, BUFSIZE);
    printf("Please enter msg: ");
    fgets(buf, BUFSIZE, stdin);

    /* send the message to the server */
    serverlen = sizeof(serveraddr);
    n = sendto(sockfd, buf, strlen(buf), 0, &serveraddr, serverlen);
    if (n < 0)
        error("ERROR in sendto");

    /* print the server's reply */
    n = recvfrom(sockfd, buf, strlen(buf), 0, &serveraddr,
&serverlen);
    if (n < 0)
```

```
        error("ERROR in recvfrom");  
        printf("Echo from server: %s", buf);  
        return 0;  
    }
```

My understanding of this code leads me to conclude that the code will block indefinitely until a message has been received. This could be rectified by adding a timeout on the `recv` function such that it will return after a certain amount of time if data has not been received allowing the program to either retry the communication or exit the program with an appropriate error. I believe though that the best action would be a combination of these options where the program will retry communication on failure up to so many times and then will return an error. This though may not apply to every situation and in this case it would probably be best just to exit with an error.