

## Spécification du Protocole(Version 1.0)

### Résumé

Ce protocole permet de mettre en oeuvre un système de messagerie instantanée entre les clients et le serveur pour que ces derniers puissent se connecter et échanger des messages.

### Sommaire

1. Introduction . . . . .	2
2. Description . . . . .	2
2.1. Contraintes . . . . .	2
2.2. Format de 1er Paquet . . . . .	3
2.2.1. type = 1 (0x01): Connexion . . . . .	4
2.2.2. type = 2 (0x02): ConnexionInvalide (Nom de client déjà utilisé) . . . . .	4
2.2.3. type = 3 (0x03): Déconnexion . . . . .	4
2.2.4. type = 4 (0x04): Demande de liste . . . . .	4
2.2.5. type = 5 (0x05): Accepter une invitation . . . . .	4
2.2.6. type = 6 (0x06): Refuser une invitation . . . . .	4
2.2.7. type = 7 (0x07): Quitter une messagerie . . . . .	4
2.2.8. type = 8 (0x08): Demander de participer à une messagerie privée . . . . .	5
2.2.9. type = 9 (0x09): Dissoudre une messagerie privée . . . . .	5
2.2.10. type = 10 (0x0A): Réponse de récepteur (ACK). . . . .	5
2.2.11. type = 11 (0x0B): EnvoieInvalide (Nom de client récepteur n'existe pas) . . . . .	5
2.2.12. type = 12 (0x0C): offrir deux modes de messagerie privée . . . . .	5
2.2.13. type = 13 (0x0D): compter nombre d'une messagerie privée dans le serveur . . . . .	5
2.2.14. type = 14 (0x0E): InvitationValide. . . . .	6
2.2.15. type = 15 (0x0F): InvitationInValide . . . . .	6
2.2.16. type = 16 (0x10): compter nombre du client à cette messagerie . . . . .	6
2.2.17. type = 17 (0x11): serveur ferme une messagerie . . . . .	6
2.2.18. type = 18 (0x12): EnvoieInvalide (Nom de client récepteur n'existe pas) . . . . .	6
2.2.19. type = 19 (0x13): Client privé reçoit un message public . . . . .	6
2.2.20. type = 20 (0x14): Messagerie privée se termine lorsqu'il y a plus de clients . . . . .	7
2.2.21. type = 21 (0x15): Nombre de client est supérieur au limitation de nombre de groupe . . . . .	7

2.3.	Format de 2er Paquet . . . . .	7
2.3.1.	type = 22 (0x16): Envoyer un message . . . . .	8
2.3.2.	type = 23 (0x17): Serveur diffuse un message . . . . .	8
2.4.	Format de 3er Paquet . . . . .	8
2.4.1.	type = 24 (0x18): Invitation à une messagerie privée . .	9
2.4.2.	type = 25 (0x19): Choisir une mode de serveur . . . . .	9
2.4.3.	type = 26 (0x1A): Changer la mode de serveur dans une messagerie privée . . . . .	9
2.4.4.	type = 27 (0x1B): Créer un nom pour une messagerie . . .	9
2.4.5.	type = 28 (0x1C): Changer nom de messagerie . . . . .	9
2.5.	Format de 4er Paquet . . . . .	10
2.5.1.	type = 29 (0x1D): Liste d'utilisateur . . . . .	10
3.	Fiabilité . . . . .	11
4.	Serveur Configuration . . . . .	11
5.	Exemple des scénarios d'usage . . . . .	11
5.1.	scénarios 1 . . . . .	11
5.2.	scénarios 2 . . . . .	19
6.	Références . . . . .	23
	Adresse d'Auteur . . . . .	23

## 1. Introduction

L'objectif du protocole est de permettre aux utilisateurs de se connecter à un serveur et pouvoir échanger des messages. Il permet aussi de créer une messagerie privée ainsi que participer à la messagerie, et puisque des utilisateurs peuvent initier une messagerie privée, en choisissant les deux modes, centralisé ou décentralisé pour les messageries privées.

## 2. Description

### 2.1. Contraintes

On utilise une unique trame pour chaque message, et ce protocole devra fonctionner au-dessus d'UDP. Toutes les adresses IP doivent être des adresses IPv4 valides.

Par défaut, on suppose que dans ce serveur de communication, il y a seule messagerie publique et il peut exister plusieurs messageries privées.

Ce tableau définit la longueur maximale de quelques éléments.

élément	taille
nombre de client	1024
caractères du nom de client	20
Texte message	1024 bytes
nombre de messagerie privée simultanée	100
nombre de limitation de personne dans une messagerie	100

Table 1: Value maximum

## 2.2. Format de 1er Paquet

Dans ce protocole, le format de paquet n'est pas le même pour tous les messages. En fait, il existe quatre formats de paquet en fonction du rôle qu'ils jouent dans les échanges entre le serveur et les clients.

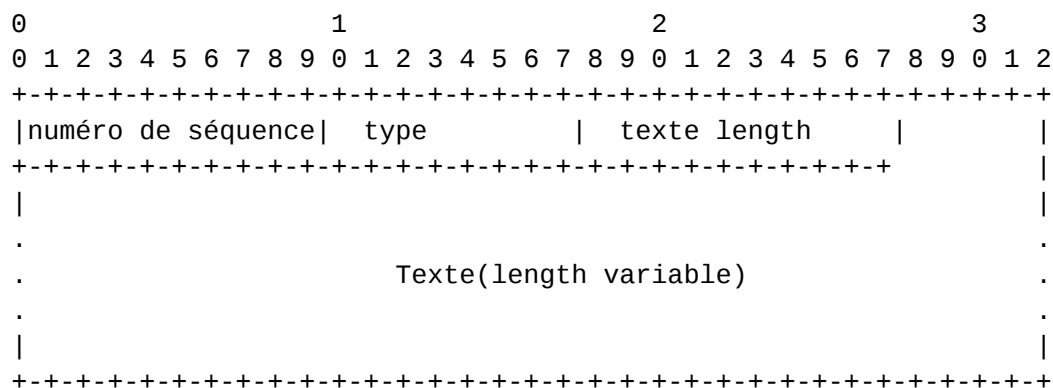


Figure 1

### Numéro de séquence (10 bits)

Ce champ définit le numéro de séquence du message. Pour chaque client ou le serveur, nous définissons un numéro de séquence unique qui augmente lors de l'envoi d'un paquet de message ou la réponse d'un paquet de message (pour un type donné, l'expéditeur soit le client soit le serveur, incrémente le numéro de séquence chaque fois qu'il envoie un nouveau paquet).

