

Group 25: c2w protocol specification proposal
spec-r328-s17-g25

Abstract

L'application Chat While Watching (c2w) permet a un utilisateur de discuter avec d'autres utilisateurs regardant le meme film que lui. Le protocole de l'application permet la communication entre un client (l'utilisateur) et un serveur. Ce service de chat permet la gestion de plusieurs salons de discussion (la salle principale et les salles de film) et l'envoi ou reception de messages.

Ce document presente la specification de la version bianire du protocole de l'application c2w.

Table of Contents

1. Introduction	1
2. Format general des messages	2
3. Login	3
4. Main Room	4
5. Movie Room	7
6. Chat	8
7. Fiabilite	9
8. Exemple de scenario	9
9. Conclusion	22
Authors' Addresses	22

1. Introduction

L'application c2w permet a un utilisateur de discuter avec des personnes regardant le meme film que lui. Le protocole doit permettre d'envoyer et de recevoir des messages entre les differents utilisateurs presents dans la meme salle. Il doit aussi assurer la connexion et la deconnexion d'un utilisateur. Il permet a un utilisateur de choisir un film parmi une liste connue par le serveur.

Chaque client doit connaitre le nom (ou l'adresse IP) ainsi que le numero de port du serveur auquel il souhaite se connecter. Il doit

egalement choisir un identifiant, compose au maximum de seize caracteres.

Ce document presente l'ensemble des specifications du protocole, comme le format d'un message ou les interactions clients/serveur dans la salle principale, ainsi qu'un exemple de scenario afin d'illustrer le protocole.

2. Format general des messages

Tous les messages ont le meme format. Le format est decrit ci-dessous :

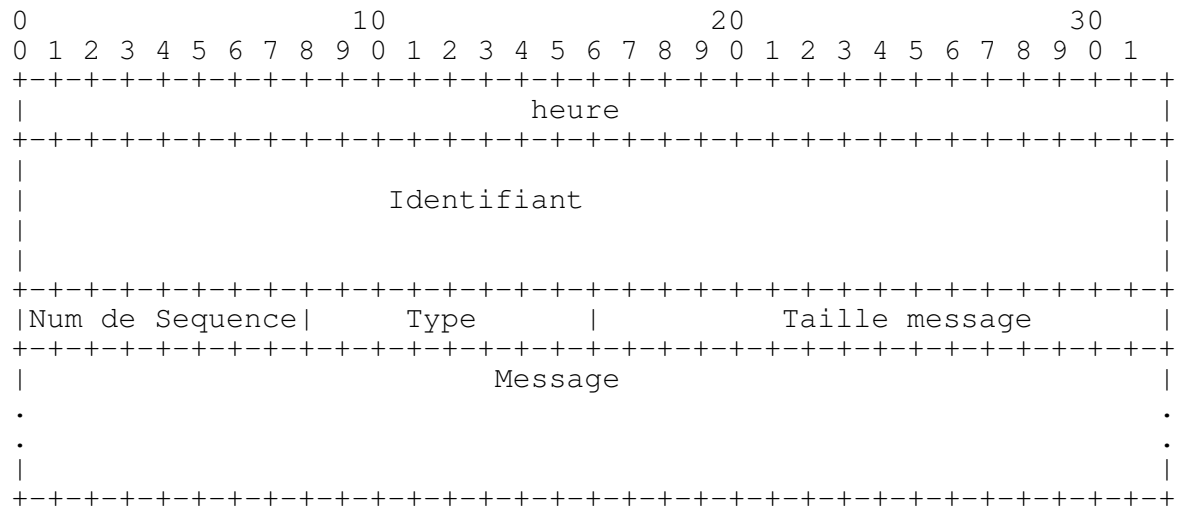


Figure 1

Champ Heure (4 octets)

Ce champ contient le nombre de seconde depuis le premier janvier 1970. Pour les reponses, c'est-a-dire les messages du serveur vers le client, le champ Heure doit etre le meme que le champ heure correspondant a la requete du client.

Champ Identifiant (16 octets)

Pour un message du client vers le serveur, ce champ contient l'identifiant choisi par l'utilisateur. Sa taille maximale est de 16 caracteres, codes en ASCII. Si l'identifiant comprend moins de 16 caracteres, les bits non utilises prennent automatiquement la valeur de zero

derriere les bits utiles. Pour un message du serveur vers le client, ce champ contient l'adresse IP du serveur.

