DISTRIBUTED CHAT APPLICATION USING PYTHON AND PHP

BY

GOWTHAMRAJ 2016103024

DHANUSH 2016103021

NARESHMOORTHY 2016103551

INTRODUCTION:

Communication is the need of the day. **Online chat** may refer to any kind of [communication](https://en.wikipedia.org/wiki/Communication) over the [Internet](https://en.wikipedia.org/wiki/Internet) that offers a [real-time](https://en.wikipedia.org/wiki/Real-time_text) transmission of [text](https://en.wikipedia.org/wiki/Text-based) messages from sender to receiver. Chat messages are generally short in order to enable other participants to respond quickly. Thereby, a feeling similar to a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as I[nternet forums](https://en.wikipedia.org/wiki/Internet_forum) and [email](https://en.wikipedia.org/wiki/Email). Online chat may address [point-to-point](https://en.wikipedia.org/wiki/Point-to-point_(telecommunications)) communications as well as [multicast](https://en.wikipedia.org/wiki/Multicast)communications from one sender to many receivers and voice and video chat, or may be a feature of a [web conferencing](https://en.wikipedia.org/wiki/Web_conferencing) service.

**Distributed chat allows many person to chat at the same time**

Python helps to create the chat roomin which the individualcan send message to the common chat room.i.e,the chat messages can be send to the group each of the other person received the message.

In php it is possible to create the chat application which allows a person to create a group for him he can send the messge to that group after registration

Modules and description:

1)socket:

In Python language, *socket*(or network socket) is a module used to communicate between two computers. It provides two types of interface to access the network, namely low-level (platform dependent connections — Example: Telnet) and high-level (application dependent connections — Example: HTTP, FTP, SMTP, etc.). This is a simple tutorial to establish the low-level socket connection between server and client to communicate messages using the TCP/IP protocol.

#### — Client —

An client script performs the sequence of functions such as socket(), and connect()to communicate with the server. The description of each functions used in the server script are given bellow:

* socket()– creates a socket using the address family, socket type and protocol
* connect()– connects to a server socket at address

— Server —

An server script performs the sequence of functions such as socket(),bind(), listen(), and accept()(repeats for more than one client) to communicate with the client. The description of each functions used in the server script are given bellow:

* socket()– creates a socket using the address family, socket type and protocol
* bind()– binds the socket to the given address (host name, and port number*pair*)
* listen()– enables a server to accept connections from the client(s)
* accept()– waits and accepts connection request from the client(s)
* gethostname()– retrieves host name of the machine
* gethostbyname()– translates a host name to IPv4 format address
* recv()– receives message sent through TCP
* decode()– decodes the message using the codec
* send()– sends message sent through TCP

**Php modules**

create the new group

insert member

select the group

send messages

**Sample coding:**

**client code:**

import sys

import socket

import select

def chat\_client():

if(len(sys.argv) < 3) :

print 'Usage : python chat\_client.py hostname port'

sys.exit()

host = sys.argv[1]

port = int(sys.argv[2])

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.settimeout(2)

# connect to remote host

try :

s.connect((host, port))

except :

print 'Unable to connect'

sys.exit()

print 'Connected to remote host. You can start sending messages'

sys.stdout.write('[Me] '); sys.stdout.flush()

while 1:

socket\_list = [sys.stdin, s]

# Get the list sockets which are readable

ready\_to\_read,ready\_to\_write,in\_error = select.select(socket\_list , [], [])

for sock in ready\_to\_read:

if sock == s:

# incoming message from remote server, s

data = sock.recv(4096)

if not data :

print '\nDisconnected from chat server'

sys.exit()

else :

#print data

sys.stdout.write(data)

sys.stdout.write('[Me] '); sys.stdout.flush()

else :

# user entered a message

msg = sys.stdin.readline()

s.send(msg)

sys.stdout.write('[Me] '); sys.stdout.flush()

if \_\_name\_\_ == "\_\_main\_\_":

sys.exit(chat\_client())  
  
**Server code:**

import sys

import socket

import select

HOST = ''

SOCKET\_LIST = []

RECV\_BUFFER = 4096

PORT = 9009

def chat\_server():

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

server\_socket.bind((HOST, PORT))

server\_socket.listen(10)

# add server socket object to the list of readable connections

SOCKET\_LIST.append(server\_socket)

print "Chat server started on port " + str(PORT)

while 1:

# get the list sockets which are ready to be read through select

# 4th arg, time\_out = 0 : poll and never block

ready\_to\_read,ready\_to\_write,in\_error = select.select(SOCKET\_LIST,[],[],0)

for sock in ready\_to\_read:

# a new connection request recieved

if sock == server\_socket:

sockfd, addr = server\_socket.accept()

SOCKET\_LIST.append(sockfd)

print "Client (%s, %s) connected" % addr

broadcast(server\_socket, sockfd, "[%s:%s] entered our chatting room\n" % addr)

# a message from a client, not a new connection

else:

# process data recieved from client,

try:

# receiving data from the socket.

data = sock.recv(RECV\_BUFFER)

if data:

# there is something in the socket

broadcast(server\_socket, sock, "\r" + '[' + str(sock.getpeername()) + '] ' + data)

else:

# remove the socket that's broken

if sock in SOCKET\_LIST:

SOCKET\_LIST.remove(sock)

# at this stage, no data means probably the connection has been broken

broadcast(server\_socket, sock, "Client (%s, %s) is offline\n" % addr)

# exception

except:

broadcast(server\_socket, sock, "Client (%s, %s) is offline\n" % addr)

continue

server\_socket.close()

# broadcast chat messages to all connected clients

def broadcast (server\_socket, sock, message):

for socket in SOCKET\_LIST:

# send the message only to peer

if socket != server\_socket and socket != sock :

try :

socket.send(message)

except :

# broken socket connection

socket.close()

# broken socket, remove it

if socket in SOCKET\_LIST:

SOCKET\_LIST.remove(socket)

if \_\_name\_\_ == "\_\_main\_\_":

sys.exit(chat\_server())

groups.php:

<?php

$userName = $\_GET['name'];

$con = mysqli\_connect('localhost','root','');

mysqli\_select\_db($con, 'final-group-chat');

echo "Groups of ".$userName;

echo "<br>";

$query2 = mysqli\_query($con, "SELECT \* FROM user WHERE user\_name = '".$userName."'");

while ($row1 = mysqli\_fetch\_array($query2))

{

$query = "SELECT \* FROM `users\_groups` WHERE user\_id = ".$row1['user\_id']."";

$result = mysqli\_query($con, $query);

while ($row = mysqli\_fetch\_array($result))

{

$sql = mysqli\_query($con, "SELECT \* FROM groups WHERE group\_id = ".$row['group\_id']."");

while($row2 = mysqli\_fetch\_array($sql))

{

if ($row['read\_chats'] < $row2['total\_chats'])

{

$unread = $row2['total\_chats'] - $row['read\_chats'];

echo $unread;

}

echo "<a href='chats.php?user=".urlencode($row1['user\_id'])."&group=".urlencode($row2['group\_id'])."'>". $row2['group\_name']."</a><br>";

}

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Groups</title>

</head>

<body>

<form action='NewGroup.php?nname=<?php echo $userName ?>' method="POST">

<button type="submit">New Group</button>

</form>

</body>

</html>

**index.php:**

<?php

$con = mysqli\_connect('localhost','root','');

mysqli\_select\_db($con, 'final-group-chat');

if(isset($\_POST['insertUser']))

{

if($\_POST['username'] != "")

{

$userName = $\_POST['username'];

$query = "SELECT \* FROM user WHERE user\_name = '".$userName."'";

$result = mysqli\_query($con, $query);

if(mysqli\_num\_rows($result)>0)

{

header('Location: groups.php?name='.$userName);

}

else

{

$query1 = "INSERT INTO `user` (`user\_name`) VALUES ('$userName')";

if(!mysqli\_query($con,$query1)){

echo "Error: Unable to Insert NewUser";

}

else{

echo "User Entered successfully";

}

}

}

else

{

echo "Error: Enter Name";

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>USER</title>

</head>

<body>

<form action="index.php" method="POST">

<label for="username">Enter Your Name:</label></br></br>

<input type="text" name="username" id="username"></br></br>

<button name="insertUser" type="submit">Submit</button>

</form>

</body>

</html>

insert chat.php

<?php

$con = mysqli\_connect('localhost','root','');

mysqli\_select\_db($con, 'final-group-chat');

$message = $\_REQUEST['message'];

$userid = $\_REQUEST['user'];