### Type (8 bits)

Définir le type de message. Ce champ doit contenir une valeur correcte ci-dessous.

### Texte Length (10 bits)

Définir la longueur de message en byte. Un émetteur doit assurer que ce champ contient une valeur correcte.

### Texte (length variable)

Définir le contenu de message.

#### 2.2.1. type = 1 (0x01): Connexion

Envoyé depuis un client vers le serveur. Par défaut, si un client fait un bon connexion, on pense qu'il reste dans un messagerie publique.  
Le champ de texte doit être TEXTE (nom d'utilisateur) <CRLF>

#### 2.2.2. type = 2 (0x02): ConnexionInvalide (Nom de client déjà utilisé)

Envoyé depuis le serveur vers un client. Lorsqu'un client se connecte et indique son nom, mais ce nom est déjà utilisé, le serveur indique une connexion invalide.  
Le champ de texte doit être TEXTE (nom de client déjà utilisé) <CRLF>.

#### 2.2.3. type = 3 (0x03): Déconnexion

Envoyé depuis un client vers le serveur. Ce message indique la volonté d'un utilisateur de quitter le service de message instantané.  
Le champ de texte doit être TEXTE (nom d'utilisateur) <CRLF>

#### 2.2.4. type = 4 (0x04): Demande de liste

Envoyé depuis un client vers le serveur. Ce message demande la liste des utilisateurs actuellement connectés au serveur.  
Le champ de texte doit être vide.

#### 2.2.5. type = 5 (0x05): Accepter une invitation

Envoyé depuis un client vers le serveur pour indiquer son acception de participer à une messagerie privée.  
Le champ de texte doit être vide.

#### 2.2.6. type = 6 (0x06): Refuser une invitation

Envoyé depuis un client vers le serveur pour indiquer son refus de participer à une messagerie privée.  
Le champ de texte doit être vide.

#### 2.2.7. type = 7 (0x07): Quitter une messagerie

Envoyé depuis un client vers le serveur pour indiquer la volonté d'un utilisateur de quitter une messagerie instantané.  
Le champ de texte doit être vide.

#### 2.2.8. type = 8 (0x08): Demander de participer à une messagerie privée

Envoyé depuis un client vers le serveur pour indique un utilisateur voudrait demander de participer à une messagerie privée. Noms de messageries doivent être identifiés par le serveur via l'adresse IP.

Le champ de texte doit être TEXTE(nom de messagerie) <CRLF>.

#### 2.2.9. type = 9 (0x09): Dissoudre une messagerie privée

Envoyé depuis un client vers le serveur pour indique un manager d'utilisateur dans une messagerie privée voudrait dissoudre cette messagerie privée.

Le champ de texte doit être vide.

#### 2.2.10. type = 10 (0x0A): Réponse positif de récepteur (ACK)

Indique qu'un message a été reçu. L'ACK doit être envoyé quand le récepteur reçoit un message.

Le champ de texte soit vide soit TEXTE(message) <CRLF>.

#### 2.2.11. type = 11 (0x0B): EnvoieInvalide (Nom de client récepteur n'existe pas)

Lorsqu'un client envoie un message a l'autre client, mais le nom de client n'existe pas, le serveur indique une erreur.

Le champ de texte doit être TEXTE(Nom de client récepteur n'existe pas) <CRLF>.

#### 2.2.12. type = 12 (0x0C): offrir deux modes de messagerie privée

Lorsqu'un client demande une invitation a une messagerie privée, le serveur devra offrir deux options afin qu'un client puisse, en précisant les modes centralisé ou décentralisé.

Le champ de texte doit être TEXTE(centralisé ou décentralisé) <CRLF>

#### 2.2.13. type = 13 (0x0D): compter nombre d'une messagerie privée dans le serveur

Pour assurer de la qualité de communication, chaque section de tous les temps, le serveur doit compter nombre d'une messagerie qu'il existe. Si le nombre d'une messagerie privée est déjà arrivée au limitation maximale, le serveur va refuser de demande d'invitation à une messagerie.

Le champ de texte doit être vide.

#### 2.2.14. type = 14 (0x0E): InvitationValide

Après un client finit une demande d'invitation à une messagerie privée et de choisir un mode (centralisé ou décentralisé), le serveur doit vérifier si le nombre d'une messagerie existant est supérieur au limitation maximale de du nombre de messagerie et compter nombre du client qui va participer à cette messagerie. S'il est inférieur et le nombre du client est plus de seuk personne, le serveur devra répondre une valide invitation au client et donner un nom pour cette messagerie par défaut.

Le champ de texte doit être vide.

#### 2.2.15. type = 15 (0x0F): InvitationInValide

Après un client finit une demande d'invitation à une messagerie privée le serveur doit vérifier si le nombre d'une messagerie existant est supérieur au limitation maximale de du nombre de messagerie et compter le nombre du clients qu'ils vont participer à cette messagerie. S'il est supérieur ou une seule personne participe à cette messagerie, le serveur devra répondre une invalide invitation au client.

Le champ de texte doit être vide.

#### 2.2.16. type = 16 (0x10): compter nombre du client à cette messagerie

Pour assurer de la qualité de communication, chaque section de tous les temps, le serveur doit compter nombre du client à cette messagerie. S'il reste une seule personne, le serveur doit fermer cette messagerie.

Le champ de texte doit être vide.

#### 2.2.17. type = 17 (0x11): serveur ferme une messagerie

Quand une messagerie reste une seule personne, le serveur doit fermer cette messagerie.

Le champ de texte doit être TEXTE(fermer cette messagerie car il reste seule personne)<CRLF>.

#### 2.2.18. type = 18 (0x12): EnvoieInvalide (Nom de client récepteur n'existe pas)

Lorsqu'un client envoie un message a l'autre client, mais le nom de client n'existe pas, le serveur indique une erreur.

Le champ de texte doit être TEXTE(Nom de client récepteur n'existe pas) <CRLF>.

#### 2.2.19. type = 19 (0x13): Client privé reçoit un message public

Lorsqu'un client privé reçoit un message public, le serveur indique une erreur.

Le champ de texte doit être TEXTE(Client privé reçoit un message public) <CRLF>.

### 2.2.20. type = 20 (0x14): Messagerie privée se termine lorsqu'il y a plus de clients

Une messagerie se termine lorsqu'il y a plus de clients, le serveur indique une erreur.

Le champ de texte doit être TEXTE(Messagerie se termine lorsqu'il y a plus de clients) <CRLF>.

### 2.2.21. type = 21 (0x15): Nombre de client est supérieur au limitation de nombre de groupe

Pour une messagerie il y a une limitation de participante. Quand le nombre de client est supérieur à la limitation de nombre de groupe, le serveur indique une erreur.

