S. SHEN
Telecom Bretagne
Fevrier 2017

Specification du Protocole(Version 1.0)

Abstract

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

Table of Contents

1. Introduction

L'objectif du protocole est de permettre aux utilisateurs de se connecter a un serveur et pouvoir echanger des messages. Il permet aussi de creer une messagerie privee ainsi que participer a la messagerie, et puisque des utilisateurs peuvent initier une messagerie privee, en choisissant les deux modes, centralise ou decentralise pour les messageries privees.

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 etre des adresses IPv4 valides.

Ce tableau definit la longueur maximale de quelques elements.

element	taille
nombre de client	1024
caracteres du nom de client	20
Text message	1024 bytes
nombre de messagerie simultane	100
nombre de limtation d'une messagerie	100

Table 1: Value maximum

2.2. Format de 1er Paquet

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

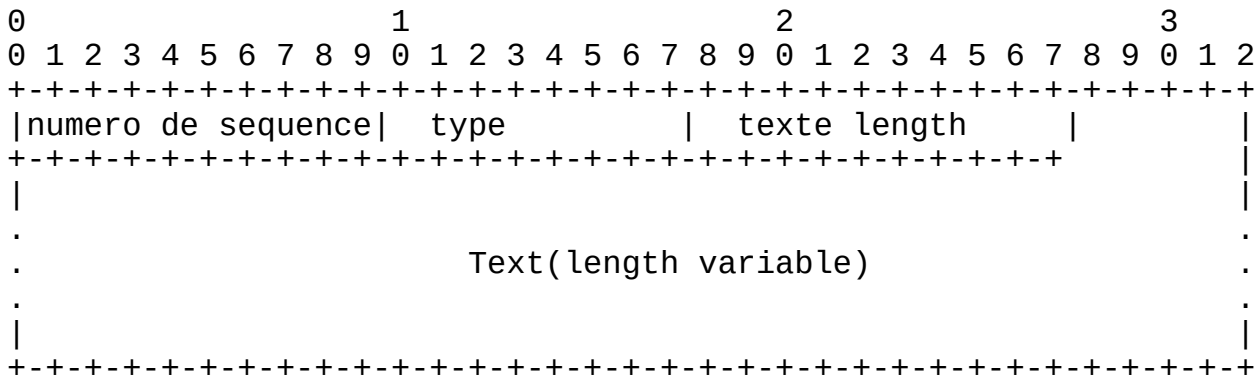


Figure 1

Numero de sequence (10 bits) Ce champ definie le numero de sequence du message. Pour chaque client ou le serveur, nous definissons un numero de sequence unique qui augmente lors de l'envoi d'un paquet de message ou la reponse d'un paquet de message (pour un type donne, l'expediteur soit le client soit le serveur, incremente le numero de sequence chaque fois qu'il envoie un nouveau paquet).

Type (8 bits) Definir le type de message. Ce champ doit contenir une value correcte ci-dessous.

Texte Length (10 bits) Definir la longueur de message en byte. Un emetteur doit assurer que ce champ contient une value correcte.

Texte (length variable): Definir le contenu de messsage.

2.2.1. type = 1 (0x01): Connexion

Envoye depuis un client vers le serveur. Le champ de texte doit etre TEXTE (nom d'utilisateur) <CRLF>

2.2.2. type = 2 (0x02): ConnexionInvalide (Nom de client deja utilise)

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

2.2.3. type = 3 (0x03): Deconnexion

Envoye depuis un client vers le serveur. Ce message indique la volonte d'un utilisateur de quitter le service de message instantane. Le champ de texte doit etre TEXTE (nom d'utilisateur) <CRLF>

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

Envoye depuis un client vers le serveur. Ce message demande la liste des utilisateurs actuellement connectes au serveur. Le champ de texte doit etre vide.

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

Envoye depuis un client vers le serveur pour indiquer son acception de participer a une messagerie privee. Le champ de texte doit etre vide.

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

Envoye depuis un client vers le serveur pour indiquer son refus de participer a une messagerie privee. Le champ de texte doit etre vide.

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

Envoye depuis un client vers le serveur pour indique la volonte d'un utilisateur de quitter une messagerie instantane. Le champ de texte doit etre vide.