Champ Numero de Sequence (1 octet)

Ce champ permet d'assurer la fiabilite du protocole. Le principe de fonctionnement est decrit dans la partie 7, intitulee Fiabilite.

Champ Type (1 octet)

Ce champ precise le type de message et permet d'avoir tout les messages au meme format.

Requete de connexion 00000001

Succes de connexion 00000010

Echec de connexion 00000100

Liste utilisateurs 00001000

Liste films 00010000

Requete_Chat 00100000

Reponse_Chat 01000000

Deconnexion 10000000

Acquittement 00000011

Choix film 00000101

Champ Taille message (4 octets)

Ce champ indique la taille du champ Message en bits.

Champ Message (taille variable)

Le message transmis est en code ASCII.

3. Login

L'utilisateur doit tout d'abord se connecter a la plateforme. Pour cela, il envoie une requete de connexion au serveur contenant son identifiant, l'adresse IP et le numero de port du serveur auquel il souhaite se connecter.

Le serveur doit conserver les informations suivantes dans une variable : adresse IP, numero de port, identifiant et localisation de chaque utilisateur. La localisation du client se traduit par l'adresse IP de la Main Room ou l'adresse IP correspondant au film visionne par le client.

Le serveur envoie une confirmation de login au client ou un message d'echec en cas d'erreur de connexion. La connexion peut echouer pour plusieurs raisons : l'identifiant choisi est trop long ou il a deja ete choisi par un autre utilisateur ; l'adresse IP ou le port entre est invalide.

Pour chaque situation, le champ Type prend les valeurs suivantes :

Requete de connexion 00000001

Succes de connexion 00000010

Echec de connexion 00000100

Le contenu du champ Message doit preciser si la connexion est etablie et dans le cas contraire la nature de l'erreur.

Une fois la connexion etablie, une nouvelle fenetre Main Room s'ouvre.

4. Main Room

Suite a la connexion reussie, le client entre dans la Main Room ou il aura acces a la liste de tous les films disponibles, la liste de tous les utilisateurs avec leurs status (Available (A) ou Movie Room (M)) et un espace de chat. Available correspond a un utilisateur dans la Main Room et Movie Room a un utilisateur visionnant un film.

Des que le client est connecte a la Main Room, le serveur doit lui envoyer la liste de tous les films disponibles. Pour cela, il envoie un message du type Liste films.

type Liste films 00010000

Le champ Message sera constitue des adresses IP de tous les films disponibles. C'est une succession de 4 octets, qui correspond a l'adresse IP de chaque film

Le client envoie ensuite un acquittement pour notifier qu'il a bien reçu la liste des films.

type Acquittement 00000011

Le meme scenario est utilise pour la liste des utilisateurs, seul le type du message est modifie.

type Liste utilisateurs 00001000

Le champ Message sera constituee d'une succession de noms d'utilisateurs avec leur statut ("A" ou "M")

Par exemple, le format du message sera le suivant :

