

# NETWORK PROGRAMMING

## Lab Assignment- 7 (UCS413)



**Submitted By:**

Name: Rishabh Jain

Roll Number: 102083054

# Q1) Multicasting:

## Server:

```
C server.c X C client.c
C server.c > main()
1  #include <sys/types.h>
2  #include <sys/socket.h>
3  #include <arpa/inet.h>
4  #include <netinet/in.h>
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main() {
9
10     int socketId = socket(AF_INET, SOCK_DGRAM, 0);
11     if (socketId < 0) {
12         printf("Error creating datagram socket\n");
13         exit(1);
14     }
15     char message[] = "This is a Multicast Message";
16
17     struct sockaddr_in servaddr;
18     bzero(&servaddr, sizeof(servaddr));
19     servaddr.sin_family = AF_INET;
20     servaddr.sin_addr.s_addr = inet_addr("225.1.1.1");
21     servaddr.sin_port = htons(5555);
22
23     char loopch = 0;
24     if (setsockopt(socketId, IPPROTO_IP, IP_MULTICAST_LOOP,
25         (char*)&loopch, sizeof(loopch)) < 0) {
26         printf("Error setting IP_MULTICAST_LOOP:");
27         close(socketId);
28         exit(1);
29     }
30
31     struct in_addr multicastInterface;
32     multicastInterface.s_addr = inet_addr("192.168.43.172");
33     if (setsockopt(socketId, IPPROTO_IP, IP_MULTICAST_IF,
34         (char*)&multicastInterface,
35         sizeof(multicastInterface)) < 0) {
36         printf("setting local interface");
37         exit(1);
38     }
39     if (sendto(socketId, message, sizeof(message), 0,
40         (struct sockaddr*)&servaddr,
41         sizeof(servaddr)) < 0) {
42         printf("Error sending datagram message");
43     }
44 }
```

# Client:

```
server.c  client.c  X
C client.c > main()
1  #include <sys/types.h>
2  #include <sys/socket.h>
3  #include <arpa/inet.h>
4  #include <netinet/in.h>
5  #include <stdio.h>
6  #include <stdlib.h>
7
8  int main() {
9      char buffer[1024];
10     int socketID = socket(AF_INET, SOCK_DGRAM, 0);
11     if (socketID < 0) {
12         printf("Error opening datagram socket");
13         exit(1);
14     }
15     int reuse = 1;
16     if (setsockopt(socketID, SOL_SOCKET, SO_REUSEADDR,
17         (char*)&reuse, sizeof(reuse)) < 0) {
18         printf("Error setting SO_REUSEADDR\n");
19         close(socketID);
20         exit(1);
21     }
22     struct sockaddr_in serveraddr;
23     bzero(&serveraddr, sizeof(serveraddr));
24     serveraddr.sin_family = AF_INET;
25     serveraddr.sin_port = htons(5555);
26     serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
27     if (bind(socketID, (struct sockaddr*)&serveraddr, sizeof(serveraddr))) {
28         printf("Error binding datagram socket\n");
29         close(socketID);
30         exit(1);
31     }
32     struct ip_mreq multicastGroup;
33     multicastGroup.imr_multiaddr.s_addr = inet_addr("225.1.1.1");
34     multicastGroup.imr_interface.s_addr = inet_addr("192.168.43.172");
35     if (setsockopt(socketID, IPPROTO_IP, IP_ADD_MEMBERSHIP,
36         (char*)&multicastGroup, sizeof(multicastGroup)) < 0) {
37         printf("Error adding multicast group");
38         close(socketID);
39         exit(1);
40     }
41     if (read(socketID, buffer, sizeof(buffer)) < 0) {
42         printf("Error reading datagram message");
43         close(socketID);
44         exit(1);
45     }
46     else
47         printf("Message from multicast datagram: %s\n", buffer);
48 }
```

## Q2) Broadcasting:

### Server:

```
server.c  x  client.c
C server.c > ...
1 // Rishabh Jain 102083054
2 // Assignment 7_2
3 #include<stdio.h>
4 #include<netinet/in.h>
5 #include<sys/types.h>
6 #include<sys/socket.h>
7 #include<netdb.h>
8 #include<string.h>
9 #include<stdlib.h>
10 #define PORT 8080
11
12 int main() {
13     int socketId;
14
15     struct sockaddr_in servaddr, cli;
16     socketId = socket(AF_INET, SOCK_DGRAM, 0);
17     int broadcastPermission = 1;
18     setsockopt(socketId, SOL_SOCKET, SO_BROADCAST,
19         (void*)&broadcastPermission, sizeof(broadcastPermission));
20     if (socketId == -1) {
21         printf("Socket creation failed...\n");
22         exit(0);
23     }
24     else
25         printf("Socket successfully created..\n");
26
27     char message[] = "This is a broadcast Message";
28     bzero(&servaddr, sizeof(servaddr));
29     servaddr.sin_family = AF_INET;
30     servaddr.sin_addr.s_addr = inet_addr("192.168.43.255");
31     servaddr.sin_port = htons(PORT);
32     sendto(socketId, message, sizeof(message), 0,
33         (struct sockaddr*)&servaddr, sizeof(servaddr));
34     printf("Broadcast Sent\n");
35     close(socketId);
36 }
37
```

# Client:

```
server.c  client.c  X
client.c > PORT
1  // Rishabh Jain 102083054
2  // Assignment 7_2
3  #include<sys/socket.h>
4  #include<netdb.h>
5  #include<string.h>
6  #include<stdlib.h>
7  #include<stdio.h>
8
9  #define PORT 8080
10
11 int main() {
12     char buffer[1024];
13     int socketID, len, n;
14     struct sockaddr_in servaddr;
15     socketID = socket(AF_INET, SOCK_DGRAM, 0);
16     if (socketID < 0) {
17         printf("socket creation failed...\n");
18         exit(0);
19     }
20     else
21         printf("Socket successfully created..\n");
22     bzero(&servaddr, sizeof(len));
23     servaddr.sin_family = AF_INET;
24     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
25     servaddr.sin_port = htons(PORT);
26
27     if ((bind(socketID, (struct sockaddr_in*)&servaddr, sizeof(servaddr))) != 0) {
28         printf("Error socket bind failed...\n");
29         exit(0);
30     }
31
32     len = sizeof(servaddr);
33
34     recvfrom(socketID, buffer, sizeof(buffer), 0, (struct sockaddr_in*)&servaddr, &len);
35     printf("Message Recieved : %s\n", buffer);
36
37     close(socketID);
38 }
39
```