St. Francis Institute of Technology
Department of Computer Engineering

Academic Year: 2021-2022 **Semester: VIII**
Subject: Distributed Computing **Class / Branch / Division: BE/CMPN/A**
Name: Rebecca Dias Roll Number:18

Experiment No: 10

Aim: Implement Group Communication

Theory:

1. **What is Group Communication?**
   Communication between two processes in a distributed system is required to exchange various data, such as code or a file, between the processes. When one source process tries to communicate with multiple processes at once, it is called Group Communication. A group is a collection of interconnected processes with abstraction. This abstraction is to hide the message passing so that the communication looks like a normal procedure call.

2. **Need for Group Communication.**
   Group communication also helps the processes from different hosts to work together and perform operations in a synchronized manner, therefore increases the overall performance of the system.

3. **Types of Group Communication**

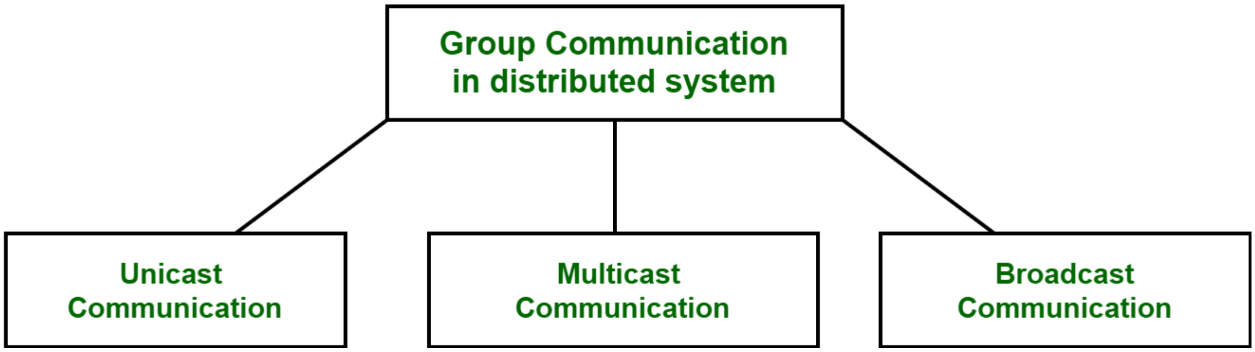


Figure No:10.1 Types of Group communication

i) Broadcast Communication :

When the host process tries to communicate with every process in a distributed system at same time. Broadcast communication comes in handy when a common stream of information is to be delivered to each and every process in most efficient manner possible. Since it does not require any processing whatsoever, communication is very fast in comparison to other modes of communication. However, it does not support a large number of processes and cannot treat a specific process individually.

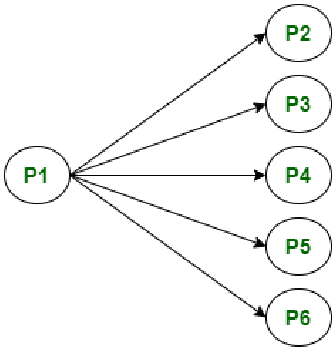


Figure No:10.2 A broadcast Communication: P1 process communicating with every process in the system

ii) Multicast Communication :

When the host process tries to communicate with a designated group of processes in a distributed system at the same time. This technique is mainly used to find a way to address the problem of a high workload on the host system and redundant information from processes in the system. Multitasking can significantly decrease time taken for message handling.

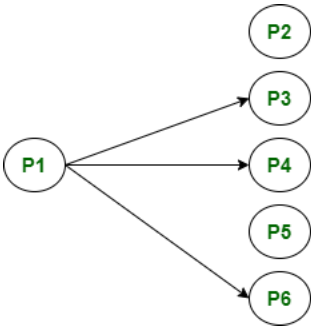


Figure No:10.3 A multicast Communication: P1 process communicating with only a group of the process in the system

iii) Unicast Communication :

When the host process tries to communicate with a single process in a distributed system at the same time. Although, same information may be passed to multiple processes. This works best for two processes communicating as only it has to treat a specific process only. However, it leads to overheads as it has to find exact process and then exchange information/data.

