

# Project 1

## Group member

*Haoyu Gong; 5589519; gong0098@umn.edu*  
*Huibing Dong; 5595957; dong0198@umn.edu*  
*Wenlong Wang; 5593784; wang9467@umn.edu*

## Design details

Our PubSub system simulates a simple distributed system through RPC and UDP. We implemented our system in the source code included in the attached files. And here we would give a table that roughly shows all the functions that we implement along with files they belong to, and also demonstrates the role of each file. Then we will give an insight to the detail of our design by explaining the implementation of these functions and their interactions of each other.

file	description	function
PubSub_client.cpp	Get list of group server's ip, call RPC and deal with different commands such as join/subscribe etc.	Getlist(),communicate_prog_1()
PubSub_server.cpp	Implement RPC functions	seek_Client() and six RPC functions
PubSub_svc.cpp	register/deregister to super server	Register(), Deregister()
client.h,client.cpp	define a class Client which maintain ip, port and types of subscription	

Here are the 6 RPCs:

- **join & leave:** Besides the basic function of join\_1\_svc(), we also maintain three vectors: clients, sockets and sockaddr. Each time a client call to join group server, the group server create a socket using client's ip and port to keep communication with the client in the future.
- **subscribe & unsubscribe:** We implement basic function in this part such as dealing with the illegal (un)subscribe format and so on.
- **publish:** When a client publishes articles, the group server scan the three vectors we said before to find out clients that meet the requirements, and then send articles to them through sockets it created in join.

- **ping:** Clients ping the group server every 2 sec. If group server doesn't respond, clients send getlist to superserver and find out another active group server to join.

We get command from .txt file which is implemented in communicate\_prog\_1() (PubSub\_client.cpp). So we just need write commands to test.txt.

The format of command is given below:

format	command description
create	create a new client
join [client_id(interger)]	client calls to join group server and its unique client id should be specified, cliend id begins with 0
leave [client_id(interger)]	client calls to leave group server
subscribe [client_id(interger)] [article]	client subscribes to group server with specified types which is define in article
unsubscribe [client_id(interger)] [article]	client unsubscribes to group server from specified topics
publish [client_id(interger)] [article]	client publishes articles to group server

## Steps to compile and run our code:

1. `make clean`
2. `make`
3. create 3 terminals, then enter the commands below at different terminals in order
4. `./superserver`
5. `./server`
6. `./client [ip address of super server]`
7. kill the `server` process, then within 10s, re-run `./server` to launch a new group server again (This is to test the fail-over mechanism, the client will issue timeout after 10s).

## Sample Test results: (input file is included)

Our tests include the 6 RPC functions, invalid subscriber()/unsubscribe()/publish() detecting, and the test of fail-over mechanism. Possible outputs are shown below.

### Registry server

```
huibingdong — ssh dong0198@cse1-kh4250-01.cselabs.umn.edu — 114x24
dong0198@cse1-kh4250-01:/home/dong0198/src $ ./super_server
Super-server starts.
2020-02-19.19:35:11 : Registered RPC 134.84.182.1:5107 ProgID : 0 Vers : 0
Heart beat message : heartbeat
strServerList is 134.84.182.1;0;0
2020-02-19.19:35:13 : GetList RPC 134.84.182.1:6000
strServerList is 134.84.182.1;0;0
2020-02-19.19:35:13 : GetList RPC 134.84.182.1:6001
strServerList is 134.84.182.1;0;0
2020-02-19.19:35:13 : GetList RPC 134.84.182.1:6002
2020-02-19.19:35:17 : Registered RPC 134.84.182.1:5107 ProgID : 0 Vers : 0
Heart beat message : heartbeat
strServerList is 134.84.182.1;0;0
2020-02-19.19:35:22 : GetList RPC 134.84.182.1:6000
Heart beat message : heartbeat
Heart beat message : heartbeat
^C
dong0198@cse1-kh4250-01:/home/dong0198/src $
```

