1) (25 points - total) Consider the following. A concurrent echo server is started on node A. The echo client on node B is started and “hello world” is entered by the user. A second later, “hello world” is echoed from the server to the client. Suddenly the server host crashes and all communication from that process is immediately ceased, however the client is not aware of the crash. The user at the client decides to enter “hello again”. Draw and label the TCP state diagram and show the transitions for the TCP client and server. Show diagrams that reflect the state of the client and server before the server crash, after the crash but before “hello again” is typed at the client, and after the crash and “hello again” is typed.

   a) Draw separate state transition diagram for the both the client and the server (for each case).
   b) Also on the diagrams, write the names of all of the sockets Application Program Interface function calls that were executed by both the client and the server in this sequence of events.
   c) For both the client and the server, list all of the sockets Application Program Interface function calls that cause the Kernel to generate network traffic.
2) (25 points – total) You are the ICANN and you are responsible for the brand new classless IP address block 152.93.0.0/18. You want to allocate your IP addresses starting at the lowest numbered one and continue until you do not have to allocate any more. At present, three new Internet Service Providers (ISP1, ISP2, and ISP3) have requested the assignment of IP addresses. ISP1 has 32 customers; each of its customers needs IP addresses for 60 machines. ISP2 has 64 customers and each of their customers needs IP addresses for 31 machines. ISP3 has 8 customers, each customer has 120 machines. These three ISPs do not anticipate any future new customers, so you do not need to guess at future need for them. You do anticipate more new ISPs to approach you in the future asking for more IP addresses, so you are saving the larger numbered unused addresses for newer ISPs.

a) (10 points) List 5 of the 32 company addresses blocks assigned to ISP1.
b) (5 points) What is the block of addresses that ISP1 would advertise to the Internet?
c) (5 points) List 5 of the 64 company addresses blocks assigned to ISP2.
d) (5 points) List all of the company addresses blocks assigned to ISP3.
3) (25 points) Below is the familiar TCP daytime server - Figure 1.9. Modify the code to make this server a proxy that doesn’t locally have the requested time, rather it must query the true day time server (10.1.1.1) when it receives a request. You do not have to do any variable declarations.

Use line numbers and write your modified code at the bottom of the page. For example, if you would like to add two lines between lines 1 and 2 you should do the following:

1.1  printf("Here is the first line I want to add");
1.2  printf("Here is the second line I want to add");

```c
#include "unp.h"
#include "time.h"

int
main(int argc, char **argv)
{
    int listenfd, connfd;
    struct sockaddr_in servaddr;
    char *buff[MAXLINE];
    time_t ticks;

    listenfd = Socket(AF_INET, SOCK_STREAM, 0);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(13); /* daytime server */
    Bind(listenfd, (SA *) &servaddr, sizeof(servaddr));
    Listen(listenfd, LISTENQ);

    for (; ; ) {
    connfd = Accept(listenfd, (SA *) NULL, NULL);
    ticks = time(NULL);
    snprintf(buff, sizeof(buff), "%4.24s\r\n", ctime(&ticks));
    Write(connfd, buff, strlen(buff));
    Close(connfd);
    }
```
4) (25 points - total) Below are two Ethereal/Wireshark snapshots of TCP traffic between a client and server. The server (machine) has a firewall that makes it invisible to attempted connections on closed ports. The first snapshot shows the beginning of the traffic capture and the second snapshot shows the end on the traffic capture. The client machine is 192.168.1.100 the server is 131.96.49.104. If there are multiple clients, their traffic would be shown sequentially (that is, there would be no intermixing of connections). Justify your answers.

a) (5 points) How many different servers (applications) did the client attempt to connect to?

b) (5 points) How many servers (applications) did the client successfully connect to?

c) (5 points) Did the client or the server enter the 2MSL wait period after the connection was terminated?

d) (5 Points) What is the round-trip-time between the client and the server?

e) (5 points) What sockets functions caused packets #296 and #1782 to be transmitted?

Snapshot 1

<table>
<thead>
<tr>
<th>Seq</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1099</td>
<td>122.310592</td>
<td>192.168.1.100</td>
<td>131.96.49.104</td>
<td>SSHv2</td>
<td>Encrypted request packet len=48</td>
</tr>
<tr>
<td>1051</td>
<td>122.350821</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=48</td>
</tr>
<tr>
<td>1085</td>
<td>122.621410</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=48</td>
</tr>
<tr>
<td>1059</td>
<td>122.310592</td>
<td>192.168.1.100</td>
<td>131.96.49.104</td>
<td>SSHv2</td>
<td>Encrypted request packet len=48</td>
</tr>
<tr>
<td>1051</td>
<td>122.350821</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=112</td>
</tr>
</tbody>
</table>

Snapshot 2

<table>
<thead>
<tr>
<th>Seq</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1099</td>
<td>122.310592</td>
<td>192.168.1.100</td>
<td>131.96.49.104</td>
<td>SSHv2</td>
<td>Encrypted request packet len=48</td>
</tr>
<tr>
<td>1051</td>
<td>122.350821</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=48</td>
</tr>
<tr>
<td>1085</td>
<td>122.621410</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=48</td>
</tr>
<tr>
<td>1059</td>
<td>122.310592</td>
<td>192.168.1.100</td>
<td>131.96.49.104</td>
<td>SSHv2</td>
<td>Encrypted request packet len=48</td>
</tr>
<tr>
<td>1051</td>
<td>122.350821</td>
<td>131.96.49.104</td>
<td>192.168.1.100</td>
<td>SSHv2</td>
<td>Encrypted response packet len=112</td>
</tr>
</tbody>
</table>