Le champ de texte doit être TEXTE (Nombre de client est supérieur au limitation de nombre de groupe) <CRLF>.

## 2.3. Format de 2er Paquet

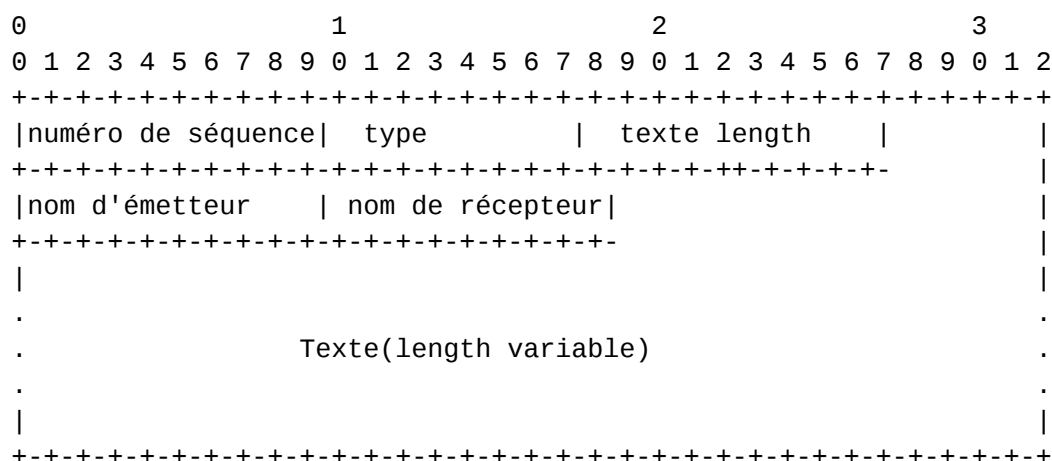


Figure 2

### Nom d'émetteur (10 bits)

Ce champ définit le nom d'émetteur du client qui envoie un message. Noms de client doivent être identifiés par le serveur via l'adresse IP.

### Nom de récepteur (10 bits)

Ce champ définit le nom de récepteur du client qui reçoit un message. Noms de client doivent être identifiés par le serveur via l'adresse IP.

### 2.3.1. type = 22 (0x16): Envoyer un message

Envoyé depuis un client vers le serveur ou depuis un client vers un (des) autre(s) client(s) ou du serveur vers un client. Il contient au minimum une chaîne de caractères représentant le message lui même.

Le champ de texte doit être TEXTE(Message) <CRLF>.

### 2.3.2. type = 23 (0x17): Serveur diffuse un message

Lorsqu'un client envoie un message texte au serveur, le serveur devra le diffuser à tous les autres utilisateurs connectés, en précisant le nom d'utilisateur de l'émetteur initial du message.

Le champ de text doit être TEXTE(message) <CRLF> TEXTE(émetteur initial envoie)  
<CRLF>

## 2.4. Format de 3er Paquet

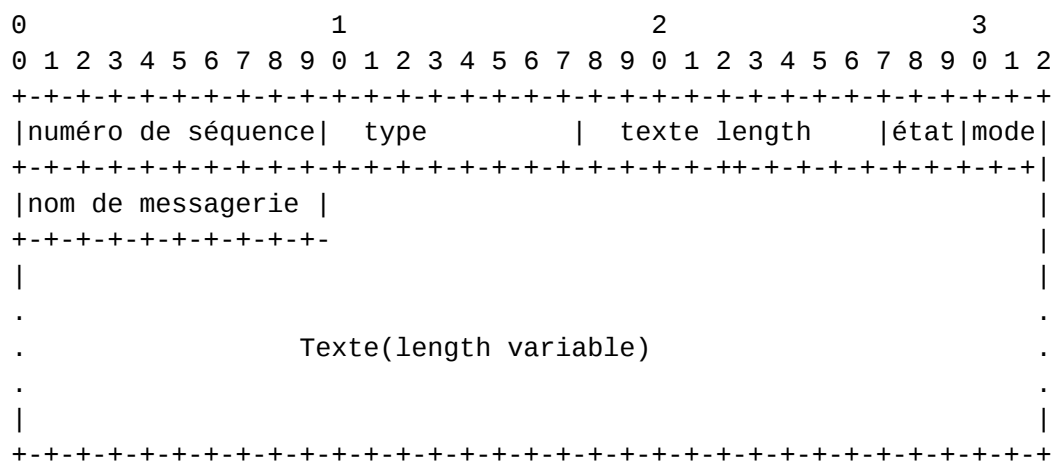


Figure 3

État (2 bits)

Définir le statut de client (messagerie publique, ou une des messageries privées).

```
00--> messagerie publique
```

01--> messagerie privée

Dans le cas d'une messagerie privée, un message envoyé par un des participants devra arriver à tous les utilisateurs ayant accepté l'invitation à la messagerie privée. Ils ne reçoivent plus (ni ne peuvent envoyer) de messages avec les utilisateurs restés dans la messagerie publique. Il pourra y avoir plusieurs messageries privées en parallèle, mais un utilisateur ne peut appartenir qu'à une seule messagerie (publique ou privée).



Mode (2 bits)

Définir le mode de serveur pour les messageries privées:

00 --> une option centralisée

01 --> une autre option décentralisée

Dans le mode centralisé, le serveur reste au milieu des conversations; c'est lui qui reçoit l'ensemble des messages et qui les relaye aux utilisateurs concernés. Dans le mode décentralisé, le serveur est exclu des communications: les messages ne circulent plus à travers le serveur, mais passent directement d'un client à un autre.

#### 2.4.1. type = 24 (0x18): Invitation à une messagerie privée

Envoyé depuis un client vers le serveur (et éventuellement relayé depuis le serveur vers un client), ce message invite à participer à une messagerie privée. Le promoteur de messagerie doit indiquer les noms de personne invitée. De plus, le client qui propose cette invitation va devenir un manager de cette messagerie. Le manager peuvent choisir ou changer un mode de serveur, etc...

Le champ de texte doit être TEXTE(nom d'utilisateur, nom d'utilisateur,...) <CRLF>.

#### 2.4.2. type = 25 (0x19): Choisir un mode de serveur

Envoyé depuis un client vers le serveur. Quand le serveur offre deux modes (centralisé ou décentralisé) pour choisir, un client doit choisir un mode. Le champ de mode doit être 00 ou 01. Le champ de texte doit être vide.

