A. Adedjoumon
B. Gameiro Costa
24 février 2021

LE PROTOCOLE ChatOS

Résumé

ChatOS est un protocole très simple utilisé pour échanger des messages. Ce document décrit le protocole et ses types de paquets. Le document aussi explique les raisons de certaines des décisions de conception.

1. Objet

ChatOS est un protocole simple pour échanger des messages via un serveur précédé d'une identification. Il utilise le protocole TCP pour transférer ses paquets entre le.s client.s et le serveur.
Toutes les chaines de caractères seront dans un premier temps encodé en UTF8.
Il y a trois modes d'envoie de message: général (envoie d'un message encodé à tous les utilisateurs connecté), direct (permet d'envoyer un message à un seul utilisateur connecté au serveur et étant identifié par une chaine de caractères) et privée (permet d'envoyer des messages privé à un autre utilisateur connecté au serveur en utilisant le serveur comme point de relais, donc aucune information ne sera lu/retenu par le serveur).
Comme précisé plus haut, les chaines de caractères (c'est-à-dire message et pseudo) sont encodé en UTF8. Mais il n'est pas impossible qu'une évolution du protocole permette de faire choisir aux utilisateurs leur jeu de caractères (avec UTF8 par défaut). Cependant dans le cas d'une connection privée, le serveur ne faisant office que de relais, l'encodage importe peu vu que le serveur se contente de réenvoyer les paquets reçu à l'identique.

2. Aperçu du protocole

La connection au serveur commence par une phase d'authentification, pendant laquelle l'utilisateur va choisir un pseudonyme. Si le pseudonyme est correct (càd non pris par un autre utilisateur), la connection est accordée.
Une fois la connection accordée, le client va recevoir des paquets venant du serveur contenant des messages (direct ou général) et il pourra à son tour envoyer des paquets au serveur représentant des messages, ou bien demander une connection privée avec un autre utilisateur.

Si il y a une demande de connection privée venant de A vers B, la suite des instructions est: le client A demande au serveur de se connecter au client B. Si B existe alors il reçoit une demande de connection privé qu'il peut accepter ou non. Dans les deux cas sont choix est transmis (au serveur puis) au client A. Si le client B accepte la connection alors il y aura une connection privée établie entre les deux clients.
Etant en TCP, il n'y a pas besoin d'acquitter les paquets cependant tous les paquets contenant du texte auront un entier (4 octets) indiquant la taille du texte suivit du texte encodé.

Dans le cas de la perte de connection avec le client, le serveur considèrera que le client s'est déconnecté en bonne et due forme même si un transfert de données était en cours. Si il y avait une connection privée impliquant ce client, la connection sera fermée sans prévenir l'autre client.
Dans le cas ou le client pert la connection avec le serveur, le client n'aura rien de spécial à faire.

Quand le serveur reçoit un paquet contenant un message ou une demande de connection privée impliquant un autre utilisateur (identifié par son pseudo) si ledit paquet contient un pseudo erroné ou non existant, un paquet d'erreur sera envoyé en retour. De même que si lors de son authentification au serveur l'utilisateur fournit un pseudo déjà utilisé par quelqu'un d'autre, il recevra un paquet d'erreur. Enfin si le serveur ou le client reçoit un paquet mal formé (ex. on indique un texte de longueur 3 mais on reçoit un texte plus long) l'expéditeur recevra un paquet d'erreur. Si la longueur d'un morceau de texte est supérieure à celle du texte reçu, il est difficile de le déterminer, donc l'erreur pourrait être envoyé à la réception du paquet suivant.

Si un client n'envoie plus de données pendant X minutes, il recevra un ping auquel il devra répondre s'il ne veut pas se faire deconnecter. De même si le serveur ne reçoit pas de réponse à un paquet envoyé (qui necessite une réponse) au bout d'une seconde il deconnectera le client.

