

Group 13: c2w protocol specification proposal
draft-ietf-xml2rfc-template-05

Abstract

Ce document presente la specification d'un protocole utile pour une application de type Chat While Watching, abrege c2w par la suite. Une telle application permet a un utilisateur de s'authentifier, d'accéder a une salle principale disposant d'un chat local et permettant de choisir un film, et d'une salle par film disposant également d'un chat local. Le protocol presente est binaire, et peut etre utilise indiffemment sur UDP ou TCP.

Table of Contents

1. Introduction	2
2. Types des messages	2
3. Format des paquets	3
3.1. Entete	3
3.2. Format Type 0 : Acquittement	4
3.3. Format Type 1 : Connexion	4
3.4. Format Type 2 : Quitter Application	5
3.5. Format 3 : Selection du Film	5
3.6. Format 4 : Quitter salon Film	6
3.7. Format Type 5 : Liste des films	6
3.8. Format Type 6 : Liste des utilisateurs	7
3.9. Format Type 7 : Acceptation connexion	8
3.10. Format Type 8 : Refus de connexion	8
3.11. Format Type 9 : Chat	8
4. Hypotheses	9
5. Exemples de scenario	9
5.1. Scenario 1: Visionage d'un film	9
5.2. Scenario 2: Chat et mise a jour des utilisateur	10
5.3. Scenario 3: Refus de connexion et Acquittement	13
6. Reaction a un message reçu	14
6.1. Cote serveur	14
6.2. Cote client	15
7. Conclusion	16
8. Acknowledgements	17
9. References	17
9.1. Normative References	17
9.2. Informative References	17

Authors' Addresses	17
------------------------------	----

1. Introduction

Le protocol presente identifie dix actions possibles. A chacune de ces actions correspond un type de message specifique, numerotes de 0 a 9. En outre, ces messages sont tries en deux categories: les messages emis du client vers le serveur, numerotes de 1 a 4, et ceux emis du serveur vers le client, numerotes de 5 a 8. Le type 0 correspond a l'acquittement. Le type 9 correspond aux messages de chat, seul type pouvant etre emis et reçu sans distinction par un serveur ou un client.

Les dix types de messages possibles sont d'abord presentes ainsi que les formats des parties data pour ces dix types de messages. Ensuite, sont detaillees pour le serveur et pour le client separement les actions a effectuer (consequences) en fonction du message reçu (cause), formant ainsi un modele cause-consequence exhaustif.

2. Types des messages

On identifie pour le fonctionnement complet de l'application un total de 10 messages differents. Les dix types possibles sont donnees dans le tableau ci-dessous:

Type	Valeur Binaire	Action	Emetteur
0	0000	Acquittement	Client et Serveur
1	0001	Envoi du Pseudo	Client
2	0010	Quitter application	Client
3	0011	Choix d'un film	Client
4	0100	Quitter salle film	Client
5	0101	Envoi liste films	Serveur
6	0110	Envoi liste users	Serveur
7	0111	Acceptation connexion	Serveur
8	1000	Refus connexion	Serveur
9	1001	Message Chat	Client et Serveur

Figure 1

3. Format des paquets

Chaque message d'un type n est constitue de deux parties: l'entete et les donnees (le contenu) du message. Nous distiguons l'entete et le decrivons en premier car celui-ci est commun a tous les types de messages. Ainsi chaque message DOIT etre constitue de l'entete suivi des donnees. Dans la description du format de chaque type, nous nous attardons uniquement sur le format de la la partie donnee. Mais evidemment, chaque message commence par l'entete. Simplement, celui-ci n'est pas rappelle a chaque fois, puisque son format est toujours le meme.

3.1. Entete

L'entete est place au debut de tous les messages echanges. Il est constitue de la longueur totale du message(entete et donnees), du numero de sequence et du type du message.

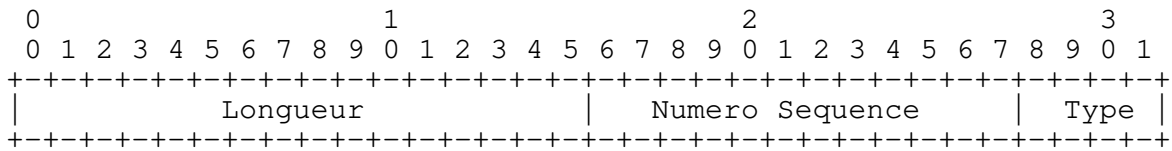


Figure 2

Longueur (16 bits)

Longueur totale du message (entete + donnees). Entier non signe sur 2 octets. Il vaut la longueur totale du paquet en octets.