2.2.8. type = 8 (0x08): Reponse de recepteur (ACK)

Indique qu'un message a ete recu. L'ACK doit etre envoye quand le recepteur recoit un message. Le champ de texte doit etre vide.

2.2.9. type = 9 (0x09): Serveur diffuse un message

Lorsqu'un client envoie un message texte au serveur, le serveur devra le diffuser a tous les autres utilisateurs connectes, en precisant le nom d'utilisateur de l'emetteur initial du message. Le champ de text doit etre TEXTE(message) <CRLF> TEXTE(emetteur initial) <CRLF>

2.2.10. type = 10 (0x0A): offrir deux modes en privee

Lorsqu'un client demande une invitation a une messagerie privee, le serveur devra offrir deux options afin qu'un client puisse, en precisant les modes centralise ou decentralise. Le champ de texte doit etre TEXTE(centralise ou decentralise) <CRLF>

2.2.11. type = 11 (0x0B): InvitationValide

Apres un client finit une demande d'invitation a une messagerie privee et de choisir un mode (centralise ou decentralise), le serveur

devra repondre une valide invitation au client. Le champ de texte doit etre vide.

2.2.12. type = 12 (0x0C): compter nombre du client a cette messagerie

Pour assurer de la qualite de communication, chaque section de tous les temps, le serveur doit compter nombre du client a cette messagerie. S'il reste une seule personne, le serveur doit fermer cette messagerie. Le champ de texte doit etre vide.

2.2.13. type = 13 (0x0D): fermer une messagerie

Quand une messagerie reste une seule personne, le serveur doit fermer cette messagerie. Le champ de texte doit etre TEXTE(fermer cette messagerie car il reste seule personne)<CRLF>.

2.2.14. type = 14 (0x0E): EnvoieInvalide (Nom de client recepteur 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 etre TEXTE(Nom de client recepteur n'existe pas) <CRLF>.

2.2.15. type = 15 (0x0F): Client privee recoit un message public

Lorsqu'un client prive recoit un message public, le serveur indique une erreur. Le champ de texte doit etre TEXTE(Client privee recoit un message public) <CRLF>.

2.2.16. type = 16 (0x10): Messagerie se termine losqu'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 etre TEXTE(Messagerie se termine losqu'il y a plus de clients) <CRLF>.

2.2.17. type = 17 (0x11): Nombre de client est superieur au limitation de nombre de groupe

Pour une messagerie il y a une limitation de participante. Quand le nombre de client est superieur a la limitation de nombre de groupe, le serveur indique une erreur. Le champ de texte doit etre TEXTE(Nombre de client est superieur au limitation de nombre de groupe) <CRLF>.

2.3. Format de 2er Paquet

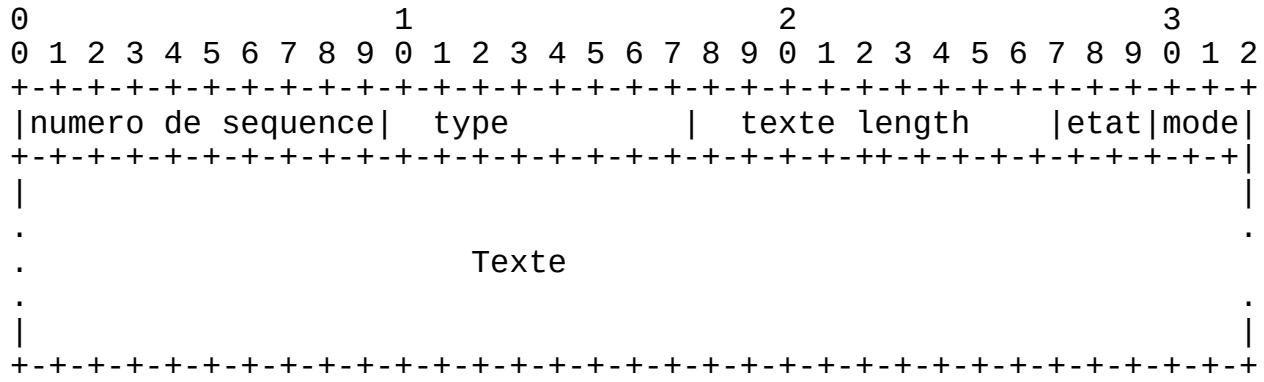


Figure 2

Etat (2 bits)

Definier le statut de client (messagerie publique, ou une des messageries privees).