#### 2.4.3. type = 26 (0x1A): Changer le mode de serveur dans une messagerie privée

Envoyé depuis un client vers le serveur. Les clients dans une messagerie privée peuvent changer le mode de serveur dans une messagerie, mais il doit être d'accord par des autres clients dans cette messagerie. Le champ de texte doit être vide.

#### 2.4.4. type = 27 (0x1B): Créer un nom pour une messagerie

Envoyé depuis un client vers le serveur. Quand un client fait une invitation à une messagerie privée, il peut créer un nom pour cette messagerie. Sinon, le serveur va donner un nom par défaut. Le champ de texte doit être EXTE(nom de messagerie) <CRLF>.

#### 2.4.5. type = 28 (0x1C): Changer nom de messagerie

Envoyé depuis un client vers le serveur. Tous les clients dans une messagerie peuvent changer le nom de messagerie. Le champ de texte doit être EXTE(nom de messagerie) <CRLF>.

## 2.5. Format de 4er Paquet

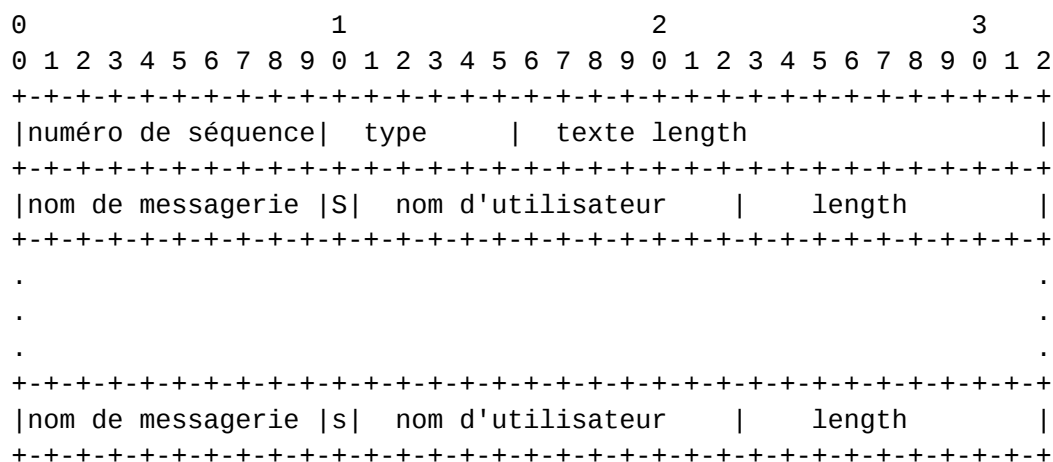


Figure 4

Nom de messagerie (10 bits)

Ce champ contient la liste des noms de messagerie.

Ce champ peut être vide.

S--statut (1 bits)

Ce champ contient le statut de client

0 -- messagerie publique

1 -- messageries privées

Nom d'utilisateur (12 bits)

Ce champ contient la liste des noms d'utilisateur.

Length (9 bits)

Définir le length de nom de messagerie et de nom d'utilisateur.

## 2.5.1. type = 29 (0x1D): Liste d'utilisateur

Quand un client demande de liste des utilisateurs actuellement connectés au serveur. Le serveur doit renvoyer la liste des noms d'utilisateur et le nom de messagerie qu'ils restent, ainsi que leur statut.

### 3. Fiabilité

Ce protocole devra fonctionner au-dessus d'UDP et assurer la fiabilité de la liaison. Ceci signifie que si un message est perdu sur le réseau entre le client et le serveur ou entre le serveur et le client, ce message devra être retransmis jusqu'à réception de ce dernier ou alors jusqu'à ce qu'un nombre d'essais maximum ait été atteint. De plus, afin de ne pas surcharger le serveur, les clients devraient attendre au moins 2 seconds avant renvoyer un message.