Numero Sequence (12 bits)

Numerotation des paquets pour controler les envois et receptions, gerer l'ordre et les pertes de paquets. Chaque entite, serveur ou client, a son propre compteur. Ce compteur est initialise a 0 a chaque connexion et on l'incrimente a chaque envoi de paquet.

Type (4 bits)

Type du message, compris entre 0 et 9 code sur 4 bits. Sa valeur est fixee comme indique dans la section 3.

3.2. Format Type 0 : Acquittement

Lorsqu'un message est envoye, l'emetteur attend un acquittement (message vide avec le meme identifiant) en l'absence duquel il renvoie le message au bout de 1 seconde. Le message d'acquittement est uniquement constitue de l'en tete. En effet, c'est le numero de sequence qui sert d'identifiant du message original et donc d'acquittement.

3.3. Format Type 1 : Connexion

Il s'agit ici d'envoyer au serveur le pseudo choisi par l'utilisateur lors de sa connexion pour qu'il puisse etre identifiable dans le Chat. Si le pseudo est deja utilise la connexion est refusee.

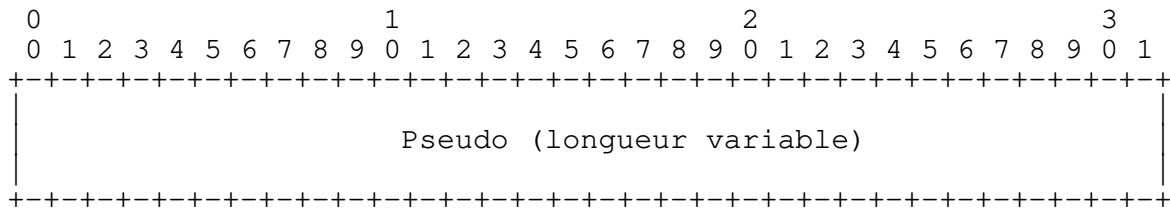


Figure 3

Pseudonyme (256 octets maximum)

Permet d'identifier un utilisateur, notamment lors de l'utilisation du Chat. Sa taille est variable.

3.4. Format Type 2 : Quitter Application

Le client envoie un paquet au serveur pour lui indiquer que l'utilisateur en question se deconnecte. Le serveur met alors a jour la liste des utilisateurs et l'envoie a tous les clients.

3.5. Format 3 : Selection du Film

Le client envoie au serveur le nom du film selectionne. Le serveur met a jour le statut de l'utilisateur qui passe de 'A' a 'M'. Si acquittement, le client rentre dans la salle du film qu'il a choisi.

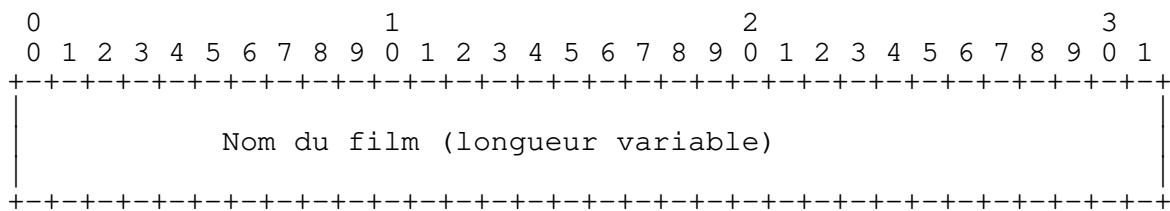


Figure 4

Nom du film

Chaine de caractere de longueur variable.

3.6. Format 4 : Quitter salon Film

Le client envoie uniquement l'entete au serveur qui met a jour le statut de l'utilisateur, il passe de 'M' a 'A'.

3.7. Format Type 5 : Liste des films

Le serveur envoie aux clients la liste de tous les films disponibles sur l'application avec pour chacun, les informations necessaires pour y avoir acces.