00--> messagerie publique

01--> messagerie privee

Dans le cas d'une messagerie privee, un message envoye par un des participants devra arriver a tous les utilisateurs ayant accepte l'invitation a la messagerie privee. Ils ne recoivent plus (ni ne peuvent envoyes) de messages avec les utilisateurs restes dans la messagerie publique. Il pourra y avoir plusieurs messageries privees en parallele, mais un utilisateur ne peut appartenir qu'a une seule messagerie (publique ou privee).

Mode (2 bits)

Definir le mode de serveur pour les messageries privees:

00 --> une option centralisee

01 --> une autre option decentralisee

Dans le mode centralise, le serveur reste au milieu des conservations; c'est lui qui recoit l'ensemble des messages et qui les relaye aux utilisateurs concernes. Dans le mode decentralise, le serveur est exclu des communications: les messages ne circulent plus a travers le serveur, mais passent directement d'un client a un autre.

2.3.1. type = 18 (0x12): Client envoie un message

Envoye depuis un client vers le serveur ou depuis un client vers un (des) autre(s) client(s). Un client doit choisir le statut (public ou privee). Noms de client doivent etre identifies par le serveur via l'adresse IP. Le champ d'etat doit etre 00 ou 01. Le champ de texte doit etre TEXTE(Message) <CRLF>.

2.3.2. type = 19 (0x13): Invitation a une messagerie

Envoye depuis un client vers le serveur (et eventuellement relaye depuis le serveur vers un client), ce message invite a participer a une messagerie privee. Le champ de texte doit etre vide.

2.3.3. type = 20 (0x14): choisir le mode

Envoye depuis un client vers le serveur. Quand le serveur offre deux modes (centralise ou decentralise) pour choisir, un client doit choisir un mode. Le champ de mode doit etre 00 ou 01. Le champ de texte doit etre vide.

2.4. Format de 3er Paquet

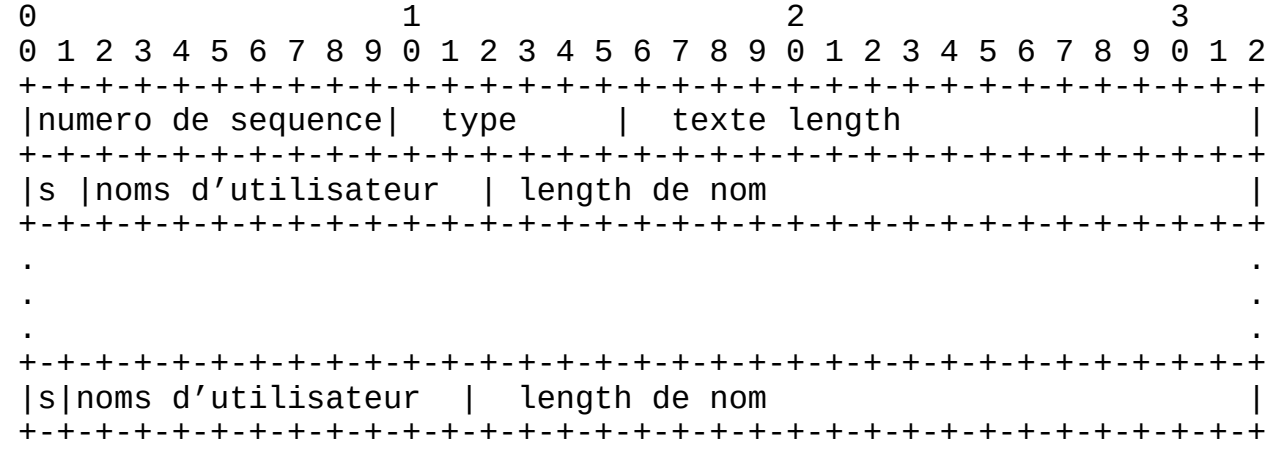


Figure 3

s--statut (2 bits)

Ce champ contient le statut de client 00 -- messagerie publique 01 -- messageries privees

nom d'utilisateur (10 bits)

Ce champ contient la liste des noms d'utilisateur.