## Group Server

```
huibingdong — ssh dong0198@cse1-kh4250-01.cselabs.umn.edu — 113x52
dong0198@cse1-kh4250-01:/home/dong0198/src $ ./server
received data: heartbeat
join is called
Client (134.84.182.1,6000) Join successfully!!
clients[0] = 134.84.182.1
join is called
Client (134.84.182.1,6001) Join successfully!!
clients[0] = 134.84.182.1
join is called
Client (134.84.182.1,6002) Join successfully!!
clients[0] = 134.84.182.1
Client 0 subscribe() Illegal failed: Illegal type.
Client 0 subscribe() failed: The first three fields are empty.
Client 0 subscribe() type Sports successfully.
Client 0 subscribe() originators bingbing successfully.
Client 0 subscribe() orgs umn successfully.
Client 1 subscribe() type Sports successfully.
Client 1 subscribe() originators haoyu successfully.
Client 2 subscribe() type Science successfully.
Client 2 subscribe() orgs hkust successfully.
Publish failed: The type Illegal is illegal!
Publish failed: There first three fields are empty
Publish failed: Empty article content
Publish failed: There first three fields are empty
server publishing article is: Sports;;;Sports-target01
server publishing article is: ;bingbing;bingbing-target0
server publishing article is: ;;umn;umn-target0
server publishing article is: Business;haoyu;;business-haoyu-targetNull
server publishing article is: Sports;umn;Sports-umn-target0
server publishing article is: Sports;umn2;Sports-umn2-targetNull
unsubscribe() Sports successfully.
server publishing article is: Sports;;;Sports-target1
Client (134.84.182.1,6001) leaves group server successfully!!
Client (134.84.182.1,6002) leaves group server successfully!!
Successful Ping.
Successful Ping.
^C
dong0198@cse1-kh4250-01:/home/dong0198/src $ ./server
received data: heartbeat
join is called
Client (134.84.182.1,6000) Join successfully!!
clients[0] = 134.84.182.1
received data: heartbeat
Successful Ping.
Successful Ping.
Successful Ping.
Successful Ping.
Successful Ping.
received data: heartbeat
Successful Ping.
^C
dong0198@cse1-kh4250-01:/home/dong0198/src $
```

## Client

```
huibingdong — ssh dong0198@cse1-kh4250-01.cselabs.umn.edu — 113x38
dong0198@cse1-kh4250-01:/home/dong0198/src $ ./client 127.0.0.1
create
Client 0 called join()
create
Client 1 called join()
create
Client 2 called join()
Client 0 called subscribe()
Client 0 called subscribe()
Client 0 called subscribe()
Client 1 called subscribe()
Client 2 called subscribe()
Client 0 published: Illegal;;;IllegalTypeContent
Client 0 published: ;;;onlyContent
Client 0 published: Business;;;
Client 0 published: ;;;;
Client 0 published: Sports;;;Sports-target01
Client 0 published: ;bingbing;;bingbing-target0
Client 1 received: Sports;;;Sports-target01
Client 0 received: Sports;;;Sports-target01
Client 0 received: ;bingbing;;bingbing-target0
Client 1 published: ;umn;umn-target0
Client 0 received: ;umn;umn-target0
Client 1 published: Business;haoyu;;business-haoyu-targetNull
Client 2 published: Sports;umn;Sports-umn-target0
Client 0 received: Sports;umn;Sports-umn-target0
Client 2 published: Sports;umn2;Sports-umn2-targetNull
Client 0 called unsubscribe()
Client 2 published: Sports;;;Sports-target1
Client 1 received: Sports;;;Sports-target1
Client 1 called leave()
Client 2 called leave()
Client 0 need to join another group.
134.84.182.1
Client 0 has re-joined to a group server
^C
dong0198@cse1-kh4250-01:/home/dong0198/src $
```

If you want to test more, please feel free to modified the `test.txt` and add commands in the corresponding syntax.