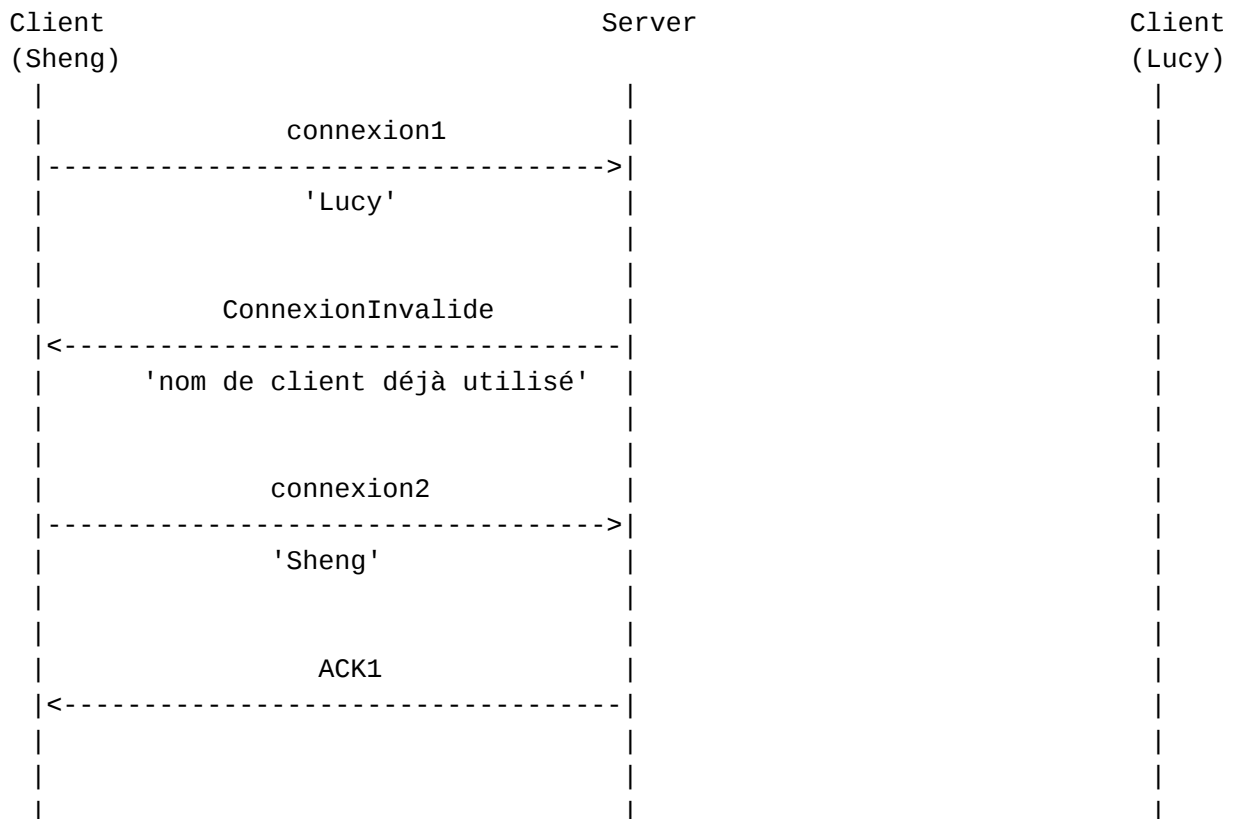
### 4. Serveur Configuration

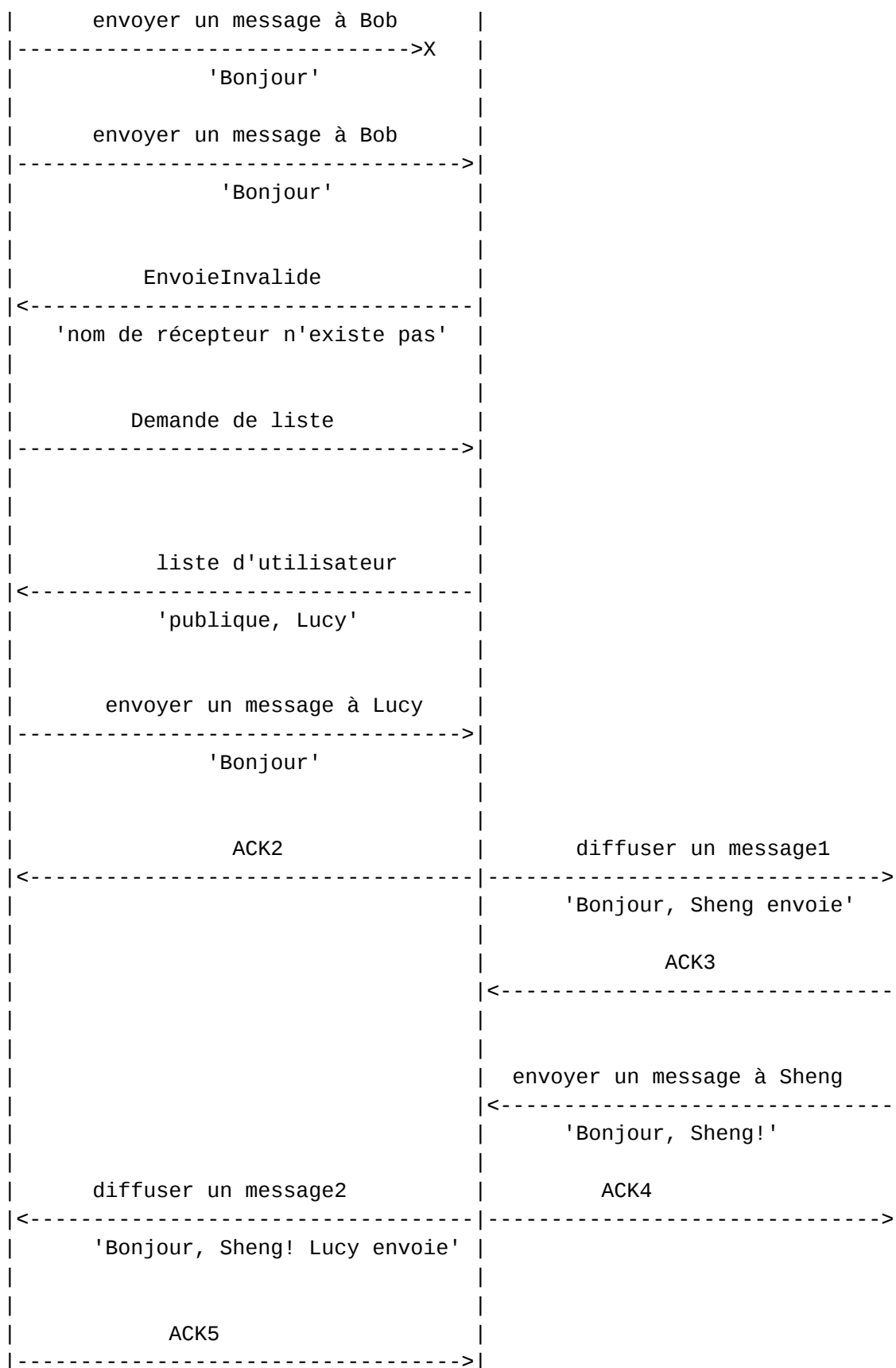
Chaque serveur doit maintenir une base de données locale mappant noms à l'adresse IP. C'est-à-dire un nom d'utilisateur doit correspondre avec une seule adresse IP.