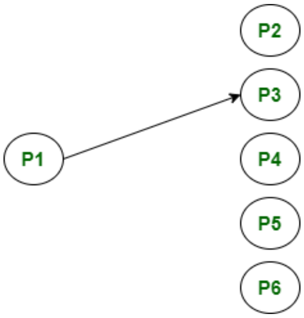


Figure No: 10.4 A broadcast Communication: P1 process communicating with only P3 process

4. **Application of group communication**
   a) Troop Messenger: Troop Messenger- a professional team chat software that combines the whole team in a single platform with user-friendly functionalities and features.
   b) HipChat: A well-developed team collaboration messaging app that makes your group communications simple. It allows you to share any files through any HipChat application by web, mobile or desktops.
   c) Campfire: A perfect group chat tool, which is designed exclusively for groups. It enables real-time communication with effective add-on features.
   d) Google+ Hangouts: Google Hangouts- A new G+ Business chat application that connects friends and colleagues in your workplace.

Implementation:

Server.py

```
'Chat Room Connection - Client-To-Client'
import threading
import socket
host = '127.0.0.1'
port = 59000
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((host, port))
server.listen()
clients = []
aliases = []


def broadcast(message):
    for client in clients:
        client.send(message)
```

```
# Function to handle clients'connections


def handle_client(client):
    while True:
        try:
            message = client.recv(1024)
            broadcast(message)
        except:
            index = clients.index(client)
            clients.remove(client)
            client.close()
            alias = aliases[index]
            broadcast(f'{alias} has left the chat room!'.encode('utf-8'))
            aliases.remove(alias)
            break
# Main function to receive the clients connection


def receive():
    while True:
        print('Server is running and listening ...')
        client, address = server.accept()
        print(f'connection is established with {str(address)}')
        client.send('alias?'.encode('utf-8'))
        alias = client.recv(1024)
        aliases.append(alias)
        clients.append(client)
        print(f'The alias of this client is {alias}'.encode('utf-8'))
        broadcast(f'{alias} has connected to the chat room'.encode('utf-8'))
        client.send('you are now connected!'.encode('utf-8'))
        thread = threading.Thread(target=handle_client, args=(client,))
        thread.start()


if __name__ == "__main__":
    receive()
```

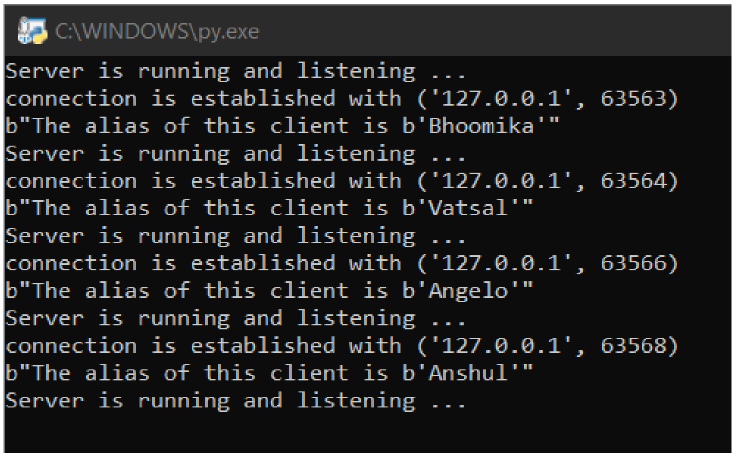
Client.py

```
import threading
import socket
alias = input('Choose an alias >>> ')
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
client.connect(('127.0.0.1', 59000))


def client_receive():
    while True:
        try:
            message = client.recv(1024).decode('utf-8')
            if message == "alias?":
                client.send(alias.encode('utf-8'))
            else:
                print(message)
        except:
            print('Error!')
            client.close()
            break


def client_send():
    while True:
        message = f'{alias}: {input("")}'
        client.send(message.encode('utf-8'))


receive_thread = threading.Thread(target=client_receive)
receive_thread.start()

send_thread = threading.Thread(target=client_send)
send_thread.start()
```

