

### Protocole du Serveur :

Le serveur attend d'abord la chaîne "hello" du client.

Si le serveur reçoit "hello", il répond avec "hello".

Ensuite, le serveur attend de recevoir la liste des numéros de protocole.

Après avoir reçu la liste des numéros de protocole, le serveur vérifie chaque numéro de protocole.

Le serveur sélectionne le protocole le plus récent parmi ceux qu'il connaît.

Si aucun des protocoles n'est valide, le serveur répond avec "0" et ferme la session.

Sinon, le serveur répond avec le numéro du protocole sélectionné.

Après avoir envoyé le numéro du protocole sélectionné ou "0", le serveur attend de recevoir le nombre de nœuds par le client , si la réponse n'était pas 0 sinon il ferme la session.

Après avoir reçu le nombre de nœuds, il attend de recevoir les nœuds.

Après avoir reçu tous les nœuds, le serveur traite les données et ferme la session.

### Protocole du Client :

Le client envoie "hello" au serveur.

Si le client reçoit "hello" en réponse, il poursuit.

Sinon, le client termine la communication.

Après avoir envoyé "hello", le client envoie la liste des numéros de protocole au serveur.

Si le client reçoit un numéro de protocole sélectionné, il envoie le nombre de nœuds qu'il va envoyer.

Si le client reçoit "0", il ferme la session et termine la communication.

Après il envoie les nœuds un par un au serveur.

Après avoir envoyé tous les nœuds, le client ferme la connexion.

Connexion avec des mauvais paramètres

```
toni@ubuntu:~/indu/protocole$ ./proto-client 10.0.2.4
usage: client hostname + number of nodes
toni@ubuntu:~/indu/protocole$
```

Connexion sur serveur fermé

```
toni@ubuntu:~/indu/protocole$ ./proto-client 10.0.2.4 3
client: connect: Connection refused
client: failed to connect
send: Bad file descriptor
recv: Bad file descriptor
bad echo
toni@ubuntu:~/indu/protocole$
```

connexion mauvais mot de passe protocoles

```
357     char buf[MAXDATASIZE];
358     char protoPass[] = "hello";
359     // Envoie du message de bienvenue
360     send(buf, "Bienvenue", 10, 0);

PROBLEMS 30 OUTPUT DEBUG CONSOLE TERMINAL PORTS

toni@ubuntu-server:~/indu/protocole$ ./proto-client 10.0.2.4 3
client: connecting to 10.0.2.4
bad echo
toni@ubuntu-server:~/indu/protocole$
```

protocole commun trouvé , le 33

```
toni@ubuntu-server:~/indu/protocole$ ./proto-client 10.0.2.4 3
client: connecting to 10.0.2.4
send list proto
protocol number is : 33
toni@ubuntu-server:~/indu/protocole$
```

le serveur reçoit le bon passe , il renvoie « hello »

il reçoit le nombre de protocole du client, puis la liste des protocoles

il compare et trouve le 33 en commun .

Il renvoie le 33 au client qui lui envoie le nombre de nœuds , puis les nœuds un par un

il les affiche et ferme la connexion

```
toni@serveur2:~/indu/protocole$ ./proto-server
server: waiting for connections...
server: got connection from 10.0.2.15
Server : good pass received , send : 5 chars
received nbr elements in protocole List from client : 4
[1, 2, 33, 15]
[1, 22, 24, 33, 45, 18, 45]
protocole is : 33
received nbr of node from client : 3
Node 1/3{
    int:    0
    string: protocol
    float:  0.123450
}
Node 2/3{
    int:    1
    string: protocol1
    float:  0.333333
}
Node 3/3{
    int:    2
    string: protocol2
    float:  0.666667
}
end of receive .

```

connexion envoi 4 nœuds : le serveur est toujours à l'écoute et reçoit les 4 nœuds

```
end of receive .
server: got connection from 10.0.2.15
Server : good pass received , send : 5 chars
received nbr elements in protocole List from client : 4
[1, 2, 33, 15]
[1, 22, 24, 33, 45, 18, 45]
protocole is : 33
received nbr of node from client : 4
Node 1/4{
    int:    0
    string: protocol
    float:  0.123450
}
Node 2/4{
    int:    1
    string: protocol1
    float:  0.333333
}
Node 3/4{
    int:    2
    string: protocol2
    float:  0.666667
}
Node 4/4{
    int:    3
    string: protocol3
    float:  1.000000
}
end of receive .
[]
```

connexion sans protocole commun

```
28 int proto[]={11,2,333,15}; // liste des protocoles connus par le client
29 int nbrProto = sizeof(proto)/sizeof(int); //nbr d'entier ds la liste de proto
```

PROBLEMS 30 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
toni@ubuntu-server:~/indu/protocole$ ./proto-client 10.0.2.4 4
client: connecting to 10.0.2.4
send list proto
protocol number is : 0
invalide protocol.
toni@ubuntu-server:~/indu/protocole$
```

réponse du serveur 0

```
server: got connection from 10.0.2.15
Server : good pass received , send : 5 chars
received nbr elements in protocole List from client : 4
[11, 2, 333, 15]
[1, 22, 24, 33, 45, 18, 45]
protocole is : 0
invalide protocol.
[]
```