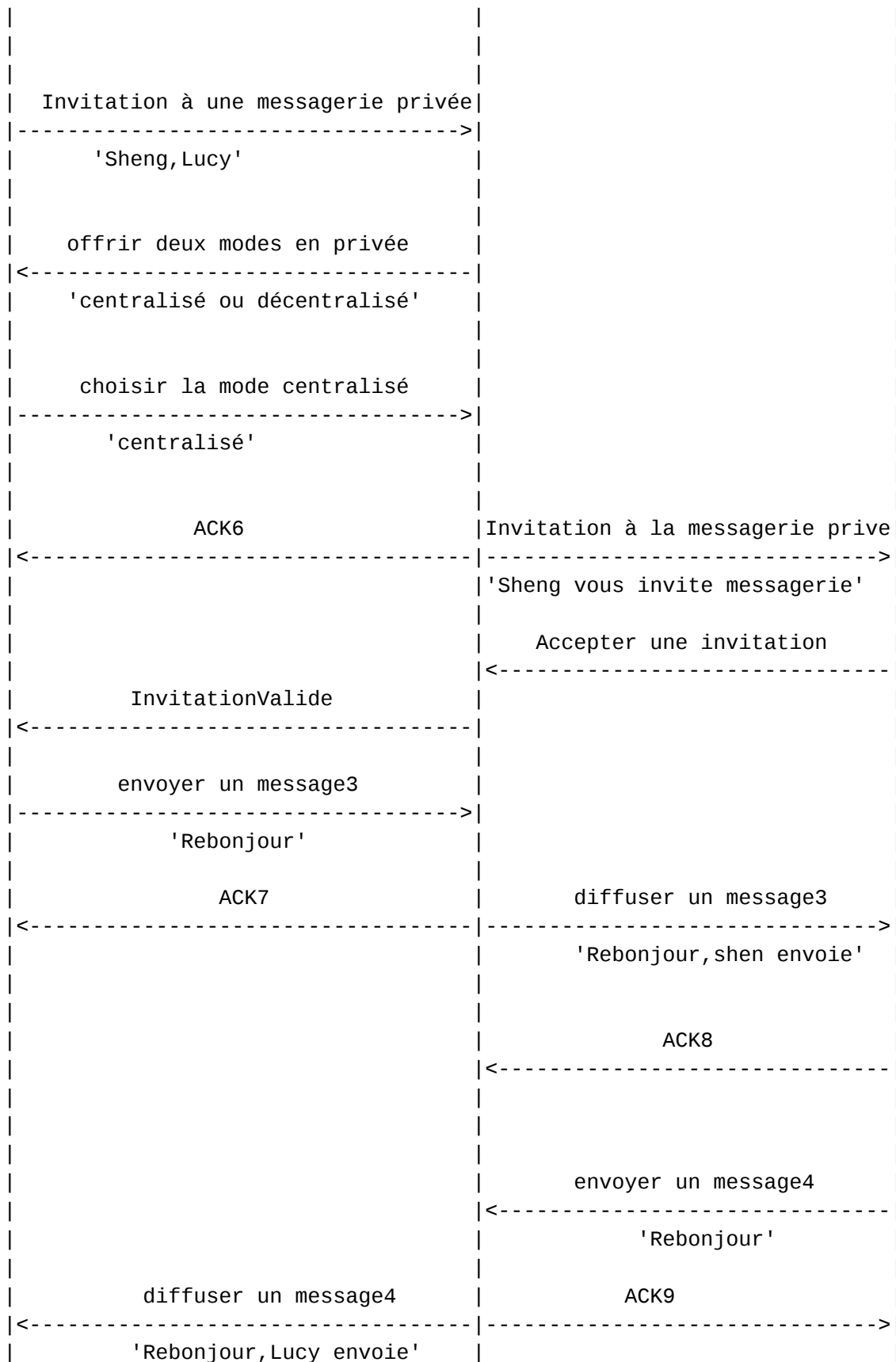
### 5. Exemple des scénarios d'usage

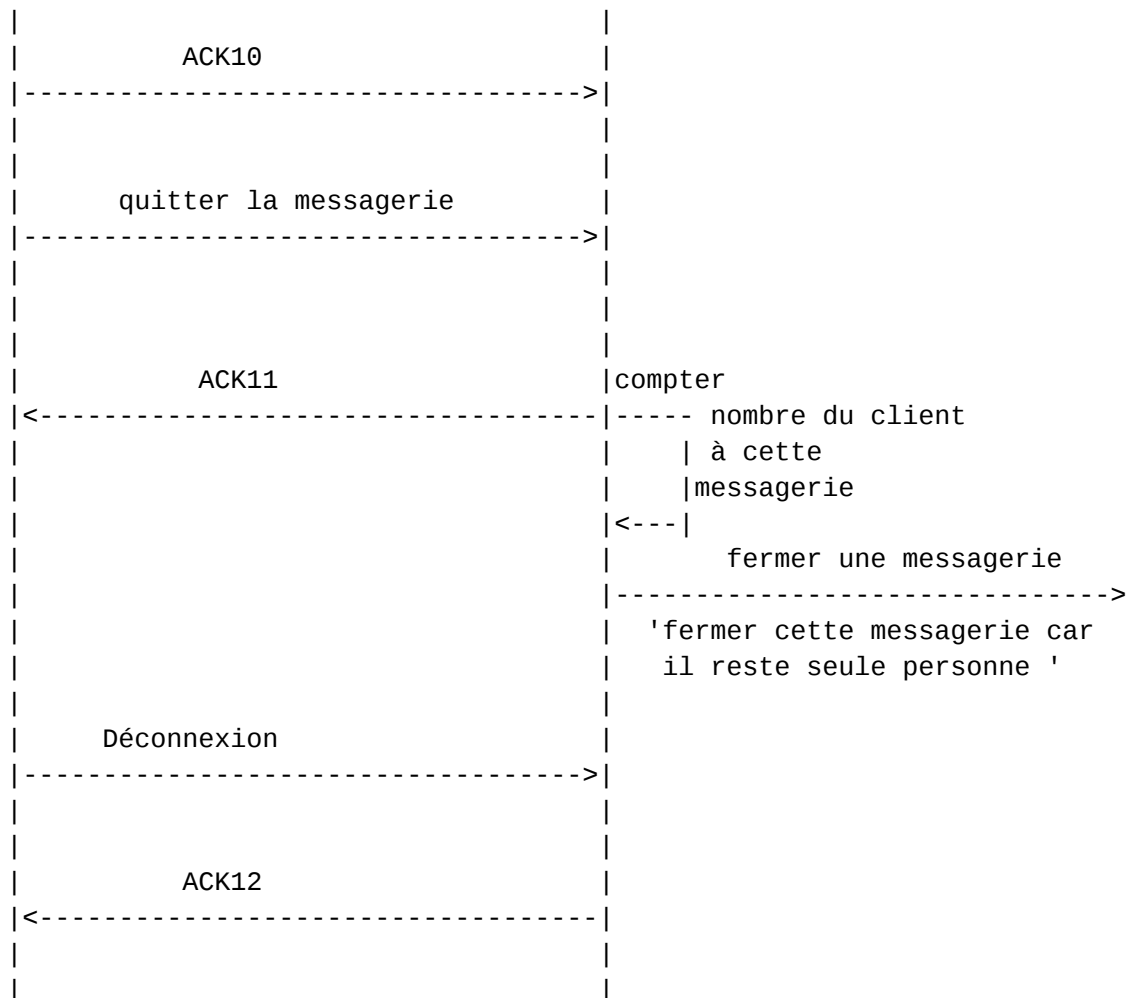
#### 5.1 Scénario 1 : Communication Simple

Pour simuler un scénario d'usage, on suppose que le client Lucy a déjà connecté le serveur et son statut est public.









détail de scénario 1 (en hexadecimal 0x):

connexion1:

numéro de séquence = 00

type = 01

texte length = 04

texte: 6c 75 63 79 0a

ConnexionInvalide :

numéro de séquence = 00

type = 02

texte length = 19

texte: 6e 6f 6d 20 64 65 20 63 6c 69 65 6e 74 20 64 65 6a 61 20 75  
74 69 6c 69 73 65

connexion2 :

numéro de séquence = 01  
type = 01  
texte length = 04  
texte: 53 68 65 6e 67

ACK1 :

numéro de séquence = 01  
type = 0A

envoyer un message à Bob :

numéro de séquence = 02  
type = 16  
texte length = 06  
texte: 42 6f 6e 6a 6f 75 72