$groupid = $\_REQUEST['group'];

mysqli\_query($con, "INSERT INTO chats (message, user\_id, group\_id) VALUES ('$message', '$userid', '$groupid')");

mysqli\_query($con, "UPDATE groups SET total\_chats = total\_chats + 1 WHERE group\_id = '".$groupid."'");

$query = mysqli\_query($con, "SELECT \* from chats ORDER BY chat\_id DESC");

while ($row = mysqli\_fetch\_array($query))

{

echo "USER ID: " . $row['user\_id'] . "MESSAGE: " . $row['message'] . "</br>";

}

?>

logs.php

<?php

$con = mysqli\_connect('localhost','root','');

mysqli\_select\_db($con, 'final-group-chat');

$group = $\_GET['group'];

$query = mysqli\_query($con, "SELECT \* FROM chats WHERE group\_id = '$group' ORDER BY chat\_id DESC");

while($row = mysqli\_fetch\_array($query))

{

echo "USER ID: " . $row['user\_id'] . "MESSAGE: " . $row['message'] . "</br>";

}

?>

newgroup.php:

<?php

// $userName = $\_GET['nname'];

$con = mysqli\_connect('localhost','root','');

mysqli\_select\_db($con, 'final-group-chat');

echo "<label>Add Members</label></br></br>";

if(isset($\_POST['insertGroup']))

{

if($\_POST['groupName'] != "" && $\_GET['nname'] != "")

{

$userName = $\_GET['nname'];

$groupName = $\_POST['groupName'];

$query2 = mysqli\_query($con, "SELECT user\_id FROM user WHERE user\_name = '".$userName."'");

while ($row1 = mysqli\_fetch\_array($query2))

{

$query1 = mysqli\_query($con, "INSERT INTO groups (group\_name, creator\_id) VALUES ('$groupName', '".$row1['user\_id']."')");

$creatorId = $row1['user\_id'];

}

$query3 = mysqli\_query($con, "SELECT group\_id FROM groups WHERE group\_name = '$groupName'");

while ($row = mysqli\_fetch\_array($query3))

{

$newGroupId = $row['group\_id'];

}

$query = "SELECT \* FROM `user`";

echo "new group id is ".$newGroupId;

$result = mysqli\_query($con, $query);

while ($row = mysqli\_fetch\_array($result))

{

echo "<a href='addMembers.php?user=".urlencode($row['user\_name'])."&group=".urlencode($newGroupId)."'>". $row['user\_name']."</a><br>";

}

echo "</br></br>";

mysqli\_query($con, "INSERT INTO users\_groups (user\_id, group\_id) VALUES ('$creatorId', '$newGroupId')");

}

else

echo "nname or groupName missing";