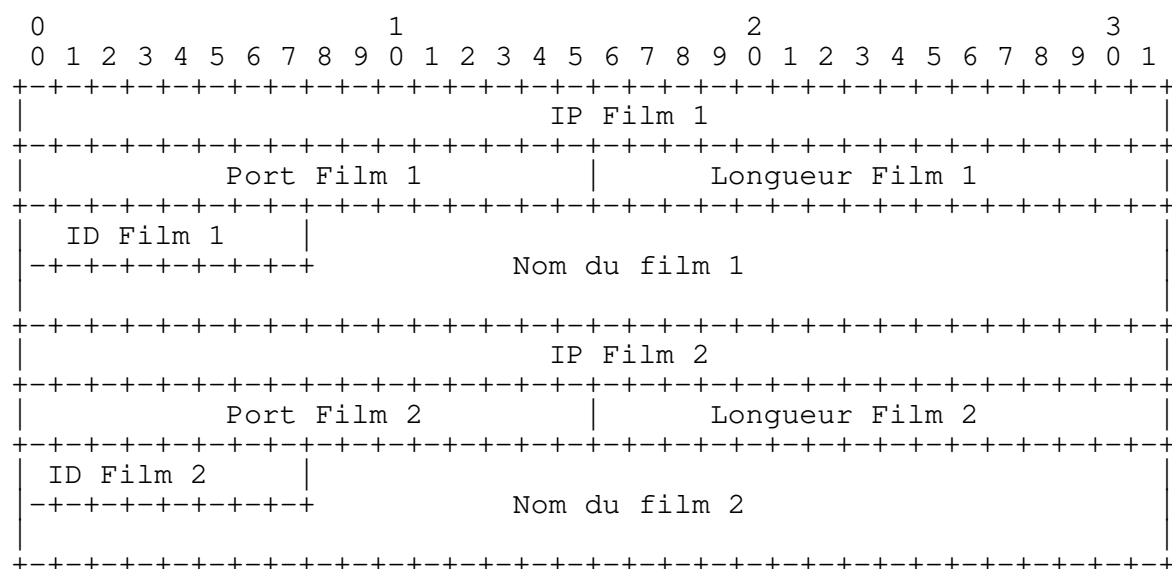


Figure 5

IP Film k (32 bits)

Adresse du serveur hebergeant le film k sur 4 octets.

Port Film k (16 bits)

Port du serveur hebergeant le film k. Entier sur 2 octets.

Longueur Film k (16 bits)

Longueur totale en octet des donnees concernant le film k (IP, Port, Longueur et Nom). Entier non signe sur 2 octets.

ID Film k (8 bits)

ID du film k. Code sur 1 octet. 0b00000000 n'est pas utilise (reservee pour la salle principale).

Nom du film

Chaine de caracteres de longueur variable.

3.8. Format Type 6 : Liste des utilisateurs

Le serveur envoie aux clients la liste de tous les utilisateurs connectes sur l'application ainsi que leur statut.

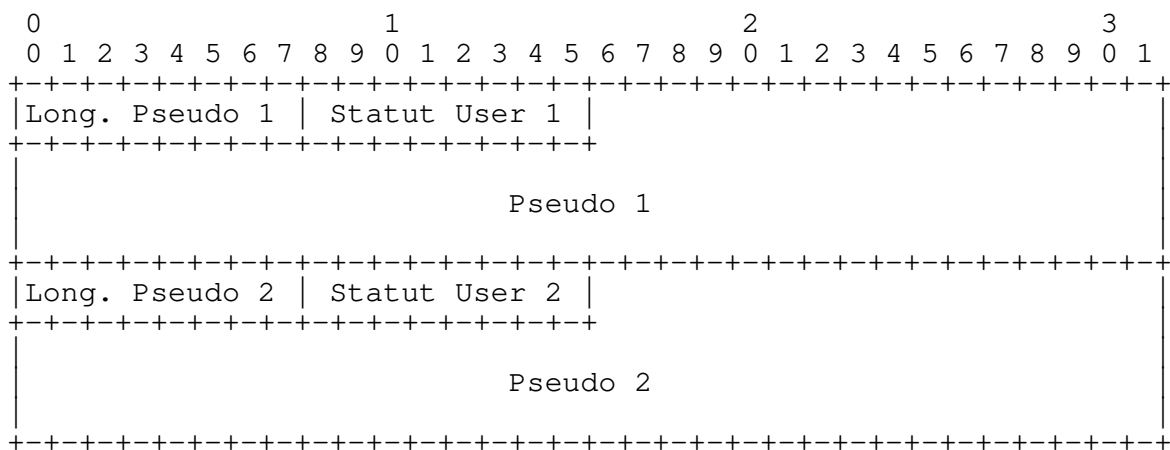


Figure 6

Long. Pseudo k (8 bits)

Longueur du Pseudo k sur 1 octet.

