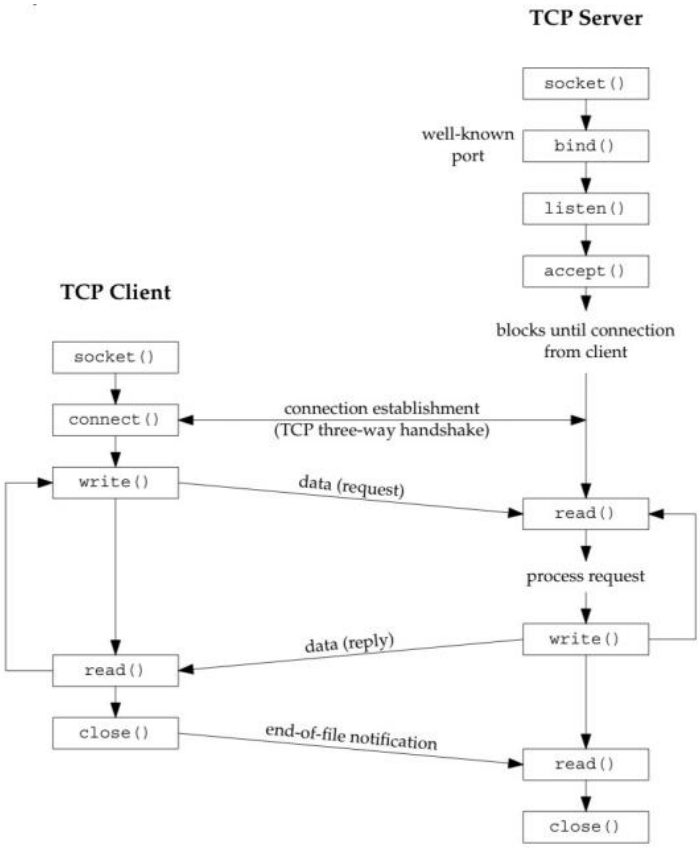
End Semester Examination, August-2018
UNIX Network Programming(CSE 4042)

1. a. Draw the timeline scenario that takes place for a TCP client/ server communication.



This figure shows a timeline of the typical scenario that takes place between a TCP client and server.

b. Write the complete 'C' code for TCP server assuming all header files are predefined. Also write the code fragment in server side to display the IP address and port number of the client to be connected

```
#include <stdio.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>

int main(int argc, char *argv[]) {
   int net_socket = socket(AF_INET, SOCK_STREAM, 0);

   struct sockaddr_in sv, cl;
   sv.sin_family = AF_INET;
   sv.sin_port = htons(0);
   sv.sin_addr.s_addr = htonl(INADDR_ANY);
```

```
    int len = sizeof(struct sockaddr_in);
    bind(net_socket, (struct sockaddr *)&sv, len);

    listen(net_socket, 10);

    getsockname(net_socket, (struct sockaddr *)&sv, &len);
    printf("Server IP: %s\nServer port: %d\n", inet_ntoa(sv.sin_addr), ntohs(sv.sin_port));

    int fd, n;
    char data[200];

    while (1) {
        len = sizeof(struct sockaddr_in);
        fd = accept(net_socket, (struct sockaddr *)&cl, &len);
        len = sizeof(struct sockaddr_in);
        getpeername(net_socket, (struct sockaddr *)&sv, &len);
        printf("Client IP: %s\nClient port: %d\n", inet_ntoa(cl.sin_addr), ntohs(cl.sin_port));

        int n = recv(fd, data, 199, 0);
        data[n] = '\0';
        printf("Client sent: %s\n", data);

        close(fd);
    }

    close(net_socket);
    return 0;
}
```

c. Create an appropriate TCP client 'C' code to send a message to the TCP server that you have coded in question 1(b).

```
#include<stdio.h>
#include<sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>
#include<stdlib.h>
#include<string.h>

int main(int argc, char *argv[]) {

    int net_socket;
    net_socket = socket(AF_INET, SOCK_STREAM, 0);

    struct sockaddr_in server_address;
    server_address.sin_family = AF_INET;
```

```
    server_address.sin_port = htons(atoi(argv[2]));
    server_address.sin_addr.s_addr = inet_addr(argv[1]);

    connect(net_socket, ( struct sockaddr * )&server_address, sizeof(server_address));

    char data[200];
    printf("Enter a sting: ");
    scanf("%s", data);
    write(net_socket, data, strlen(data));

    close(net_socket);
    return 0;
}
```