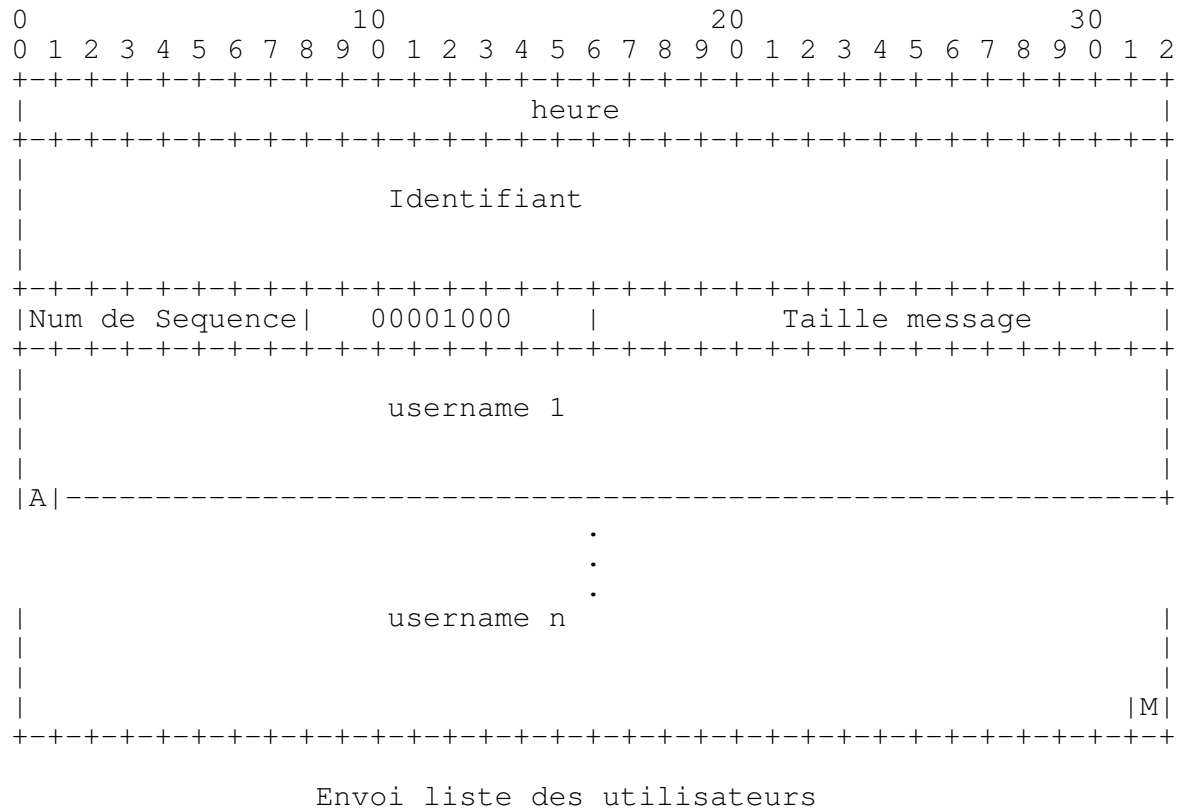


Figure 2

Une fois dans la Main Room, le client peut envoyer un message aux autres utilisateurs presents dans la Main Room. Le message sera tout d'abord envoye au serveur qui le redistribuera aux autres utilisateurs. Le protocole specifique au chat sera decrit dans la partie 6. La liste des utilisateurs et de leur localisation

conservee par le serveur doit etre mise a jour des qu'un utilisateur se connecte ou se deconnecte, c'est-a-dire des que le serveur envoie un message du type Succes de connexion ou recoit un message du type Deconnexion.

type Succes de connexion 00000010

type Deconnexion 10000000

Le client peut rejoindre une Movie Room en specifiant l'IP Address du film correspondant. Le serveur possede une liste des adresses IP et port de chaque film. Le client envoie un message du type Choix film au serveur. Apres avoir reçu l'acquittement, il entre dans la Movie Room correspondante.

Choix film 00000101

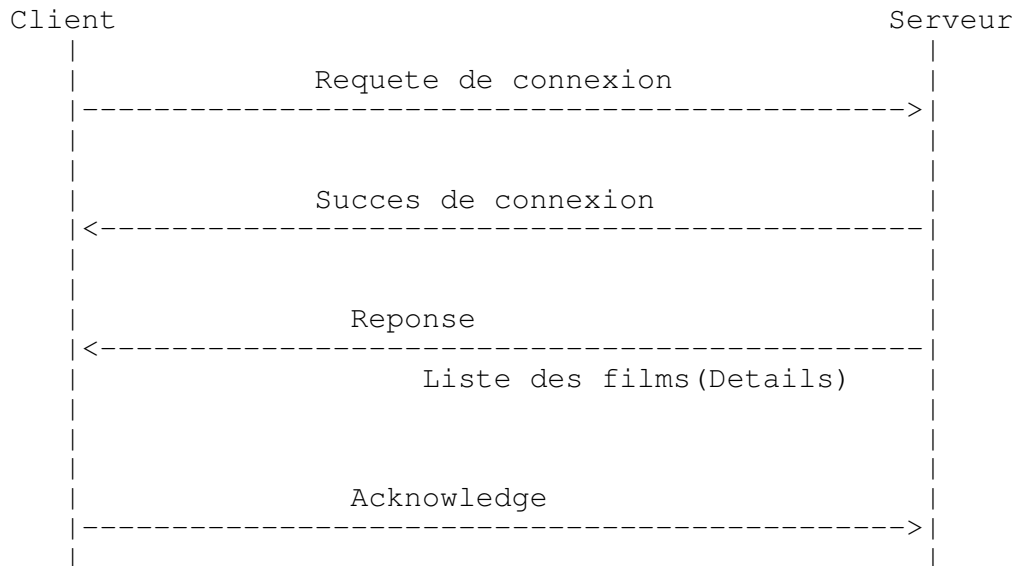
Le client pourra quitter l'application en envoyant un message de Deconnexion.