EnvoieInvalide :

numéro de séquence = 02  
type = 12  
texte length = 1D  
texte: 6e 6f 6d 20 64 65 20 72 65 63 65 70 74 65 75 72 20 6e 27 65  
78 69 73 74 65 20 70 61 73

Demande de liste :

numéro de séquence = 03  
type = 04

liste d'utilisateur :

numéro de séquence = 03  
type = 1D  
texte length = 0D  
texte: 70 75 62 6c 69 71 75 65 2c 20 4c 75 63 79

envoyer un message a Lucy :

numéro de séquence = 04  
type = 16  
texte length = 06  
texte: 42 6f 6e 6a 6f 75 72

ACK2 :

numéro de séquence = 04  
type = 0A

diffuser un message1 :

numéro de séquence = 04

type = 17

texte length = 14

texte: 42 6f 6e 6a 6f 75 72 2c 20 53 68 65 6e 67 20 65 6e 76 6f 69  
65

ACK3 :

numéro de séquence = 04

type = 0A

envoyer un message à Sheng :

numéro de séquence = 05

type = 16

texte length = 0E

texte: 42 6f 6e 6a 6f 75 72 2c 20 53 68 65 6e 67 21

diffuser un message2 :

numéro de séquence = 05

type = 17

texte length = 1A

texte: 42 6f 6e 6a 6f 75 72 2c 20 53 68 65 6e 67 21 20 4c 75 63 79  
20 65 6e 76 6f 69 65

ACK4 :

numéro de séquence = 05

type = 0A

ACK5 :

numéro de séquence = 05

type = 0A

Invitation à une messagerie privée :

numéro de séquence = 06

type = 18

texte length = 09

texte: 53 68 65 6e 67 2c 4c 75 63 79

offrir deux modes en privée :

numéro de séquence = 06



type = 0C  
texte length = 19  
texte: 63 65 6e 74 72 61 6c 69 73 65 20 6f 75 20 64 65 63 65 6e 74  
72 61 6c 69 73 65

choisir la mode centralisé :

numéro de séquence = 07  
type = 19  
texte length = 09  
texte: 63 65 6e 74 72 61 6c 69 73 65

ACK6 :

numéro de séquence = 07  
type = 0A

Invitation à la messagerie privée :

numéro de séquence = 08  
type = 18  
texte length = 1B  
texte: 53 68 65 6e 67 20 76 6f 75 73 20 69 6e 76 69 74 65 20 6d 65  
73 73 61 67 65 72 69 65

Accepter une invitation :

numéro de séquence = 08  
type = 05

InvitationValide :

numéro de séquence = 08  
type = 0E

envoyer un message3 :

numéro de séquence = 09  
type = 16  
texte length = 08  
texte: 52 65 62 6f 6e 6a 6f 75 72

ACK7 :

numéro de séquence = 09  
type = 0A

diffuser un message3 :

numéro de séquence = 0A

type = 17

texte length = 14

texte: 52 65 62 6f 6e 6a 6f 75 72 2c 73 68 65 6e 20 65 6e 76 6f 69  
65

ACK8 :

numéro de séquence = 0A

type = 0A

envoyer un message4 :

numéro de séquence = 0B

type = 16

texte length = 08

texte: 52 65 62 6f 6e 6a 6f 75 72

ACK9 :

numéro de séquence = 0B

type = 0A

diffuser un message4 :

numéro de séquence = 0C

type = 17

texte length = 14

texte: 52 65 62 6f 6e 6a 6f 75 72 2c 4c 75 63 79 20 65 6e 76 6f 69  
65

ACK10 :

numéro de séquence = 0C

type = 0A

quitter la messagerie :

numéro de séquence = 0D

type = 07

ACK11 :

numéro de séquence = 0D

type = 0A

compter nombre du client à cette messagerie :

numéro de séquence = 0E  
type = 10

fermer une messagerie :

numéro de séquence = 0F  
type = 11  
texte length = 22  
texte: 66 65 72 6d 65 72 20 63 65 74 74 65 20 6d 65 73 73 61 67 65  
72 69 65 20 63 61 72 20 69 6c 20 72 65 73 74 65 20 73 65 75 6c 65  
20 70 65 72 73 6f 6e 6e 65

Déconnexion :