2.4.1. type = 21 (0x15): Liste d'utilisateur

Quand un client demande de liste des utilisateurs actuellement connectes au serveur. Le serveur doit renvoyer la liste des noms d'utilisateur, ainsi que leur statut.

3. Fiabilite

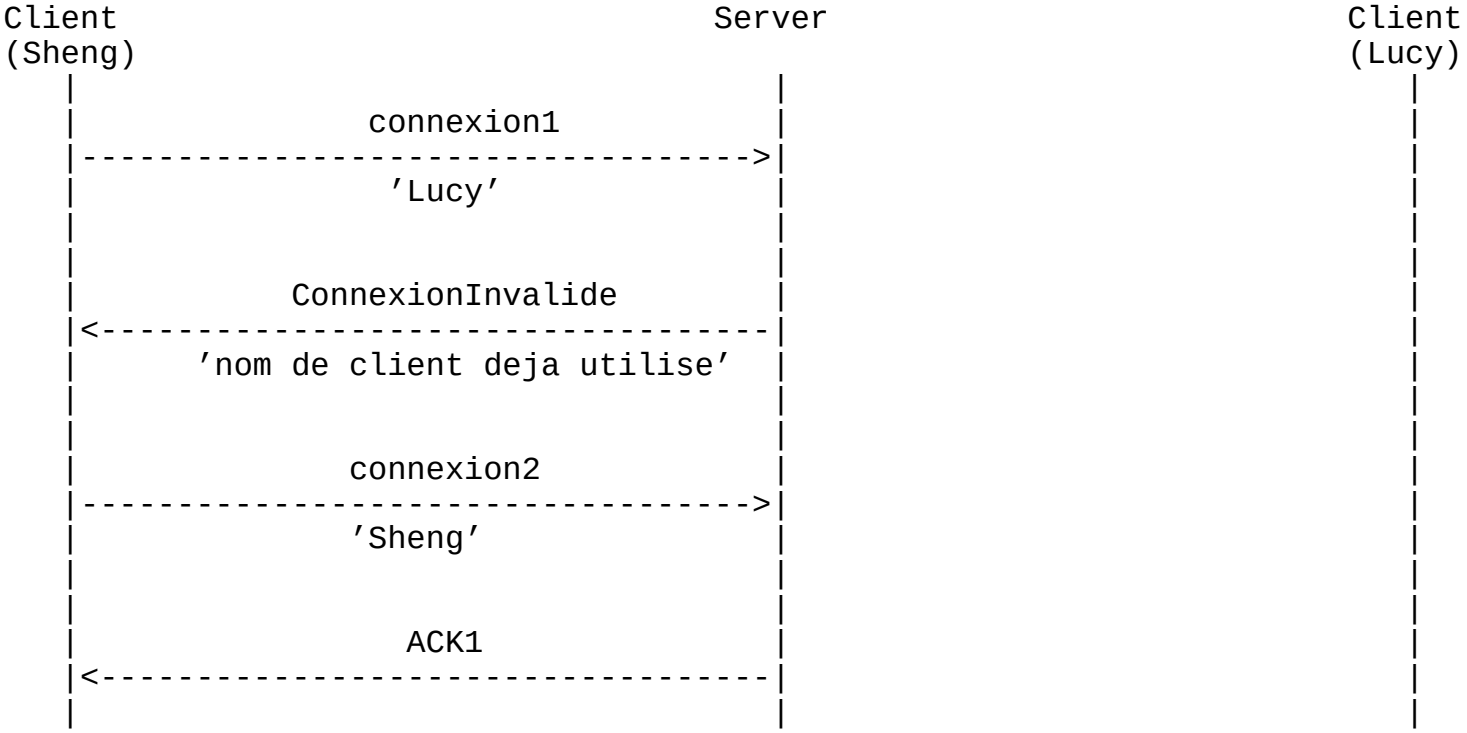
Ce protocole devra fonctionner au-dessus d'UDP et assurer la fiabilite de la liaison. Ceci signifie que si un message est perdu sur le reseau entre le client et le serveur ou entre le serveur et le client,ce message devra etre retransmis jusqu'a reception de ce dernier ou alors jusqu'a ce qu'un nombre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