Deconnexion 10000000

L'ensemble des interactions qui peuvent avoir lieu dans la Main Room sont donc les suivantes :

- Envoi de la liste des utilisateurs (serveur)
- Envoi de la liste des films (serveur)
- Acquittement (client)
- Envoi message via le chat (client)
- Transfert message a tous les utilisateurs du meme salon (serveur)
- Joindre une movie room (client)
- Se deconnecter

Par exemple, on peut illustrer la demande de rejoindre un film par le diagramme de sequence suivant :



5. Movie Room

L'utilisateur a la possibilité de discuter via une zone de chat avec les utilisateurs présents dans la même salle.

Lorsque l'utilisateur rentre dans la Movie Room (après la requête Choix Film décrite dans la partie) précédente, l'information concernant sa localisation doit être mise à jour. Elle prend la valeur de l'adresse IP de la Movie Room correspondante. La liste des utilisateurs de la Movie Room correspondante est mise à jour en ajoutant ce nouvel utilisateur. Cette liste doit être mise à jour par le serveur d'un utilisateur entre ou sort de la Movie Room c'est-à-dire dès que le serveur reçoit une requête du type Choix film ou du type Deconnexion. À chaque mise à jour de la liste, la nouvelle liste est envoyée à tous les utilisateurs.

Choix film 00000101

Deconnexion 10000000

Le serveur doit envoyer la liste des utilisateurs présents dans la même salle. Il envoie donc un message du type Liste utilisateurs.

Type Liste utilisateur 00001000

Lorsque l'utilisateur souhaite quitter sa Movie Room, il envoie une requete de deconnexion au serveur. Le message envoye est du type Deconnexion.

Type Deconnexion 00000101

Apres avoir reçu l'acquittement de sa requete, l'utilisateur est redirige vers la fenetre Main Room, dont le fonctionnement a ete decrit dans la partie precedente.

L'utilisateur a acces a une zone de chat ou il peut recevoir et envoyer des messages aux utilisateurs de la meme Movie Room. Le protocole specifique au chat sera decrit dans la partie suivante, intitulee Chat.

6. Chat

Le chat de la Main Room et des Movies Room fonctionnent de la meme maniere. La seule difference est la liste d'utilisateurs. Dans la Movie Room, seuls les utilisateurs presents dans cette salle peuvent communiquer entre eux.

Afin de discuter avec les gens presents dans la salle, l'utilisateur envoie un message de type Requete_chat au serveur. Le serveur recoit le message et envoie un acquittement a l'utilisateur et un message de type response_chat a tout les autres utilisateurs presents dans la salle. Le serveur doit stocker la liste de tous les utilisateurs connectes ainsi que leur localisation. Connaissant la localisation de l'emetteur du message, il doit envoyer le message aux utilisateurs ayant la meme localisation

Les messages de type response_chat ne pouvant etre envoyes qu'a une seule personne, le serveur envoie donc autant de fois le message qu'il y a d'utilisateurs.

A l'ouverture du message par un utilisateur, un acquittement est envoye au serveur pour signaler la bonne reception du message.

Requete_chat 00100000

Response_chat 01000000

Acquittement 00000011

7. Fiabilite

Lorsqu'un client utilise TCP, la fiabilite est assuree grace au champ Taille, indiquant la taille du message et permettant de savoir lorsque le message en entier a ete transmis.

Lorsque le client utilise UDP, la fiabilite est assuree grace au champ Numero de Sequence. Il permet de detecter les pertes de message et d'eviter les redondances.

Principe de Fonctionnement :

L'emetteur envoie un message, ou le champ Numero de sequence vaut X. Par default, la valeur du Champ Numero de sequence du premier message envoye vaut 0, et a chaque nouveau message, cette valeur est incrementee.

Lorsque le recepteur recoit le message, il envoie un message du type Acquittement avec un Numero de Sequence valant X.