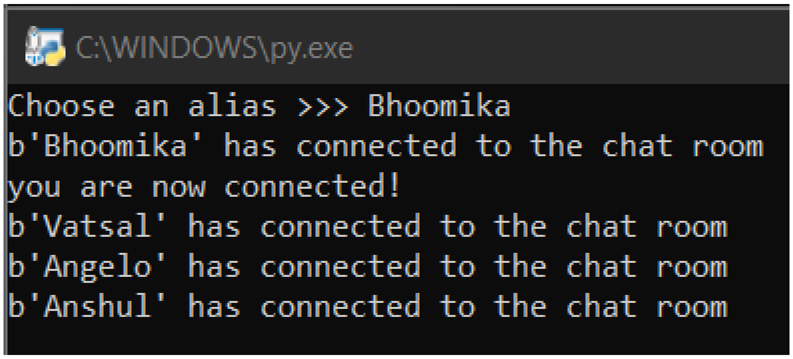
Output:

Server running :



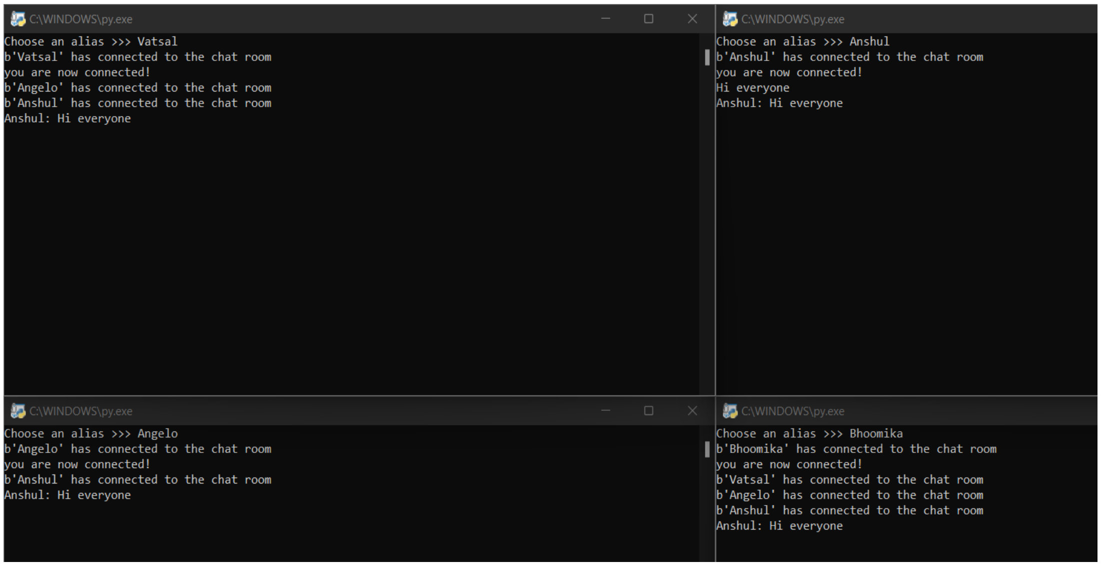
```
Server is running and listening ...
connection is established with ('127.0.0.1', 63563)
b"The alias of this client is b'Bhoomika'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63564)
b"The alias of this client is b'Vatsal'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63566)
b"The alias of this client is b'Angelo'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63568)
b"The alias of this client is b'Anshul'"
Server is running and listening ...
```

Client joining server



```
Choose an alias >>> Bhoomika
b'Bhoomika' has connected to the chat room
you are now connected!
b'Vatsal' has connected to the chat room
b'Angelo' has connected to the chat room
b'Anshul' has connected to the chat room
```