Statut User k (8 bits)

Statut de l'utilisateur k sur 1 octet. Si la valeur est 0, alors l'utilisateur est dans la salle principale, sinon, la valeur indique l>ID de la salle du film.

Pseudo k

Chaine de caractere de longueur variable.

Chaine de caracteres de taille non capee contenant le texte du message.

4. Hypotheses

Hypothese : Les champs contenant du texte sont encodes en UTF-8.

Hypothese : On fixe le nombre maximal de reemissions a 7 et l'intervalle de reemission a 1 seconde.

Hypothese : Les octets sont organises dans l'ordre standard des communications reseau (big endian).

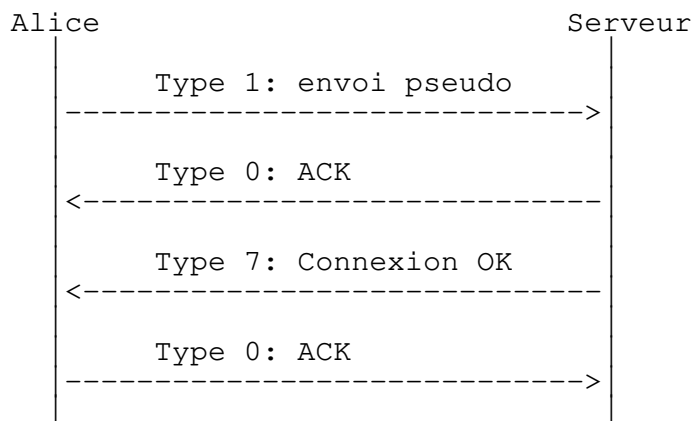
Hypothese : Lors de la connexion d'un nouvel utilisateur, la liste des films est toujours envoyee avant la liste des utilisateurs.

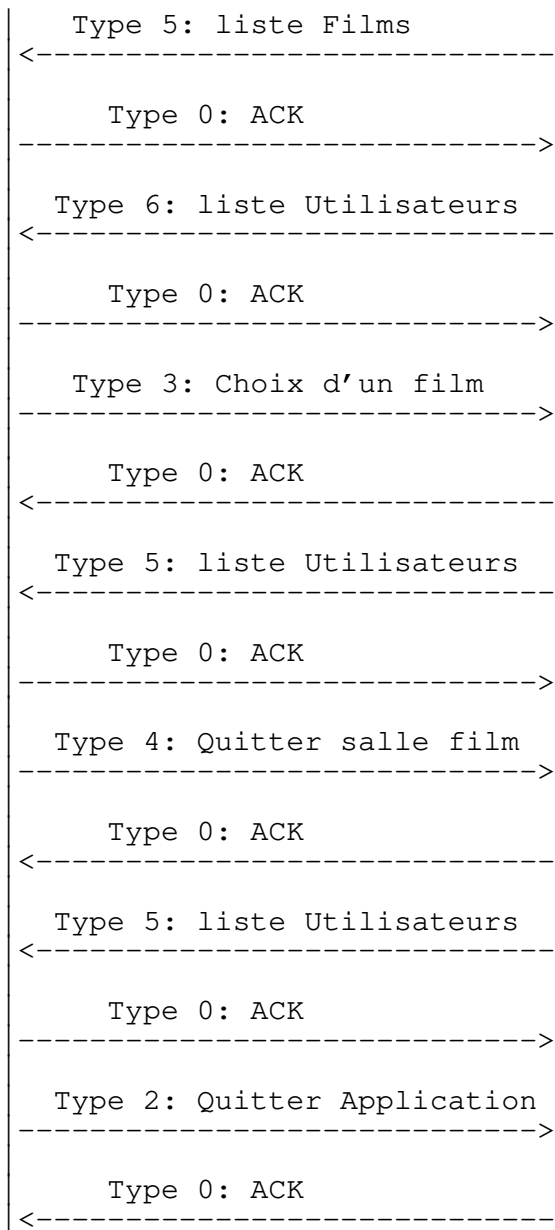
5. Exemples de scenario

5.1. Scenario 1: Visionnage d'un film

Dans ce premier exemple simple, Alice est la seule utilisatrice de l'application. Alice se connecte a l'application. Elle choisit un film et rentre dans la salle qui lui correspond. A la fin du film, elle se deconnecte. A chaque changement de statut (entree ou sortie d'une salle), la liste des Users est mise a jour donc renvoyee a tous les utilisateurs, c'est a dire dans notre cas uniquement a Alice. (Si Bob etait connecte en plus d'Alice, la liste des Users serait alors envoyee a Bob ET a Alice) Dans ce premier scenario, aucun probleme ne survient, ie chaque message est suivi de son acquittement sans soucis.

