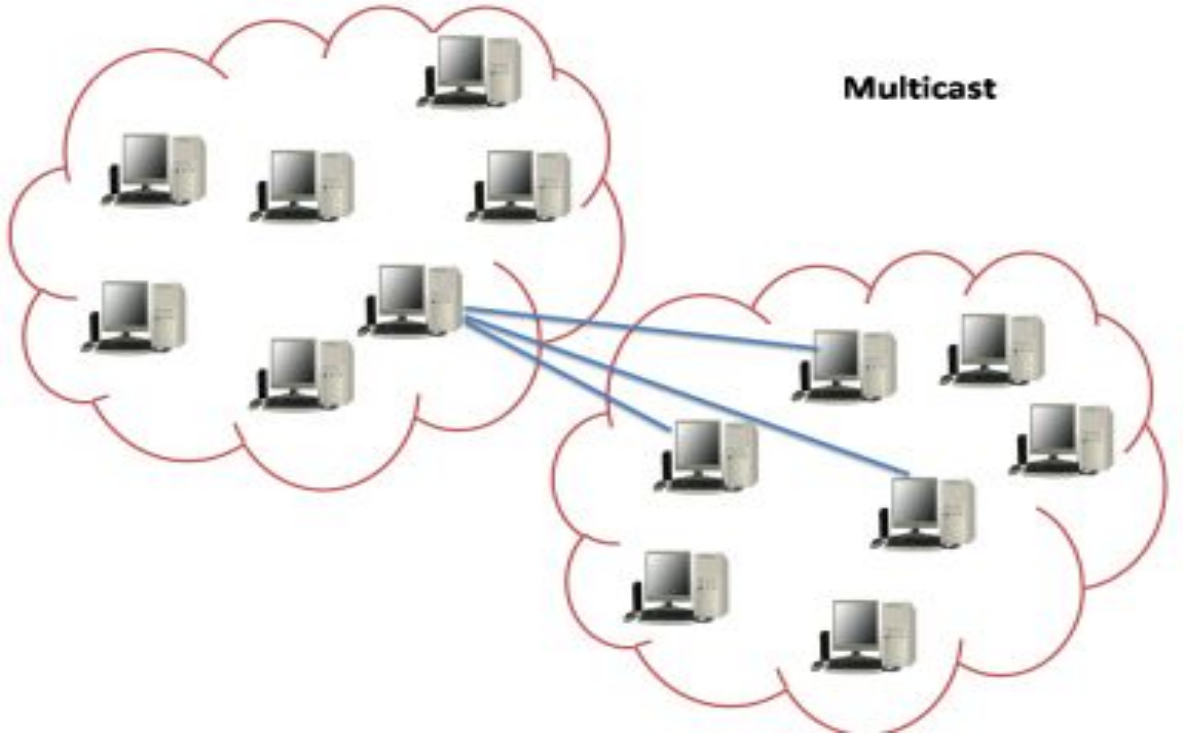
Socket Programming

Section 7

MULTICAST

- Multicast is communication between a single sender and multiple receivers on a network..