}

?>

<!DOCTYPE html>

<html>

<head>

<title>NEW GROUP</title>

</head>

<body>

<form name = "groupForm" method="POST">

<label>Enter Group Name:</label></br></br>

<input type="text" name="groupName"></br></br>

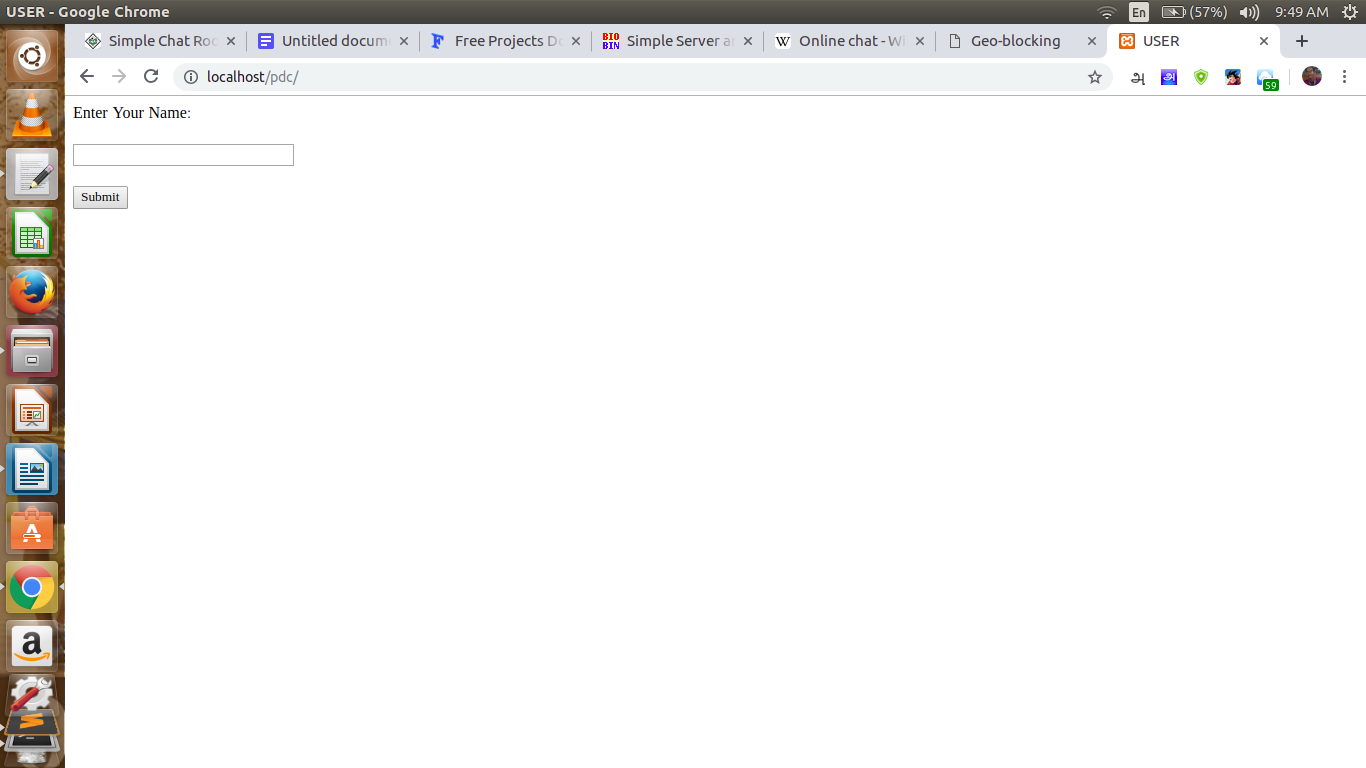
<button name="insertGroup" type="submit">Submit</button>

</form>

</body>

</html>

Screenshots:



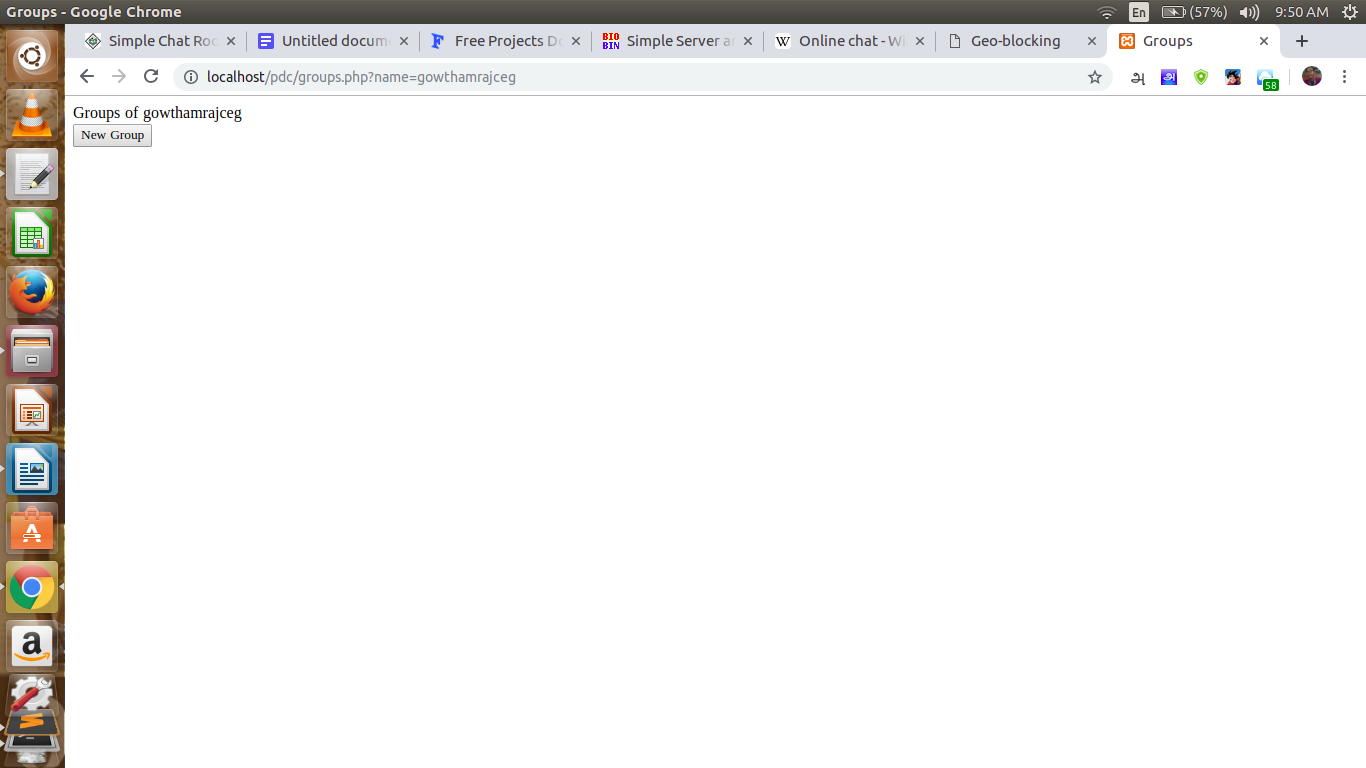


fig: creating group for a user

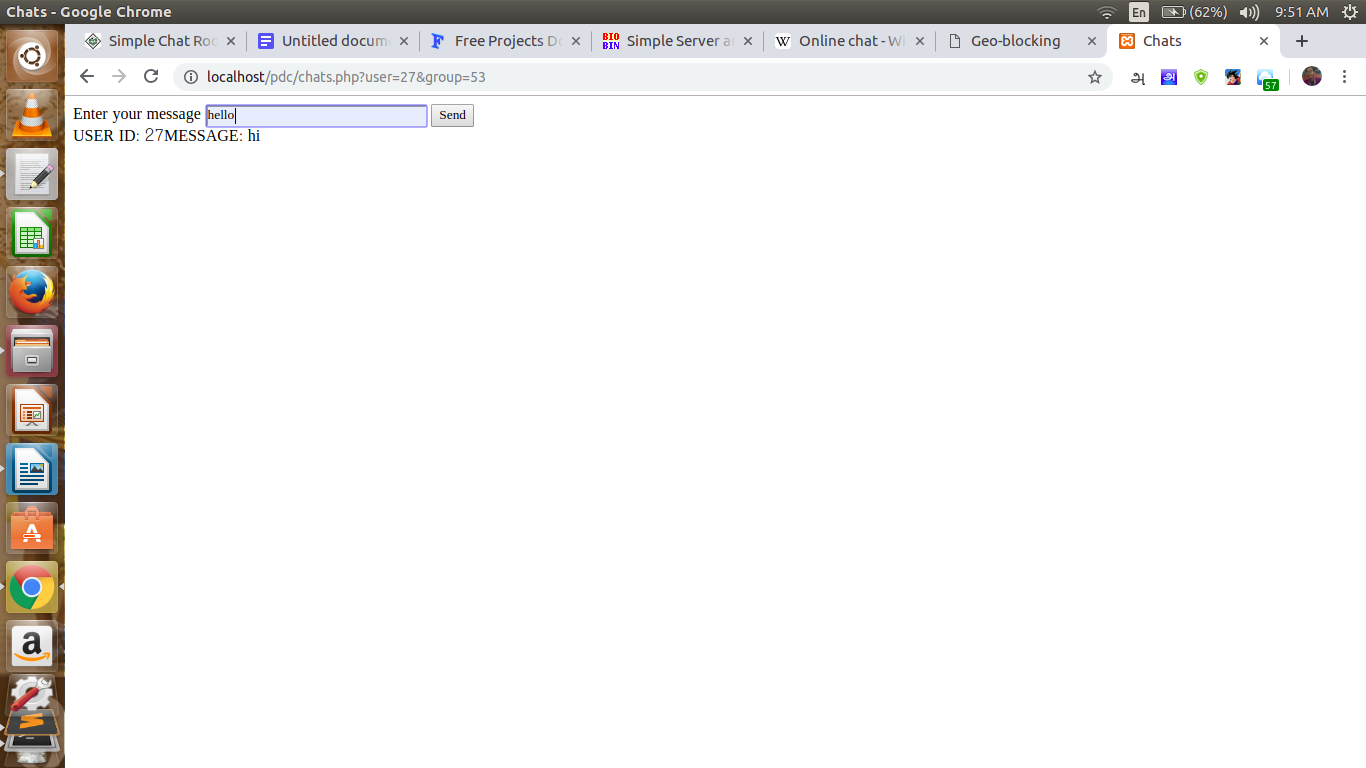


fig: typing message to a particular message

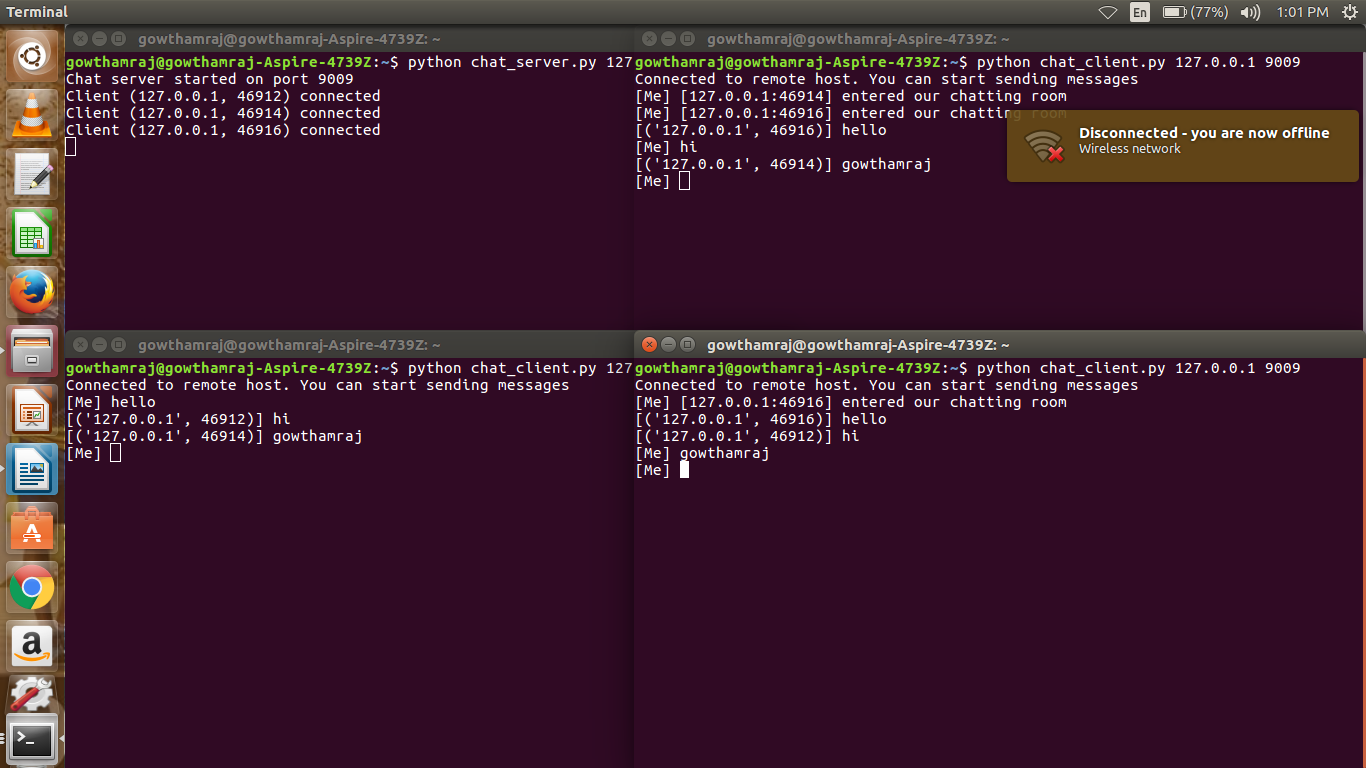


fig: Distributed chat application via python socket

Result :

Thus we implemented a chat system that allows both group and persoinal chat.

Chat lobbies are a very convenient way of communicating with non friends in the RetroShare network. For the moment, we have observed chat lobbies of almost 100 peers, which seems to work fairly well.

Among future improvements, we plan to make the private lobby a bit more handy. One issue at the present state is that when a peer disconnects from a private lobby, he needs to be re-invited by one of his friends participating to the lobby. It would be easier if the friend kept advertising about this lobby to disconnected peers so that they can come again by themselves.