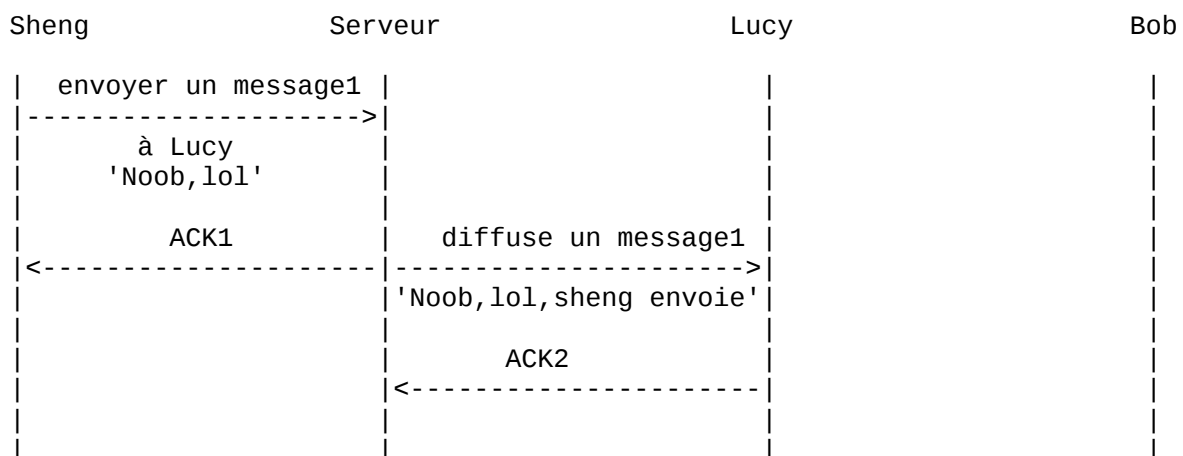
numéro de séquence = 10  
type = 03

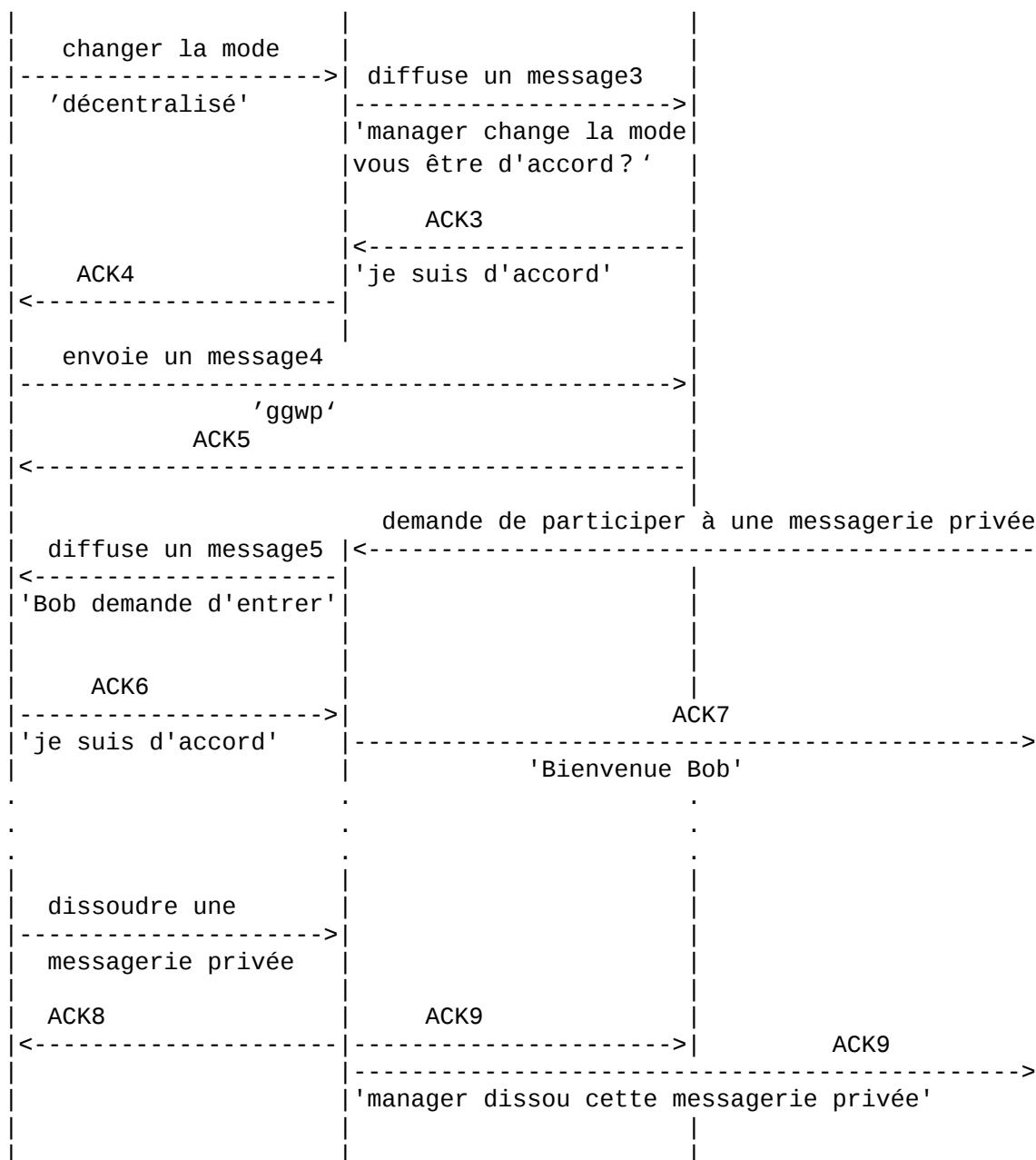
ACK12 :

numéro de séquence = 10  
type = 0A

## 5.2 Scénario 2: Communiatio dans une messagerie privée.

Ce scénario explique l'opération dans une messagerie. Pour un manager dont un client fait un invitation à une messagerie, peut choisir et changer la mode de serveur. Il peut aussi dissoudre une messagerie privée. Pour bien l'explquer, on suppose qu'il exsite 3 clients, Sheng, Lucy, Bob, dans le serveur, et Bob existe dans la messagerie 1 dont son nom est "dota2". De plus, Sheng et Lucy existe dans la messagerie 2 dont son nom est "lol". La mode de serveur est centralisé.





détail de scénario 2 (en hexadecimal 0x):

envoyer un message1 à Lucy :

numéro de séquence = 00

type = 16

texte length = 07

texte: 4e 6f 6f 62 2c 6c 6f 6c

ACK1 :

numéro de séquence = 00  
type = 0A

diffuse un message1 :

numéro de séquence = 01  
type = 17  
texte length = 14  
texte: 4e 6f 6f 62 2c 6c 6f 6c 2c 73 68 65 6e 67 20 65 6e 76 6f 69 65

ACK2 :

numéro de séquence = 01  
type = 0A

changer la mode :

numéro de séquence = 02  
type = 1A  
mode = 01

diffuse un message3 :

numéro de séquence = 03  
type = 17  
texte length = 2A  
texte: 6d 61 6e 61 67 65 72 20 63 68 61 6e 67 65 20 6c 61 20 6d 6f 64 65 20 76  
6f 75 73 20 ea 74 72 65 20 64 27 61 63 63 6f 72 64 ff 1f

ACK3 :

numéro de séquence = 03  
type = 0A  
texte: 6a 65 20 73 75 69 73 20 64 27 61 63 63 6f 72 64

ACK4 :

numéro de séquence = 02  
type = 0A

envoie un message4 :

numéro de séquence = 04  
type = 16  
texte length = 04  
texte: 67 67 77 70

ACK5 :

numéro de séquence = 04  
type = 0A

demande de participer à une messagerie privée :

numéro de séquence = 05  
type = 08

diffuse un message5 :

numéro de séquence = 06  
type = 17  
texte length = 13  
texte: 42 6f 62 20 64 65 6d 61 6e 64 65 20 64 27 65 6e 74 72 65 72

ACK6 :

numéro de séquence = 06  
type = 0A  
texte: 6a 65 20 73 75 69 73 20 64 27 61 63 63 6f 72 64

ACK7 :

numéro de séquence = 05  
type = 0A  
texte: 42 69 65 6e 76 65 6e 75 65 20 42 6f 62

dissoudre une messagerie privée :

numéro de séquence = 07  
type = 09

ACK8

numéro de séquence = 07  
type = 0A

ACK9

numéro de séquence = 07  
type = 0A  
texte: 6d 61 6e 61 67 65 72 20 64 69 73 73 6f 75 20 63 65 74 74 65 20 6d 65  
73 73 61 67 65 72 69 65 20 70 72 69 76 e9 65

## 6. Références

### Adresse d'Auteur

Sheng SHEN  
Télécom Bretagne  
Brest  
France

Email: sheng.shen@telecom-bretagne.eu