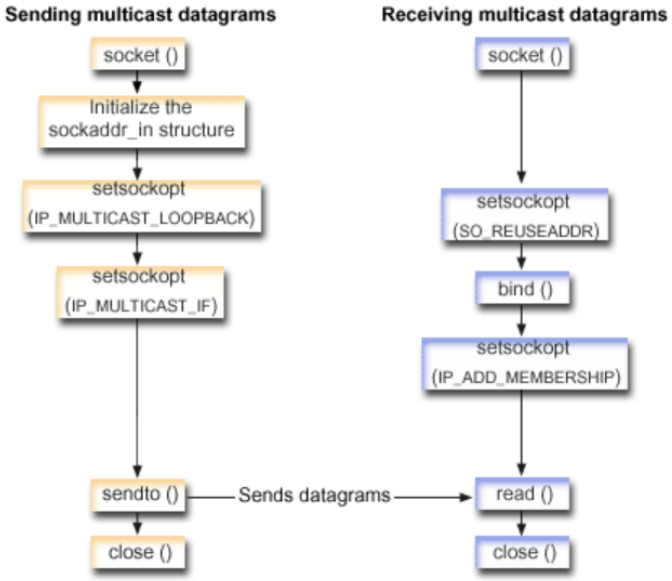
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Thapar Institute of Engineering and Technology, Patiala

Computer Science and Engineering
Network Programming Laboratory (UCS413)
Assignment: 7

1. **Design sending and receiving processes to implement multicasting using socket programming.**



Steps need to preform multicasting for connection-less client/server communication are:

Sequence of API calls for sending multicast datagrams:

a) Create an AF_INET, SOCK_DGRAM type socket.
b) Initialize a sockaddr_in structure.
c) Set the IP_MULTICAST_LOOP socket option according to whether the sending system should receive a copy of the multicast datagrams that are transmitted.
d) Set the IP_MULTICAST_IF socket option to define the local interface over which you want to send the multicast datagrams.
e) Send the datagram.

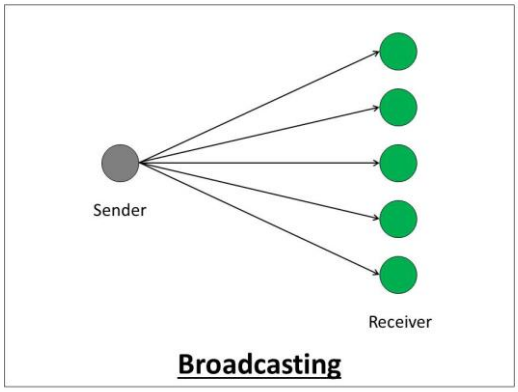
Sequence of API calls for receiving multicast datagrams:

a) Create an AF_INET, SOCK_DGRAM type socket.
b) Set the SO_REUSEADDR option to allow multiple applications to receive datagrams that are destined to the same local port number.

c) Use the bind() verb to specify the local port number. Specify the IP address as INADDR_ANY in order to receive datagrams that are addressed to a multicast group.
d) Use the IP_ADD_MEMBERSHIP socket option to join the multicast group that receives the datagrams. When joining a group, specify the class D group address along with the IP address of a local interface. The system must call the IP_ADD_MEMBERSHIP socket option for each local interface receiving the multicast datagrams.
e) Receive the datagram.

2. Design sending and receiving processes to implement broadcasting using connectionless socket programming.

Each network segment has a corresponding broadcast address. Take the class C network segment 192.168.1.x as an example, where the smallest address 192.168.1.0 represents the network segment; and the largest address 192.168.1.255 is the broadcast address in the network segment. When we want to send a data packet to this address, all hosts on the network segment will receive and process it.



Broadcast packets are sent and received through UDP sockets.

The broadcast packet sending process is as follows:

a. Create a UDP socket; socket(AF_INET, SOCK_DGRAM, 0)
b. Fill the broadcast information structure; struct sockaddr_in
c. Set socket options to allow broadcast packets to be sent; setsockopt(, ,SO_BROADCAST,)
d. Send data packet; sendto()

The broadcast packet receiving process is as follows:

a. Create a UDP socket; socket(AF_INET, SOCK_DGRAM, 0)
b. Fill the broadcast information structure; struct sockaddr_in
c. Bind address and port; bind()
d. Receive data packet; recvfrom()