2. **a. A system has a n-layer protocol hierarchy. Applications generate messages of length M bytes. At each of the layers, an h-byte header is added. What fraction of the network bandwidth is filled with headers.**

   Length of message = M (bytes)

   Total headers = n (layers) x h (bytes)

   Fraction of the network bandwidth is filled with headers = nh / (M + nh)

   b. Name the different addressing mechanism related to the specific layer in the TCP/IP architecture.

   There are basically four types of addresses: physical, logical, port and specific address.

   The application layer which contains processes use specific address.

   The Transport layer which contains the TCP or UDP protocol uses port address.

   Network layer which contains the IP and other protocols uses logical address.

   Physical layer and data link layer uses physical address.

   c. Padhaye nhi h :)

3. **a. Imagine the length of a 10Base-5 thick co-axial cable is 2500 meters. If the speed of propagation in that cable is 60% of the speed of light, how long does it take for a bit to travel from the beginning to the end of the cable. (Speed of light = 3 × 10^8 meters/sec)**

   Propagation speed = 0.6 * 3 x 10 ^ 8 = 1.8 x 10 ^ 8 m/s
   Length = 2500 m
   Time = Length / Speed = 2500 / (1.8 x 10 ^ 8) = 13.88 micro sec

b. Given a channel with an intended capacity of 20 Mbps. The bandwidth of the channel is 3MHz. Determine the signal-to-noise ratio required in order to achieve this capacity.

Capacity = Bandwidth x $\log_2$ (S / N), S / N = the signal-to-noise ratio

20 x 10^6 = 3 x 10^6 x $\log_2$ (S / N)

20 / 3 = $\log_2$ (S / N)

2 ^ (20/3) = 2 ^ (6.67) = S / N

c. A digital signalling system is required to operate at 9600 bps.

C = 2B x log2 M

(i) If a signal element encodes a 4-bit word, find the minimum required bandwidth of the channel. Repeat part

log2 M = 4

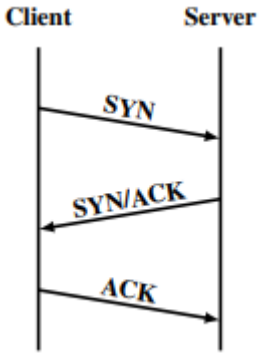
B = C / (2 x log2 M) = 9600 / (2 x 4) = 1200 Hz

(ii) for 8-bit word

log2 M = 8

B = C / (2 x log2 M) = 9600 / (2 x 8) = 1200 Hz = 600 Hz

4. a. The operation of TCP with regard to connection establishment can be specified with a TCP three-way handshake. Draw the TCP thee-way handshake scenario and write the number of packets required to be exchanged during connection setup.



Number of packets required to be exchanged during connection setup = 3
(SYN, SYN + ACK, ACK)

b. UNP lab machines are configured with loopback and Ethernet interfaces. The client-server program is running in the given cases:
(i) both client and server are running in a single machine,
(ii) client and server are running in different machines.

Write the suitable interfaces to be used for both the cases and fill the IPV4 socket address structure in client side for case-(ii)

(i). loopback interface

```
struct sockaddr_in sv;
sv.sin_family = AF_INET;
sv.sin_port = htons(atoi(argv[1]));
sv.sin_addr.s_addr = inet_addr("127.0.0.1");
```

(ii). ethernet interface

```
struct sockaddr_in sv;
sv.sin_family = AF_INET;
sv.sin_port = htons(atoi(argv[2]));
sv.sin_addr.s_addr = inet_addr(argv[1]);
```

c. Give an example of a socket address. The socket pair for a TCP connection is a four-tuple that defines the two endpoints of a connection. State the four-tuple to identify the TCP connection on a network.