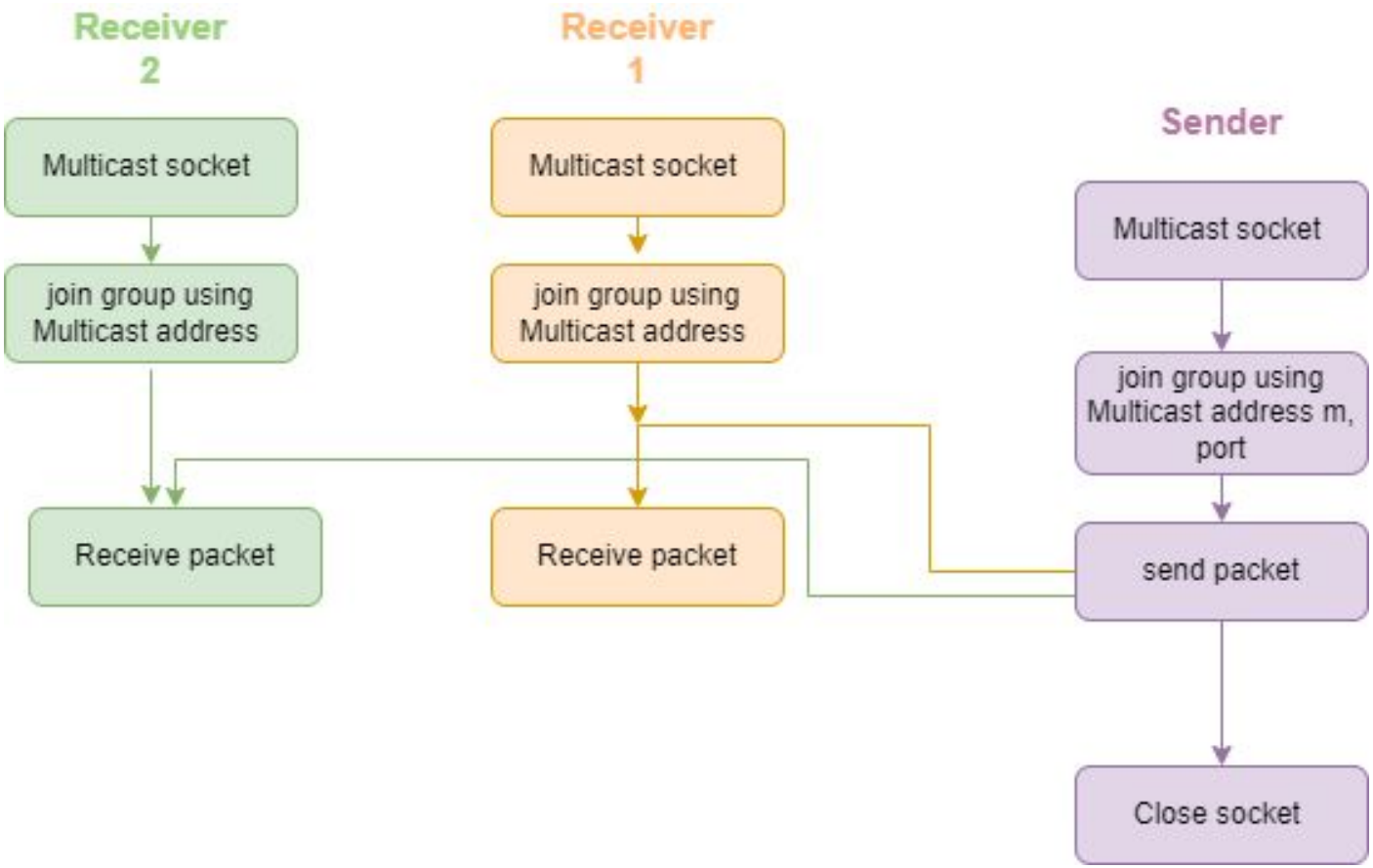
- The most common transport layer protocol to use multicast addressing is User Datagram Protocol (UDP).

- A MulticastSocket is a (UDP) DatagramSocket, with additional capabilities for joining "groups" of other multicast hosts on the internet.