La figure ci-dessous illustre chronologiquement l'ensemble des echanges. Pour chaque message est indique le type et le contenu.



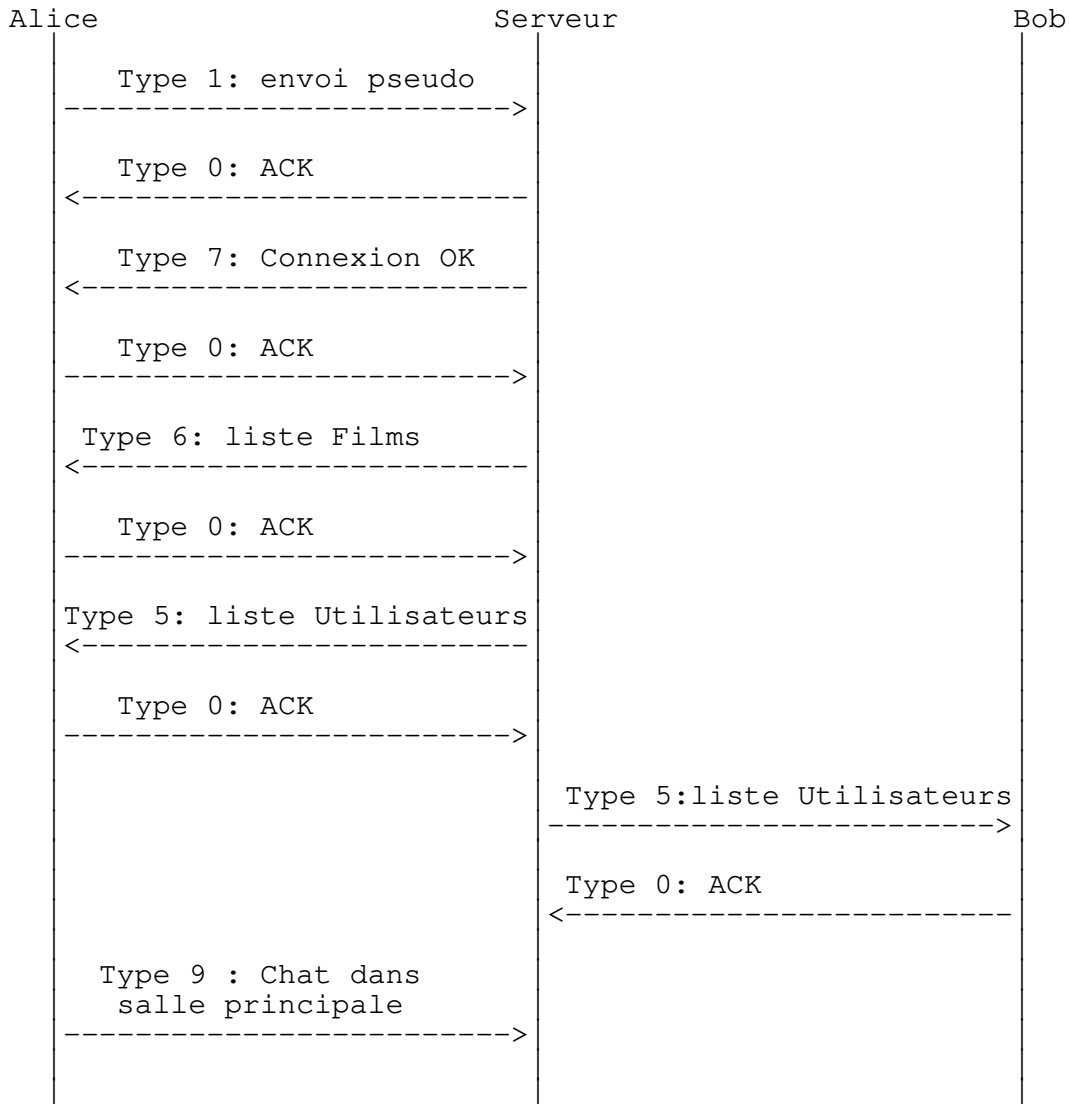


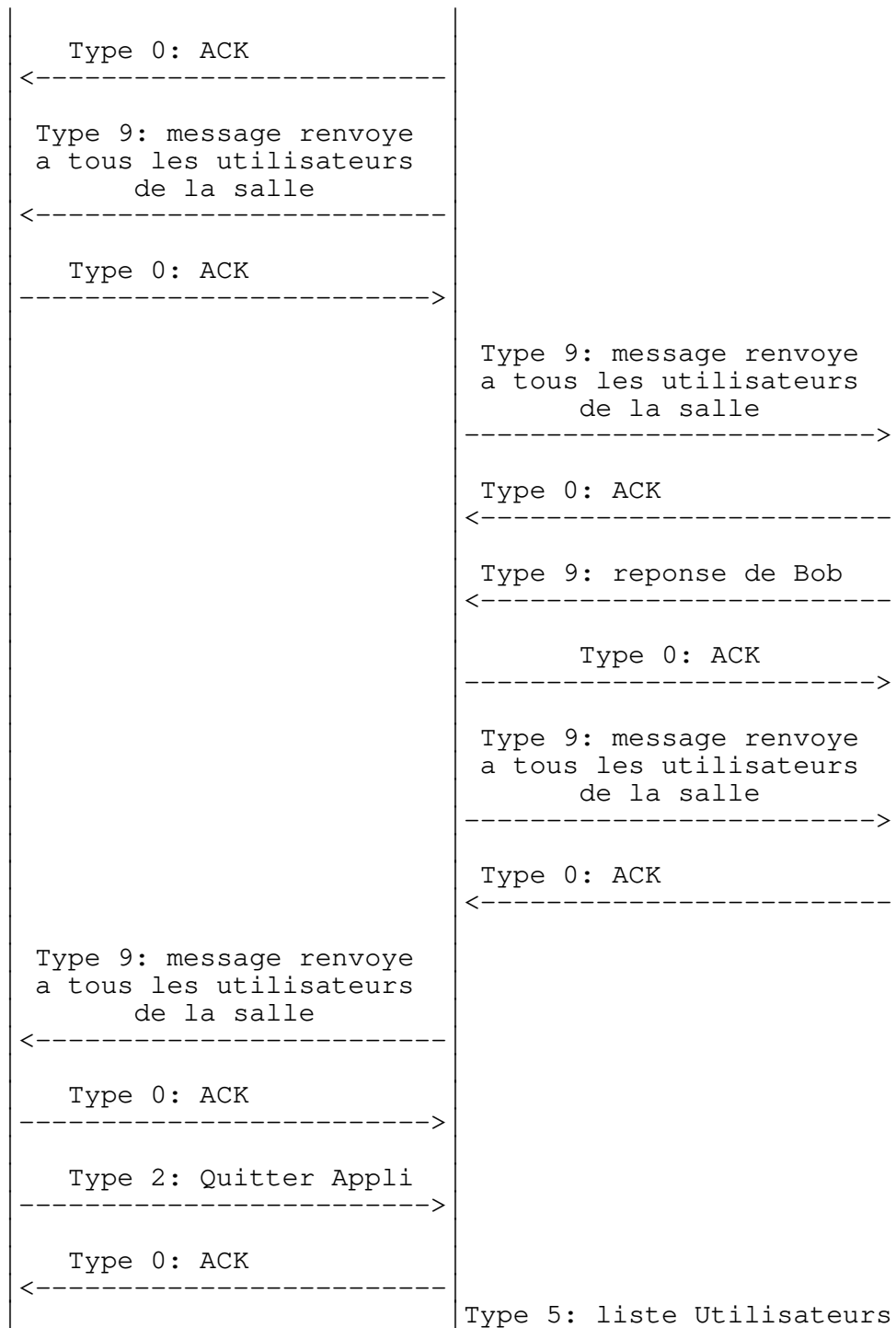
5.2. Scenario 2: Chat et mise a jour des utilisateur

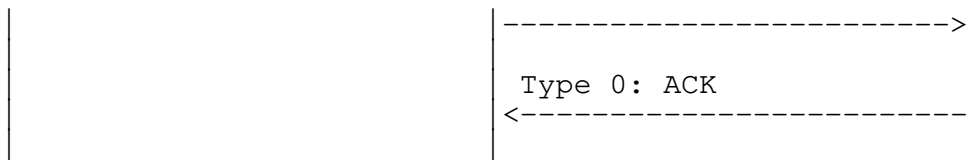
Alice se connecte a l'application. A ce moment, Bob est deja connecte. Il recoit donc l'information de la connexion d'Alice, c'est a dire la liste des utilisateurs mise a jour. Ils echangent

des messages chat dans la salle principale. Bob se deconnecte. Comme pour le premier scenario, aucun probleme ne survient, ie chaque message est suivi de son acquittement sans soucis.

