

RFC du protocole utilisé pour le projet Matou

Maeva KOS, Gaspard DIOP-DUBOIS

I- Résumé:

Le protocole utilisé pour le service Matou permet d'échanger des messages entre plusieurs clients connectés sur un serveur. Il permet également de réaliser des échanges privés de messages et de fichiers entre deux clients sans passer par le serveur. Ce document décrit le protocole et la gestion des paquets (création, envoi etc...).

II -Vue global du protocole:

On peut distinguer trois types d'échanges : client vers serveur, serveur vers clients, client vers client. Chaque type d'échange pourra effectuer différentes requête reconnue par des FLAGS indiqués en début de paquet.

Liste des Flags:

- 0 INIT : flag indiquant une nouvelle connexion d'un client.
- 1 PUB: flag indiquant un message publique entre le client et le serveur.
- 2 PRIVIP: flag indiquant une demande d'adresse IP au serveur.
- 3 REPIP: flag indiquant la réponse du serveur à une demande d'adresse.
- 4 ASK: flag indiquant la demande de connexion pour une communication privée.
- 5 ANS: flag indiquant la réponse à une demande de connexion.
- 6 PRIV: flag indiquant une message d'une communication privée.
- 7 ASKF: flag indiquant la demande d'envoi de fichier.
- 8 ANSF: flag indiquant la réponse à une demande d'envoi de fichier.
- 9 FILE: flag indiquant l'envoi de paquet d'un fichier.
- 11 NINIT : flag indiquant un refus de connexion d'un client.

La réponse négative ou positive à une demande de communication privée, de demande d'envoi de fichier et la demande d'une adresse IP sera spécifié par les codes suivant:

- 0 FAILURECODE: code d'échec de la demande
- 1 SUCCESSCODE: code de réussite de la demande

Lors d'une demande de connexion privée le type d'adresse IP (ipv6 ou ipv4) sera indiqué par les codes suivants :

- 0 CODEIPV4
- 1 CODEIPV6

Les integer sont en Big Endian.

Les chaines correspondant à un type d'encodage sont en US-ASCII.

L'encodage des messages sera spécifié dans des octets du paquet envoyé avec ce message.

L'encodage des nom de fichier sera indiqué dans des octets du paquet envoyé avec ce nom de message.

L'encodage des pseudos sera indiquer dans des octets du paquet.

Tous les transferts de paquets sont en TCP.

Initialisation du chat Matou :

Le serveur et le client sont deux instances différentes.

Le serveur devra d'abord être lancé et sera en attente de connexion de futurs clients.

Lorsqu'un client se connecte, il doit fournir dans la ligne de commande le pseudo qu'il souhaite utiliser. Ce pseudo sera alors envoyé au serveur qui va vérifier si le pseudo reçu n'est pas déjà pris par un autre client, si un client possède déjà ce pseudo alors la connexion du nouveau client est refusée, sinon la connexion est établie.

Le paquet d'initialisation de connexion du client vers le serveur aura la forme suivante :

byte	int	string	string
INIT	taille encodage	nom encodage	pseudo

Dans le cas où la connexion est refusée, le paquet aura la forme suivante :

byte	byte
NINIT	0 ou 1

0 : nom déjà pris

1 : trop de clients

III - 1) Client vers Serveur (une fois que le client est connecté):

Paquet des messages qui seront transmis à tous les clients connectés:

Le client et le serveur sont connectés, le client peut envoyer des messages publics qui seront envoyés par le serveur à tous les autres clients.

byte	int	string	int	string	int	string	string
PUB	taille encodage	nom encodage	taille encodage pseudo	nom encodage pseudo	taille pseudo	pseudo	message

Paquet de demande d'adresse IP pour une communication privée:

Le client1 indique le client qui engage la connexion privée, le client2 indique celui qui reçoit la demande de connexion privée.

Pour une demande de communication privée, le client1 fait d'abord une requête au serveur afin de récupérer l'adresse IP du client2.

byte	int	string	string
PRIVIP	taille encodage pseudo	nom encodage pseudo	pseudo

Dans cet échange, le serveur joue le rôle d'un transmetteur, il fait passer les paquets correspondants à la demande de connexion du client1 vers le client2. Les paquets du serveur vers le client correspondant à la demande de connexion privée sont décrits en détail dans la partie III - 2)

Paquet de réponse à la communication privée :

Après avoir reçu la demande du client1 retransmise par le serveur, le client2 peut donner une

réponse positive ou négative, le client2 va en plus fournir un identifiant unique qui sera donné au client1 afin que personne ne puisse se faire passer pour le client1. Cet identifiant sera nécessaire pour tous les messages de la connexion privée.(partie III-3)

réponse positive :

byte	byte	int	string	int	string	long
REPPRIV	SUCCESSCODE	taille encodage pseudo	nom encodage pseudo	taille pseudo	pseudo	identifiant unique

ou négative :

byte	byte
REPPRIV	FAILURECODE

III - 2) Serveur vers client:

Paquet des messages qui seront transmis à tous les clients connectés:

Le serveur diffuse à tout les autres clients les message publics reçu d'un client.

byte	int	string	int	string	int	string	string
PUB	taille encodage	nom encodage	taille encodage pseudo	nom encodage pseudo	taille pseudo	pseudo	message

Paquet de demande d'adresse IP pour une communication privée:

Le serveur retransmet au client concerné le paquet de demande de connexion

byte	int	string	string
PRIVIP	taille encodage pseudo	nom encodage pseudo	pseudo

Paquet de réponse à la communication privée :

Après avoir reçu une réponse positive du client2, le serveur renvoie au client 1 l'adresse demandé.

byte	byte	int	string	long	
REPPRIV	SUCCESSCODE	taille encodage pseudo	nom encodage pseudo	identifiant unique	
byte	byte	byte	byte	byte	int
CODEIPVX	ip octet 1	ip octet 2	ip octet 3	ip octet 4	--- (4 octets pour ipv4 et 16 octets pour ipv6)
X = 4 ou 6					
string					
pseudo					

OU

byte	byte
REPPRIV	FAILURECODE

III – 3) Client vers client:

Paquet de message client vers client:

Dans chaque paquet de message privée ce trouve un identifiant unique, si l'identifiant unique ne correspond pas à celui qui est associé à cette connexion privée, le paquet est ignoré.

byte	long	int	string	int	string	int

PRIV	identifiant unique	taille encodage	nom encodage	taille encodage pseudo	nom encodage pseudo	taille pseudo

string	string					

pseudo	message					

Paquet d'envoi de fichier:

Le client qui reçoit le fichier connaîtra déjà la taille final du fichier grâce à la demande d'envoi. L'émetteur enverra le fichier par morceaux de 1024 octets.

byte	byte	bytes

FILE	ID UNIQUE	Octets du fichier

IV – Envoi de fichier

Pour une connexion privée, chaque client possède deux connexions qui s'exécuteront en parallèle, l'une servant à l'échange de message, l'autre servant à l'envoi de fichier. Pour établir la deuxième connexion, le port de connexion sera donné lors de la demande et de la réponse d'une demande d'envoi de fichier.

Paquet de demande d'envoi de fichier:

byte	int	int	int	string	string	string	int	string	string

ASKF	port	taille du fichier	taille encodage pseudo	nom encodage pseudo	taille pseudo	nom pseudo	taille encodage fichier	nom encodage fichier	

string									

nom du fichier									

Paquet de réponse de demande d'envoi de fichier:

byte	byte	int

ANSF	SUCCESSCODE	port

byte	byte

ANSF	FAILURECODE