3. Paquets ChatOS

ChatOS prend en charge six types de paquets:

Codes = 0 Erreur
- 1 Connection (CON)
- 2 Message général (GMSG)
- 3 Message direct (DMSG)
- 4 Connection privée (CP)
- 5 Demande d'activité (DA)

Le premier octet de chaque paquets représente l'un des codes ci-dessus

A. Connection (CON)

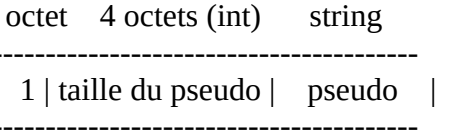


Figure 3-A-1: paquet CON (auth)

Le paquet CON est très simple. C'est le premier paquet envoyé au serveur par le client, il contient (en plus du code) un entier sur 4 octets représentant la longueur en octet du pseudo encodé suivit du pseudo encodé.

Suite à l'envoi de ce paquet, le client attend que le serveur lui renvoie un paquet de validation contenant juste 1 ou un paquet d'erreur dans le cas ou le pseudo est déjà utilisé ou si le paquet est mal formé.

Une fois l'authentification réussi, le client peut maintenant envoyer et recevoir des messages.

B. Message général (GMSG)

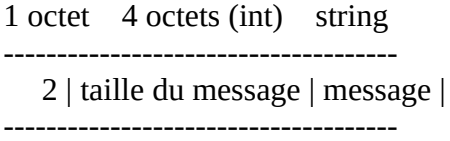


Figure 3-B-1: paquet GMSG (client -> serveur)

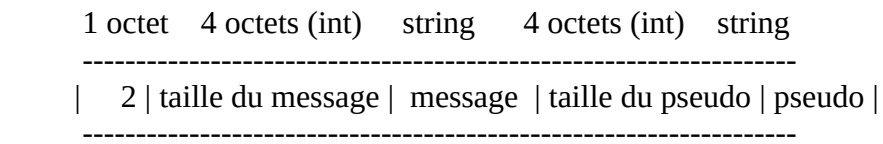


Figure 3-B-2: paquet GMSG (serveur -> client)

Ce paquet permet d'envoyer à tous les clients connecté un message.
Il existe deux types de GMSG, les paquets client et les paquets serveur.
Les paquets client contienent le code 2 suivit de la taille (en octets) du message précédant le message.
Le serveur recevant ce paquet va envoyer à chaque clients connecté un paquet contenant le même message suivit d'un entier (4 octets) représentant la taille du pseudo de l'expéditeur du message qui suivra cet entier.

Le client ne reçoit pas de réponse suite à l'envoie de son message.

C. Message général (DMSG)

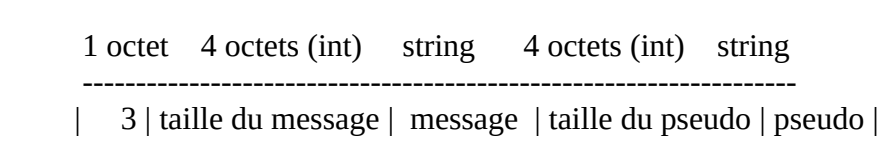
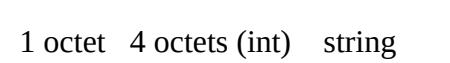


Figure 3-C-1: paquet DMSG

Ce paquet permet à un client d'envoyé un message direct à un autre utilisateur identifié par son pseudonyme.
Le paquet envoyé par le client est composé du code (3) suivit de la taille du message (4 octets) puis du message et enfin un pseudo précédé de la taille en octets du pseudo.
Le pseudonyme se trouvant à la fin du paquet représente - dans le cas d'un paquet envoyé par le client - le destinataire ou l'expéditeur - dans le cas d'un paquet envoyé par le serveur.
Si le client reçoit un paquet d'erreur suite à l'envoie d'un message direct c'est que l'utilisateur demandé n'est pas connecté au serveur.
(A-t-on besoin d'identifier le message pour que l'utilisateur sache quel message est incorrect? Si oui ajouter au message d'erreur pseudo incorrect!)