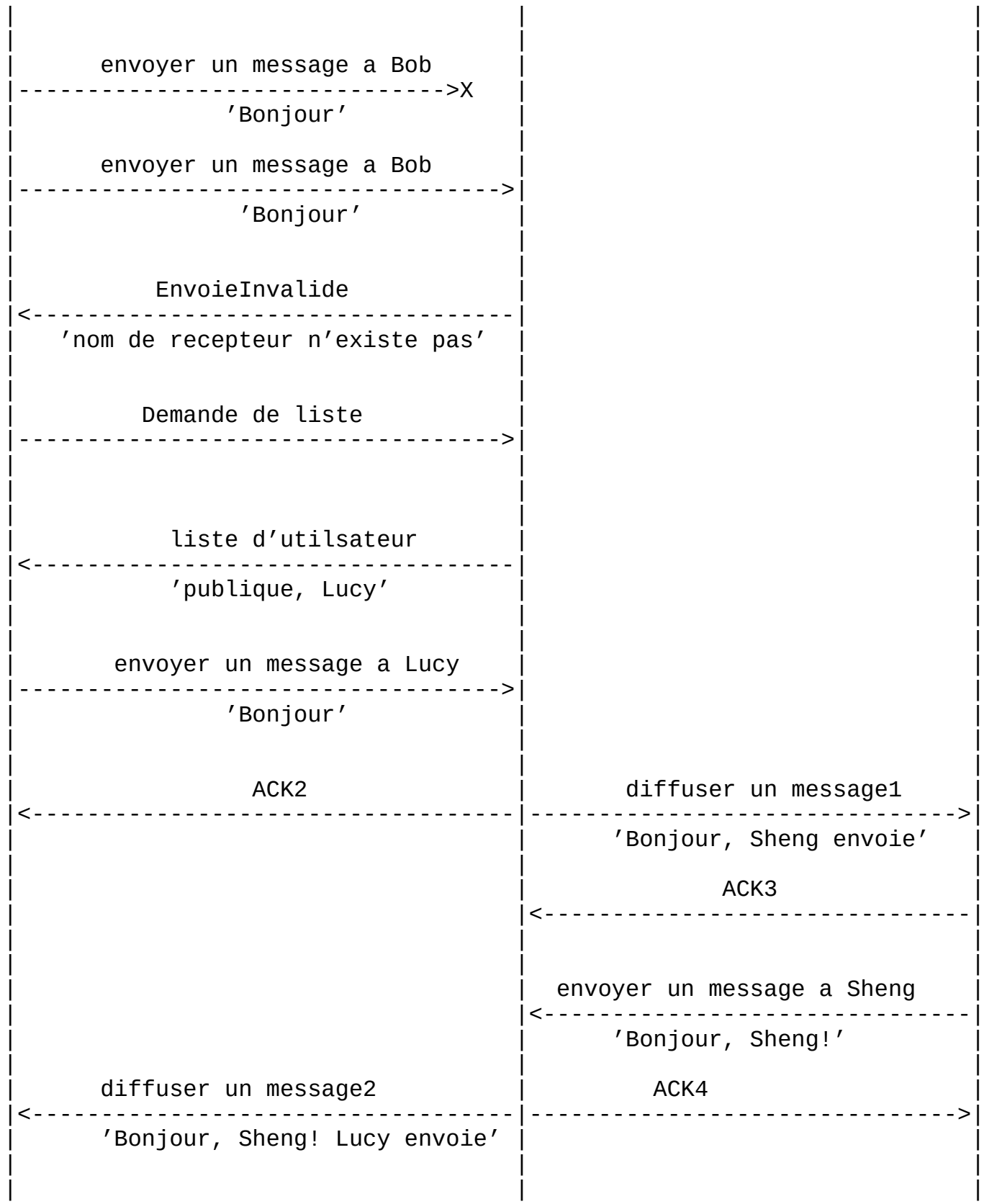
4. Serveur Configuration

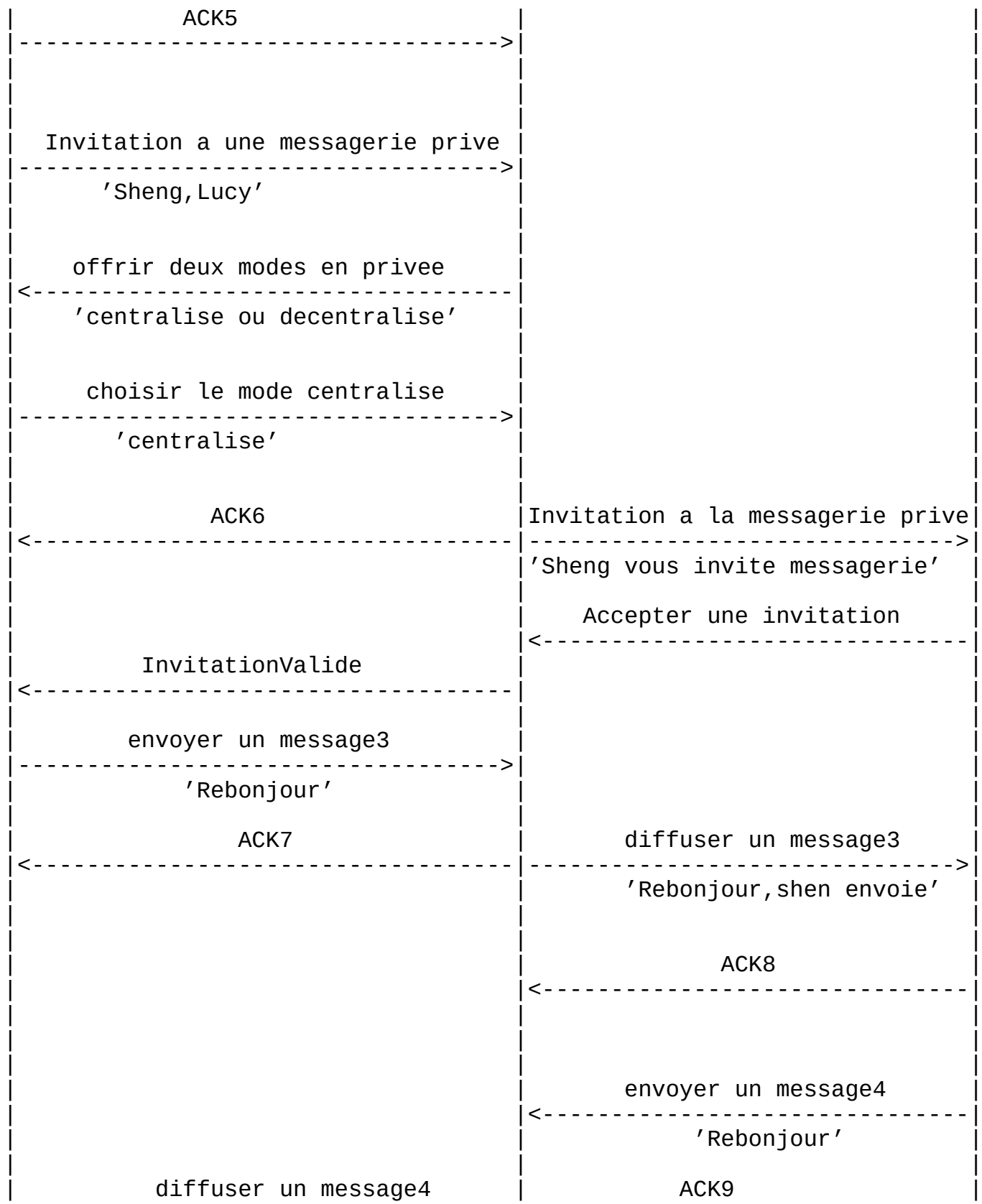
Chaque serveur doit maintenir une base de donnees locale mappant noms a l'adresse IP. C'est-a-dire un nom d'utilisateur doit correspondre avec une seule adresse IP.