Example of a socket address:

```
struct sockaddr_in sv;
sv.sin_family = AF_INET;
sv.sin_port = htons(0);
sv.sin_addr.s_addr = htonl(INADDR_ANY);
```

TCP socket is identified by a four-tuple: (**source IP address, source port number, destination IP address, destination port**)

5. **a. If you run the ifconfig command in UNP lab, it gives the IP and subnet mask as 172.17.152.101/255.255.240.0. Assuming the IP address is in classful, Find the number of subnets and the number of hosts per subnet.**

172.17.152.101 -> Class B -> 16 bits reserved for network
255.255.240.0 in binary -> 11111111. 11111111. 11110000. 00000000
-> 20 bits reserved for network and 12 bits for host

Number of subnets = 2 ^ (20 - 16) = 2 ^ 4 = 16
Number of hosts = (2 ^ 12) – 2 = 4096 – 2 = 4094 (excluding loopback and broadcast)

b. Suppose that instead of using 16 bits for the network part of a class B address originally, 20 bits had been used. Calculate the number of class B networks and number of hosts per network.

With a 2-bit prefix, there would have been 20-2 = 18 bits left over to indicate the network.
Consequently, the number of networks would have been 2^18 = 262144.
Number of hosts per network = 2^12 – 2 = 4096 – 2 = 4094 (excluding loopback and broadcast)

c. Convert the IP addresses C22F1582, and FEB46923 in hexadecimal representation to dotted decimal notation. Also find binary notation of the IP addresses 150.12.56.89, and 10.2.36.78.

C22F1582 -> 194.47.21.130
FEB46923 -> 254.180.105.35
150.12.56.89 -> 10010110.00001100.00111000.01011001
10.2.36.78 -> 00001010.00000010.00100100.01001110

6. **a. Consider a sender has n data packets, each of size L bits are transmitted to a receiver at a distance D meter away. The sender can send one packet at a time and wait for the acknowledgement called ACK packet from the receiver (the size of ACK is very very less than the data packet), then it can send the next packet. The process will continue till to the last frame. Find the efficiency of this mechanism assuming the working bandwidth of the network is B bps, and propagation speed is V m/s.**

Tt = L / B, Tp = D / V

We know,
Total time = Tt(data) + Tp(data) + Tq + Tpro + Tt(ack) + Tp(ack)
where Tp(ack) = Tp(data), Tt(ack) << Tt(data), Tq = 0 and Tpro = 0

Total time = Tt + 2 x Tp
Efficiency = useful time / total cycle time = Tt / (Tt + 2 x Tp)
Efficiency = 1 / (1 + 2a), a = Tp / Tt = DB / VL

b. Calculate the length or size of the packet, if the efficiency is at least 50%, bandwidth (B) = 6 Mbps, and propagation time (Tp) = 1 milisecond.

Efficiency = 1 / (1 + 2a) >= 0.5, a = Tp / Tt
a <= 0.5
Tt = Tp / a = 10^{-3} / 0.5 = 2 ms
Size of packet = Tt x B = 2 x 10^{-3} x 6 x 10^{6} = 12000 bits

c. Consider a closed-loop network (e.g. a token ring network) with bandwidth 100 Mbps and propagation speed of 2 × 108 m/s. Calculate the circumference of the loop be to exactly contain one 250 byte packet?