Type Acquittement 00000011

Si l'emetteur ne recoit rien dans un delais de 2 secondes, il renvoie le message de numero de sequence X.

S'il recoit l'acquittement, il envoie le message suivant en incrementant la valeur du champ Numero de sequence.

Si le recepteur recoit le message X en double, par exemple en cas de perte d'un acquittement, il ne considere pas la redondance et envoie un nouvel acquittement.

8. Exemple de scenario

Dans cette partie, nous allons donner un exemple simple de cas d'utilisation, de la connexion et a la deconnexion d'un utilisateur. Cet utilisateur choisira un film, et recevra et enverra des messages dans sa Movie Room.

Dans un premier temps, l'utilisateur se connecte. Pour cela, il envoie un message de type " Requete de connexion" (00000001)

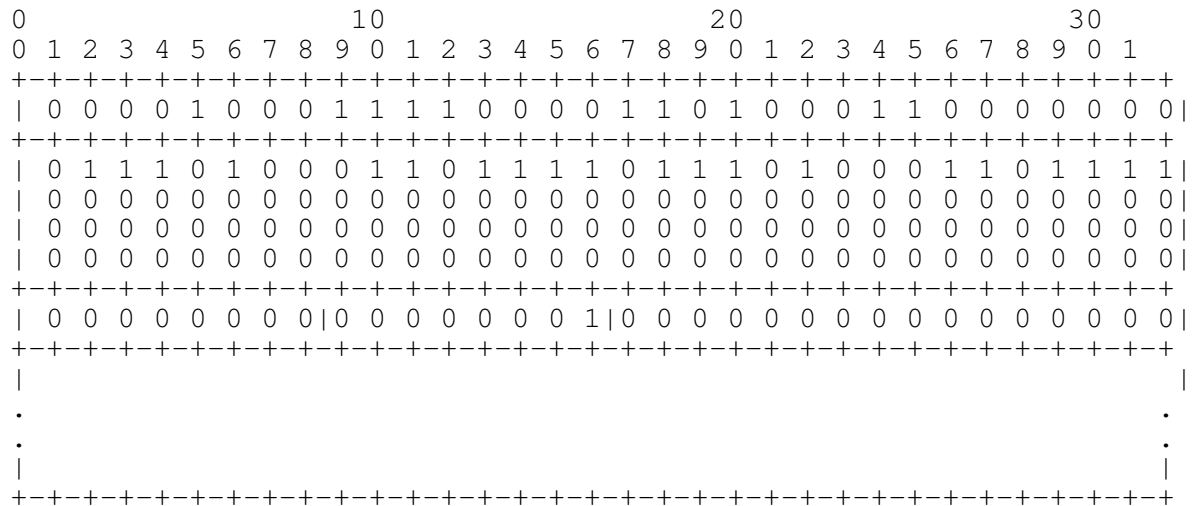


Figure 3

heure = 1000111100001101000110000000

identifiant = toto = 01110100 01101111 01110100 01101111

@IP Serveur = 66.249.64.201 = 1000010.11111001.1000000.11001001

Numero de sequence = 0

Type = 00000001

Taille message = 0

Message = vide

La requete de connexion est envoye au serveur. Aucun acquittement est recue par le client apres deux secondes. Le client renvoie le meme message au serveur.

La demande de connexion est prise en compte par le serveur. Deux cas sont possibles. Premier Cas, Echec de Connexion :