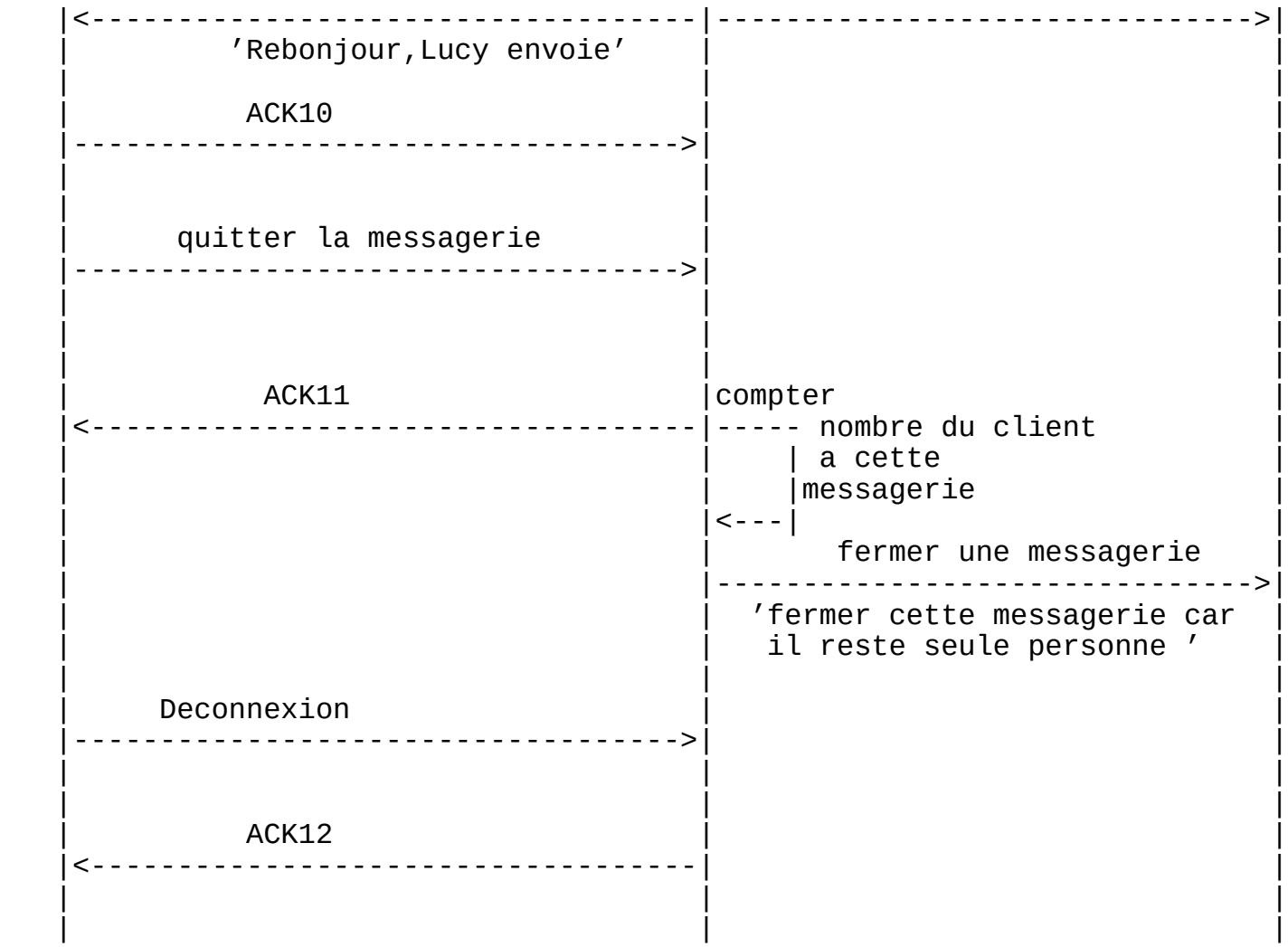
5. Exemple des scenarios d'usage

Pour simuler un scenario d'usage, on suppose que le client Lucy a deja connecte le serveur et son statut est public.