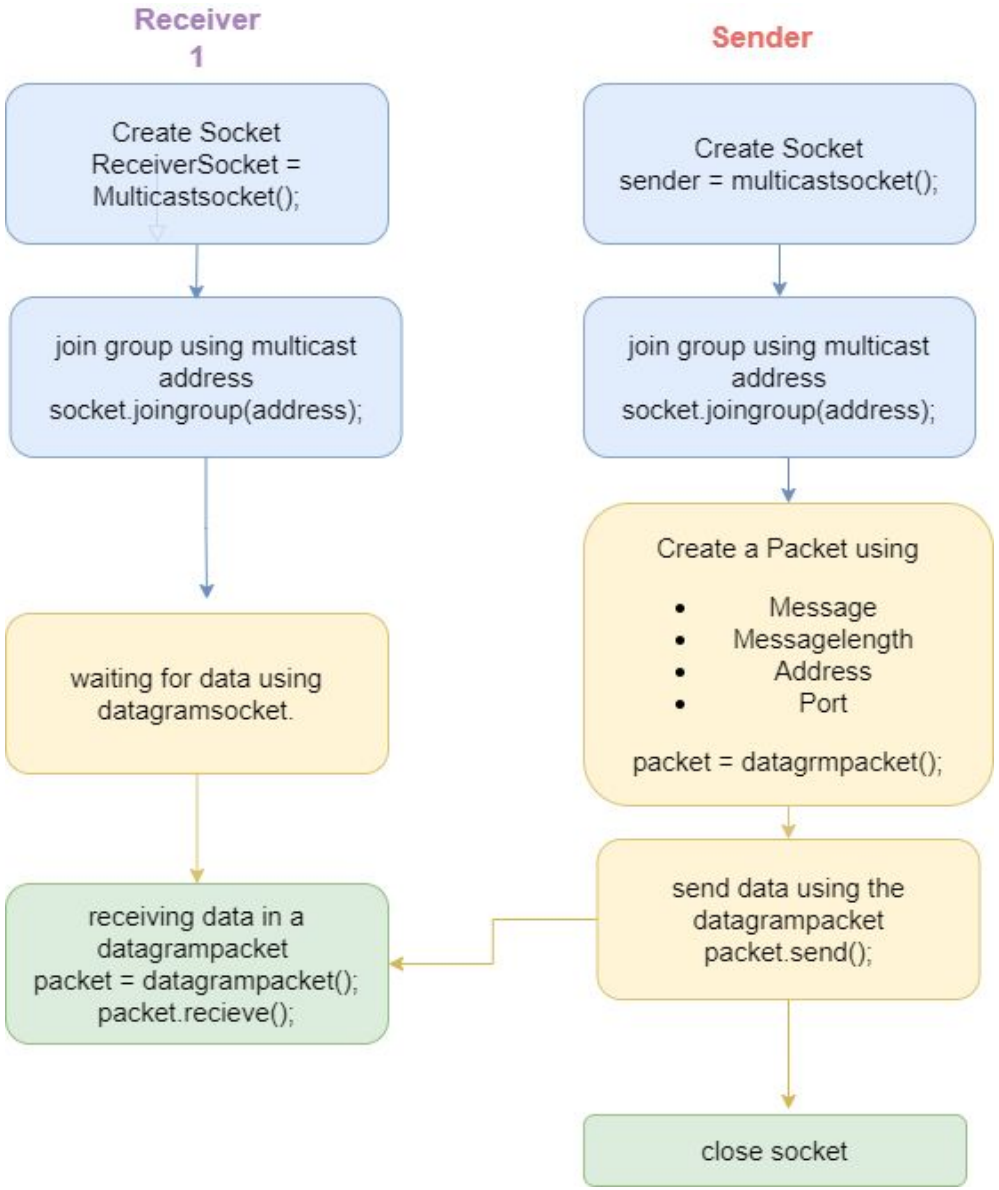
- The IP address that defines a multicast group is a Class D address (224.0. 0.0 to 239.255. 255.255).

- Any packet that's addressed to one of these IP addresses is automatically assumed to be a multicast, and is delivered according to the multicast recipient group.

Multicast sockets



Multicast socket programming in java



Multicast socket file transfer from sender

```
public class Sender {

    public static void main(String[] args) throws Exception{
        // TODO Auto-generated method stub

        MulticastSocket m = new MulticastSocket();
        InetAddress add = InetAddress.getByName("224.1.1.1");
        int port = 9999;
        m.joinGroup(add);
        FileInputStream fi = new FileInputStream("D://Sockets.txt");
        byte []b = new byte[2000];
        fi.read(b,0,b.length);
        DatagramPacket dp = new DatagramPacket(b, b.length,add,port);
        m.send(dp);
        m.close();

    }
```