La figure ci-dessous illustre chronologiquement l'ensemble des echanges. Pour chaque message est indique le type et le contenu. Les acquittements ne sont pas indiques sur cette figure. Gardez a l'esprit que chaque message echange entre deux entites (parmi Alice, Bob, ou le serveur) est suivi d'un message d'acquittement (Type 0) entre les deux memes entites.



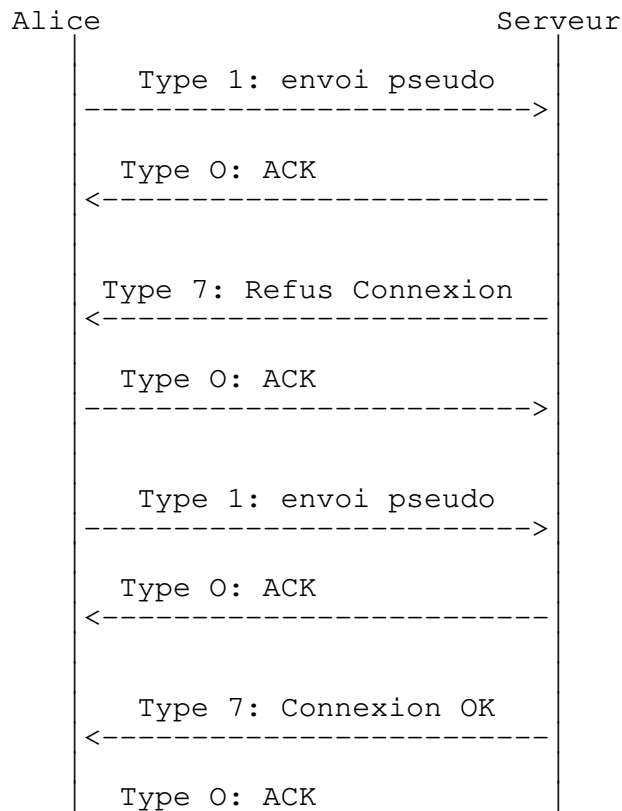


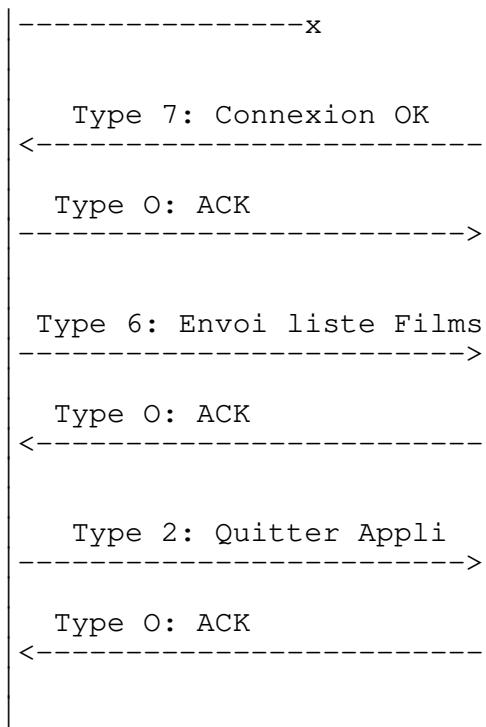


5.3. Scenario 3: Refus de connexion et Acquittement

Alice se connecte a l'application. Le pseudo qu'elle choisi est deja utilise donc sa connexion est refusee. Elle en choisi un autre, non utilise. Sa connexion est acceptee mais le serveur ne recoit pas l'acquittement du message d'acceptation de connexion. Le serveur renvoi alors le message d'acpetation de connexion apres un delai d'une seconde. Une fois connectee, Alice se deconnecte.