envoyer un message a Bob
'Bonjour'
envoyer un message a Bob
'Bonjour'
EnvoieInvalide
'nom de recepteur n'existe pas'
Demande de liste
liste d'utilsateur
'publique, Lucy'
envoyer un message a Lucy
'Bonjour'
ACK2
diffuser un message1
'Bonjour, Sheng envoie'
ACK3
envoyer un message a Sheng
'Bonjour, Sheng!'
diffuser un message2
'Bonjour, Sheng! Lucy envoie'
ACK4


ACK5
Invitation a une messagerie prive
'Sheng,Lucy'
offrir deux modes en privee
'centralise ou decentralise'
choisir le mode centralise
'centralise'
ACK6
Invitation a la messagerie prive
'Sheng vous invite messagerie'
Accepter une invitation
InvitationValide
envoyer un message3
'Rebonjour'
ACK7
diffuser un message3
'Rebonjour,shen envoie'
ACK8
envoyer un message4
'Rebonjour'
diffuser un message4
ACK9



detail de scenario (en hexadecimal 0x):

connexion1:

numero de sequence = 00
type = 01
texte length = 04
texte: 6c 75 63 79 0a

ConnexionInvalide :

numero de sequence = 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 :

numero de sequence = 01
type = 01
texte length = 04
texte: 53 68 65 6e 67

ACK1 :

numero de sequence = 01
type = 08

envoyer un message a Bob :

numero de sequence = 02
type = 12
texte length = 06
texte: 42 6f 6e 6a 6f 75 72

InvalideEnvoie :

numero de sequence = 02
type = 0E
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 :

numero de sequence = 03
type = 04

liste d’utilsateur :

numero de sequence = 03
type = 15
texte length = 0D
texte: 70 75 62 6c 69 71 75 65 2c 20 4c 75 63 79

envoyer un message a Lucy :

numero de sequence = 04
type = 12
texte length = 06
texte: 42 6f 6e 6a 6f 75 72

ACK2 :

```
   numero de sequence = 04
   type = 08

diffuser un message1 :

   numero de sequence = 04
   type = 09
   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 :

   numero de sequence = 04
   type = 08

envoyer un message a Sheng :

   numero de sequence = 05
   type = 12
   texte length = 0E
   texte: 42 6f 6e 6a 6f 75 72 2c 20 53 68 65 6e 67 21

diffuser un message2 :

   numero de sequence = 05
   type = 09
   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 :

   numero de sequence = 05
   type = 08

ACK5 :

   numero de sequence = 05
   type = 08

Invitation a une messagerie prive :

   numero de sequence = 06
   type = 13
   texte length = 09
   texte: 53 68 65 6e 67 2c 4c 75 63 79
```

offrir deux modes en privee :

numero de sequence = 06
type = 0A
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 le mode centralise :

numero de sequence = 07
type = 14
texte length = 09
texte: 63 65 6e 74 72 61 6c 69 73 65

ACK6 :

numero de sequence = 07
type = 08

Invitation a la messagerie prive :

numero de sequence = 08
type = 13
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 :

numero de sequence = 08
type = 05

InvitationValide :

numero de sequence = 08
type = 0B

envoyer un message3 :

numero de sequence = 09
type = 12
texte length = 08
texte: 52 65 62 6f 6e 6a 6f 75 72

ACK7 :

numero de sequence = 09

```
      type = 08

   diffuser un message3 :

      numero de sequence = 0A
      type = 09
      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 :

      numero de sequence = 0A
      type = 08

   envoyer un message4 :

      numero de sequence = 0B
      type = 12
      texte length = 08
      texte: 52 65 62 6f 6e 6a 6f 75 72

   ACK9 :

      numero de sequence = 0B
      type = 08

   diffuser un message4 :

      numero de sequence = 0C
      type = 09
      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 :

      numero de sequence = 0C
      type = 08

   quitter la messagerie :

      numero de sequence = 0D
      type = 07

   ACK11 :

      numero de sequence = 0D
```

type = 08

compter nombre du client a cette messagerie :

numero de sequence = 0E
type = 0C

fermer une messagerie :

numero de sequence = 0F
type = 0D
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

Deconnexion :

numero de sequence = 10
type = 03

ACK12 :

numero de sequence = 10
type = 08

Author’s Address

Sheng SHEN
Telecom Bretagne
Brest
France

Email: sheng.shen@telecom-bretagne.eu