Multicast socket file transfer to Receiver1

```
public class Reciever {

    public static void main(String[] args) throws Exception{
        // TODO Auto-generated method stub
        MulticastSocket m = new  MulticastSocket(9999);
        InetAddress add = InetAddress.getByName("224.1.1.1");
        m.joinGroup(add);
        byte [] bi = new byte[2000];
        FileOutputStream fo = new FileOutputStream("D://Receiver1.txt");
        DatagramPacket dp = new DatagramPacket(bi, bi.length);
        m.receive(dp);
        fo.write(bi);

    }
```

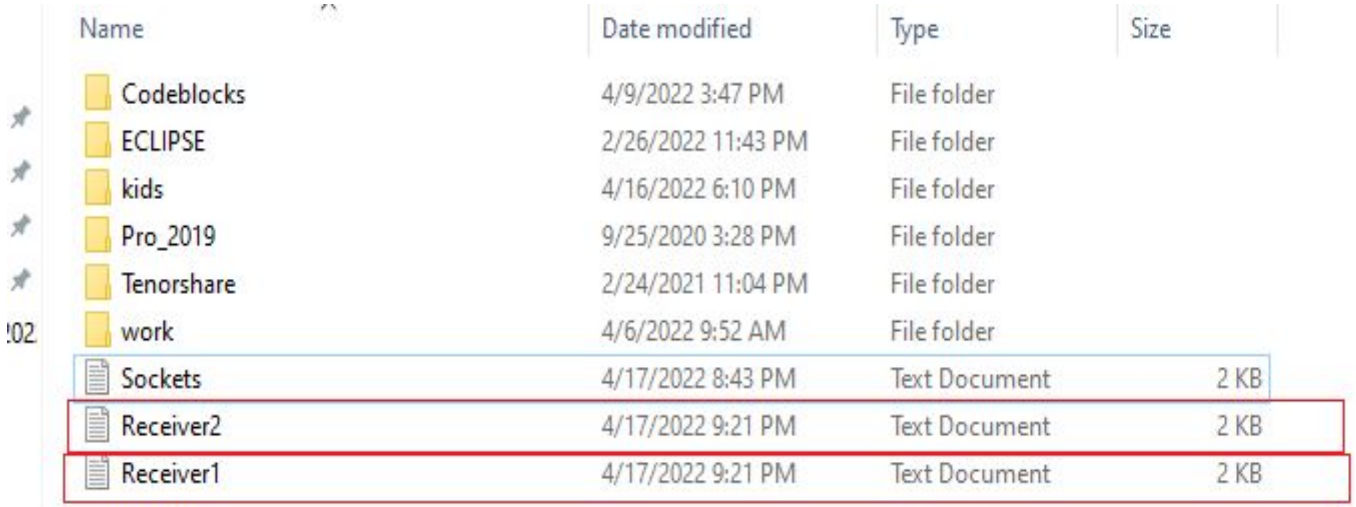
Multicast socket file transfer to Receiver2

```
public class reciever2 {

    public static void main(String[] args) throws Exception{
        // TODO Auto-generated method stub
        MulticastSocket m = new  MulticastSocket(9999);
        InetAddress add = InetAddress.getByName("224.1.1.1");
        m.joinGroup(add);
        byte [] bi = new byte[2000];
        FileOutputStream fo = new FileOutputStream("D://Receiver2.txt");
        DatagramPacket dp = new DatagramPacket(bi, bi.length);
        m.receive(dp);
        fo.write(bi);
    }

}
```

Received files



Name	Date modified	Type	Size
Codeblocks	4/9/2022 3:47 PM	File folder	
ECLIPSE	2/26/2022 11:43 PM	File folder	
kids	4/16/2022 6:10 PM	File folder	
Pro_2019	9/25/2020 3:28 PM	File folder	
Tenorshare	2/24/2021 11:04 PM	File folder	
work	4/6/2022 9:52 AM	File folder	
Sockets	4/17/2022 8:43 PM	Text Document	2 KB
Receiver2	4/17/2022 9:21 PM	Text Document	2 KB
Receiver1	4/17/2022 9:21 PM	Text Document	2 KB