Client sending message in group



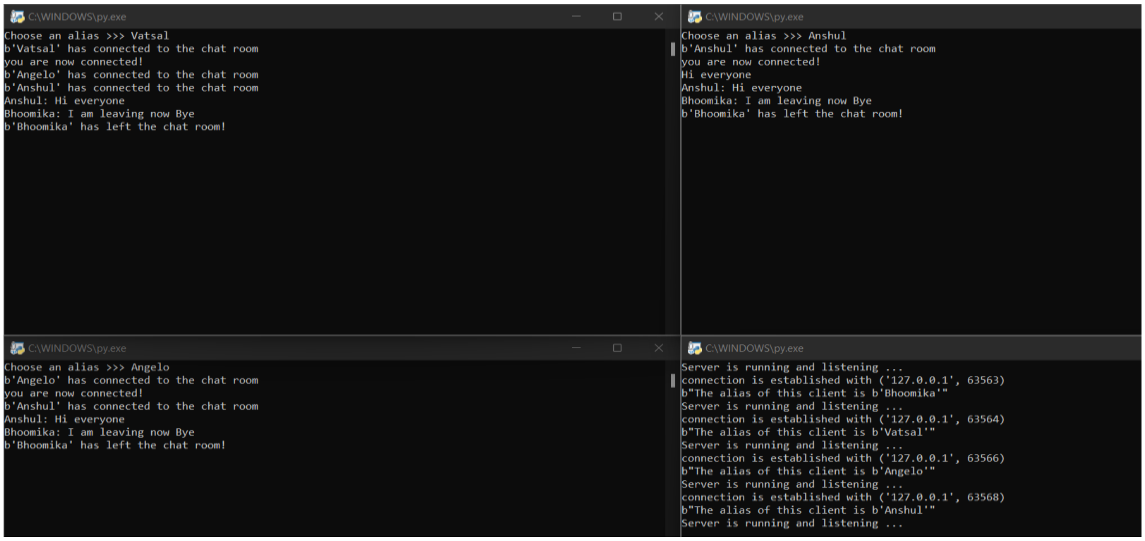
```
Choose an alias >>> Vatsal
b'Vatsal' has connected to the chat room
you are now connected!
b'Angelo' has connected to the chat room
b'Anshul' has connected to the chat room
Anshul: Hi everyone
```

```
Choose an alias >>> Anshul
b'Anshul' has connected to the chat room
you are now connected!
Hi everyone
Anshul: Hi everyone
```

```
Choose an alias >>> Angelo
b'Angelo' has connected to the chat room
you are now connected!
b'Anshul' has connected to the chat room
Anshul: Hi everyone
```

```
Choose an alias >>> Bhoomika
b'Bhoomika' has connected to the chat room
you are now connected!
b'Vatsal' has connected to the chat room
b'Angelo' has connected to the chat room
b'Anshul' has connected to the chat room
Anshul: Hi everyone
```

One member of the group leaving



```
Choose an alias >>> Vatsal
b'Vatsal' has connected to the chat room
you are now connected!
b'Angelo' has connected to the chat room
b'Anshul' has connected to the chat room
Anshul: Hi everyone
Bhoomika: I am leaving now Bye
b'Bhoomika' has left the chat room!
```

```
Choose an alias >>> Anshul
b'Anshul' has connected to the chat room
you are now connected!
Hi everyone
Anshul: Hi everyone
Bhoomika: I am leaving now Bye
b'Bhoomika' has left the chat room!
```

```
Choose an alias >>> Angelo
b'Angelo' has connected to the chat room
you are now connected!
b'Anshul' has connected to the chat room
Anshul: Hi everyone
Bhoomika: I am leaving now Bye
b'Bhoomika' has left the chat room!
```

```
Server is running and listening ...
connection is established with ('127.0.0.1', 63563)
b"The alias of this client is b'Bhoomika'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63564)
b"The alias of this client is b'Vatsal'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63566)
b"The alias of this client is b'Angelo'"
Server is running and listening ...
connection is established with ('127.0.0.1', 63568)
b"The alias of this client is b'Anshul'"
Server is running and listening ...
```

Conclusion:

In this experiment successfully implemented group chat in python using socket programming containing a single server and broadcasting messages to other members as well and displaying messages when an error occurs or someone joined or left the group.