T = L / B = 250 x 8 / (100 x 10^{6}) = 20 micro sec
Circumference = Propagation speed x T = 2 x 10^{8} x 20 x 10^{-6} = 4000 m

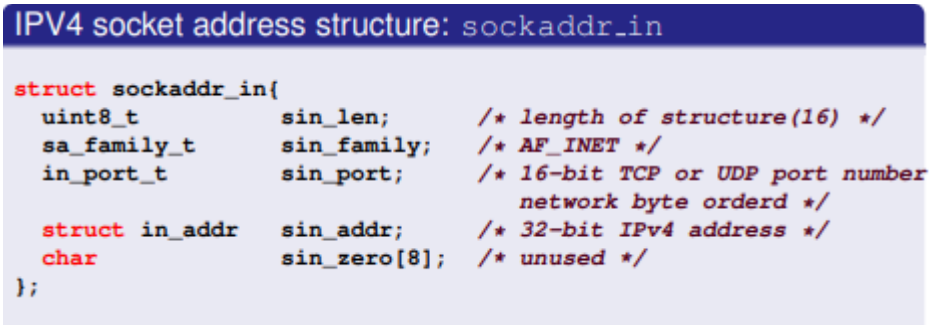
7. **a. ITER network administrator has configured classful IPV4 address and subnet mask for computers C1 and C2. C1 has IP address 172.17.35.227, and subnet mask 255.255.240.0. C2 has IP address 172.17.144.10, and subnet mask 255.255.255.240. Compute the network address of C1, and C2. Decide whether they belong to same network or different network.**

Network address = IP address bitwise AND subnet mask
For C1:
IP Address = 172.17.35.227 = 10101100. 00010001. 00100011. 11100011
Mask = 255.255.240.0 = 11111111. 11111111. 11110000. 00000000
Bitwise AND
Network Address = 10101100. 00010001. 00100000. 00000000 = 172.17.32.0

For C2:
IP Address = 172.17.144.10 = 10101100. 00010001. 10010000. 00001010
Mask = 255.255.255.240 = 11111111. 11111111. 11111111. 11110000
Bitwise AND
Network Address = 10101100. 00010001. 10010000. 00000000 = 172.17.144.0

C1 and C2 belong to different networks.

9. **a. An IPv4 socket address structure called "Internet socket address structure" is named sockaddr in. States the fileds of sockaddr in.**



b. Write the 'C' statements to assign the IP 205.32.56.45 to the variable struct sockaddr in server using two different functions call inet addr(), and inet aton().

struct sockaddr_in server;
server.sin_addr.s_addr = inet_addr("205.32.56.45");

struct sockaddr_in server;
inet_aton("205.32.56.45", &server.sin_addr);

c. Assume that a client is successfully connected to a server running on IP : 172.17.144.145 and port : 45689. Write the appropriate 'C' statements to display the IP address and port number of the connected client after the accept function in the server code.

// server side code
// after accept() function

struct sockaddr_in client;
int len = sizeof(struct sockaddr_in);
getpeername(net_socket, (struct sockaddr *)&client, &len);
printf("Client IP: %s\nClient port: %d\n", inet_ntoa(client.sin_addr), ntohs(client.sin_port));

10. a. We have used three address conversion functions in TCP socket programming to convert IP addresses between ASCII string and binary values. Write the function prototype of each function and their return type.

in_addr_t inet_addr(const char *strptr);
Returns: 32-bit binary network byte orderedIPv4 address,
INADDR_NONE if error

int inet_aton(char *strptr, struct in_addr *addptr);
Returns: 1 if string was valid, 0 on error

int inet_pton(int family, char *strptr, void *addptr);
Returns: 1 if OK,
0 if input not a valid presentation format,
-1 on error

b. Sate the byte ordering used by a system. Write the byte ordering conversion functions used in socket.

The ordering used on a computer is called host-byte ordering.
A host system may be little-endian but when sending data into the network,
it must convert data into big-endian format. Likewise, a little-endian machine
must first convert network data into little-endian before processing it.

htons() -> host to network short
htonl() -> host to network long
ntohs() -> network to host short
ntohl() -> network to host long

c. If a server contains four accept() calls and all accept function executed successfully, then at last how many clients would be connected to the server.

4 clients would be connected to the server.