D. Connection Privée (CP)



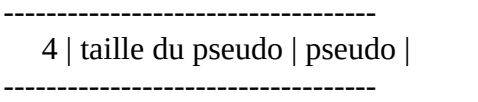


Figure 3-D-1: paquet CP (demande)

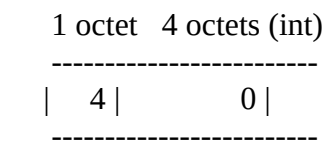


Figure 3-D-2: paquet CP (validation)

Le paquet CP demande indique qu'un client (A) veut se connecter à un autre client (B).
Si ce paquet a été envoyé par le client A, le serveur interpretera le pseudonyme comme celui du destinataire et un paquet de requête de connection privée (CP) sera envoyé au client B contenant cette fois ci le pseudo du client A.
Ensuite le client B aura deux choix: accepter la connection (auquel cas il va envoyer un paquet similaire à celui envoyé par le client A, càd contenant le pseudo du client A) ou refuser la connection (auquel cas il enverra un paquet CP de validation).
Le serveur reçoit donc la réponse du client B et transmet le choix au client A, si la connection est refusée alors il recevra un paquet d'erreur. Si la connection est acceptée alors il recevra un paquet CP de validation.
Suite a l'acceptation de la connection, les clients peuvent communiquer entre eux comme si c'était une connection TCP normale (le serveur relais seulement les paquets sans les modifier).

E. Demande d'activité (DA)

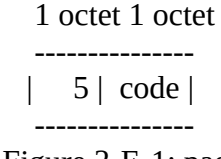


Figure 3-E-1: paquet DA

Le paquet de demande d'activité est envoyé par ke serveur pour vérifié qu'un utilisateur n'est pas AFK (Away From the Keyboard = "loin du clavier").
Les codes possibles sont:
- 1 pour demander au client de signaler sa présence
- 2 pour signaler au serveur qu'on est actif

Le paquet DA est envoyé au client avec le code 1 et le client doit répondre avec un paquet DA contenant le code 2 avant un certain delais pour ne pas se faire deconnecter.

4. Résiliation normale

La fin d'une connection peut se produire de plusieurs façons:
- Si le client ferme la connection avec le serveur.
- Si une connection privée a été mise en place entre deux clients et que l'un des deux se fait déconnecter (naturellement ou par manque d'activité) alors l'autre client sera aussi déconnecté.

Lorsqu'un client se fait déconnecter, il reçoit un paquet d'erreur contenant le code DISCONNECTED (bien que dans le cas d'une fin de connection naturelle ce soit un paquet perdu).

5. Résiliation prématurée

La connection entre le client et le serveur est soumise à une contrainte de temps et de cohérence. Donc si un des cas suivant est rencontré, le client sera déconnecté par l'envoie d'un paquet d'erreur avec le code DISCONNECTED:
- Si le serveur ne peut pas interpréter correctement un paquet reçu.
- Si le client n'envoie plus de données pendant X minutes, il recevra alors un paquet DA auquel il doit répondre avant Y minutes pour ne pas se faire déconnecter. Si il ne répond pas avant le temps imparti il sera déconnecté.
- Si le serveur est en attente de la réponse d'un client (donc si le serveur envoie un paquet necessitant un paquet retour) depuis plus de Z secondes alors le client sera déconnecté.

I. Annexe

Formats ChatOS

Type	octet	4 octets	string	4 octect	string
CON	1	taille du pseudo	pseudo		
GMSG (c->s)	2	taille du message	message		
GMSG (s->c)	2	taille du message	message	taille du pseudo	pseudo
DMSG	3	taille du message	message	taille du pseudo	pseudo
DA	4	taille du pseudo	pseudo		
	octet	octet			
ERROR	5	ErrCode			

Codes d'erreur

Valeur Signification

0 : Erreur générique.
1 : Erreur d'authentification. (AUTH_ERROR)
2 : Erreur de destinataire. (DEST_ERROR)
3 : Connection rejetée. (REJECT)
4 : Déconnection. (DISCONNECTED)

Considérations de sécurité

La protocole ChatOS n'a pas de sécurisation en tant que tel. Cependant l'utilisation d'un pseudonyme permet l'anonymat des utilisateurs envers les autres utilisateurs présent sur le serveur.