La figure ci-dessous illustre chronologiquement l'ensemble des echanges. Pour chaque message est indique le type et le contenu.





6. Reaction a un message reçu

6.1. Cote serveur

Liste des reactions du serveur a la reception d'un message emis par un client.

Type du message	Reaction
1: Envoi Pseudo	<ul style="list-style-type: none"> - Si pseudo deja pris, refus de connexion: Envoyer Type 8 - Si pseudo OK, acceptation de connexion: Envoyer Type 7 - Utilisateur place dans la salle principale - Stockage Pseudo, IP et Port de l'utilisateur - Mettre a jour la liste des utilisateurs - Envoyer la liste des films a l'utilisateur qui vient de se connecter: Envoyer Type 6 - Envoyer la liste des utilisateurs a tous les utilisateurs: Envoyer type 5
2: Deconnexion	<ul style="list-style-type: none"> - Supprimer le donnees pour l'utilisateur concerne - Mettre a jour la liste des utilisateurs - Envoyer la liste des utilisateurs a tous les utilisateurs: Envoyer type 5
Type 3: Choix Film	<ul style="list-style-type: none"> - Utilisateur place dans la salle concernee - Mettre a jour la liste des utilisateurs - Envoyer la liste des utilisateurs a tous les utilisateurs: Envoyer type 5
Type 4: Quitter salle film	<ul style="list-style-type: none"> - Utilisateur place dans la salle principale - Mettre a jour la liste des utilisateurs - Envoyer la liste des utilisateurs a tous les utilisateurs: Envoyer type 5
Type 9: Message Chat	<ul style="list-style-type: none"> - Renvoi le message a tous les utilisateurs (emetteur compris) presents dans la salle de l'emetteur
Type 0: Acquittement	<ul style="list-style-type: none"> - stopper les reemissions pour le paquet acquitte - mise a jour de la liste des paquets a acquitter

Figure 8

6.2. Cote client

Liste des reactions du client a la reception d'un message emis par le serveur.

Type du message	Reaction
Type 5: Envoi liste films	<ul style="list-style-type: none"> - Stockage liste film - Affichage de cette liste dans l'interface graphique - Utilisateur place dans la salle principale
Type 6: Envoi liste users	<ul style="list-style-type: none"> - Mise a jour de l'affichage des utilisateurs en fonction de la salle dans laquelle le client est present.
Type 4: Quitter salle film	<ul style="list-style-type: none"> - Utilisateur place dans la salle principale - Mettre a jour la liste des utilisateurs - Envoyer la liste des utilisateurs a tous les utilisateurs: Envoyer type 5
Type 7: Connexion OK	<ul style="list-style-type: none"> - idle
Type 8: Refus Connexion	<ul style="list-style-type: none"> - affichage message d'erreur
Type 9: Message Chat	<ul style="list-style-type: none"> - s'il est l'emetteur, ne fait rien - sinon, affichage du message dans l'interface graphique du chat
Type 0: Acquittement	<ul style="list-style-type: none"> - stopper les reemissions pour le paquet acquitte - mise a jour de la liste des paquets a acquitter

Figure 9

7. Conclusion

Le protocole propose est base sur le principe suivant: puisque l'application consideree est simple, le nombre d'actions executables total est faible (en l'occurrence 10). En utilisant les types de messages, on identifie tres facilement la fonction d'un message reçu. Il ne reste alors plus qu'a definir les reactions a un type donne de message reçu. Ces deux elements permettent de definir facilement un protocol fiable, exhaustif et, nous l'esperons, clair.

8. Acknowledgements

This template was derived from an initial version written by Pekka Savola and contributed by him to the xml2rfc project.

This document is part of a plan to make xml2rfc indispensable [DOMINATION].

9. References

9.1. Normative References

[min_ref] authSurName, authInitials., "Minimal Reference", 2006.

9.2. Informative References

[DOMINATION]
Mad Dominators, Inc., "Ultimate Plan for Taking Over the World", 1984, <<http://www.example.com/dominator.html>>.

[IDSample]
Davies, E., "Internet Draft Sample", 2006,
<<http://tools.ietf.org/tools/templates/draft-davies-template-bare.txt>>.

Authors' Addresses

Olivier Famechon
IMT Atlantique
Brest
France

Email: olivier.famechon@imt-atlantique.net

Luc Vedrenne
IMT Atlantique
Brest
France

Email: luc.vedrenne@imt-atlantique.net