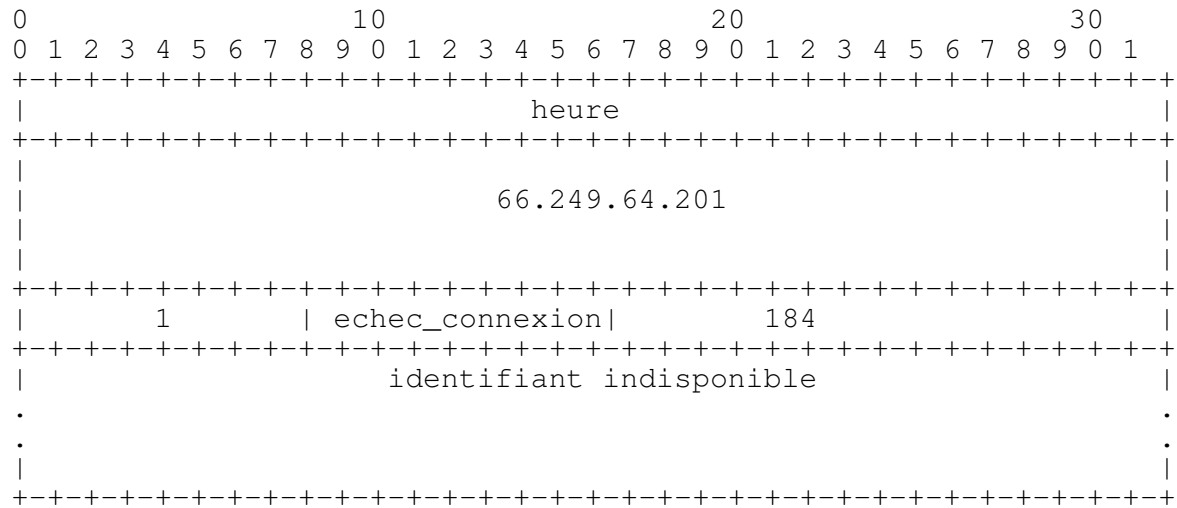


Figure 4

heure = 1000111100001101000110000010

66.249.64.201 = 1000010.11111001.1000000.11001001

numero de sequence = 1

type = 00000100

Taille message = 10111000

message = 01101001 01100100 01100101 01101110 01110100 01101001
 01100110 01101001 01100001 01101110 01110100 00100000 01101001
 01101110 01100100 01101001 01110011 01110000 01101111 01101110
 01101001 01100010 01101100 01100101

Second Cas, Succes de Connexion:

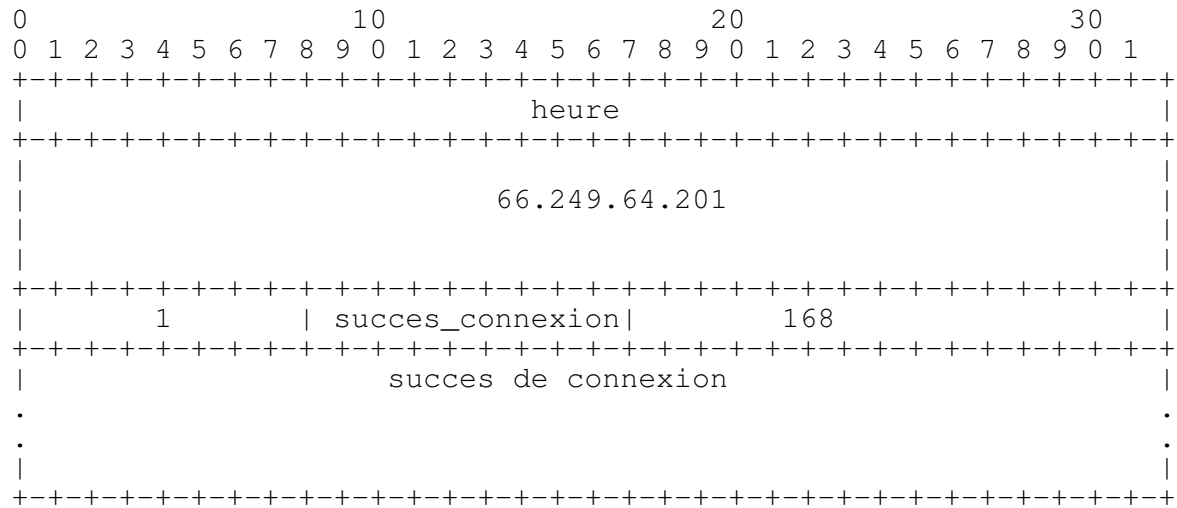


Figure 5

heure = 1000111100001101000110000010

66.249.64.201 = 1000010.11111001.1000000.11001001

numero de sequence = 1

type = 00000010

Taille message = 10101000

message = 01110011 01110101 01100011 01100011 11101000 01110011
00100000 01100100 01100101 00100000 01100011 01101111 01101110
01101110 01100101 01111000 01101001 01101111 01101110 00001101
00001010

L'utilisateur est maintenant connecte. Le serveur le rajoute a la liste des utilisateurs connectes Le serveur lui envoie la liste des utilisateurs ainsi que la liste des films. Il y a trois autres utilisateurs present et il y a 2 films proposes.

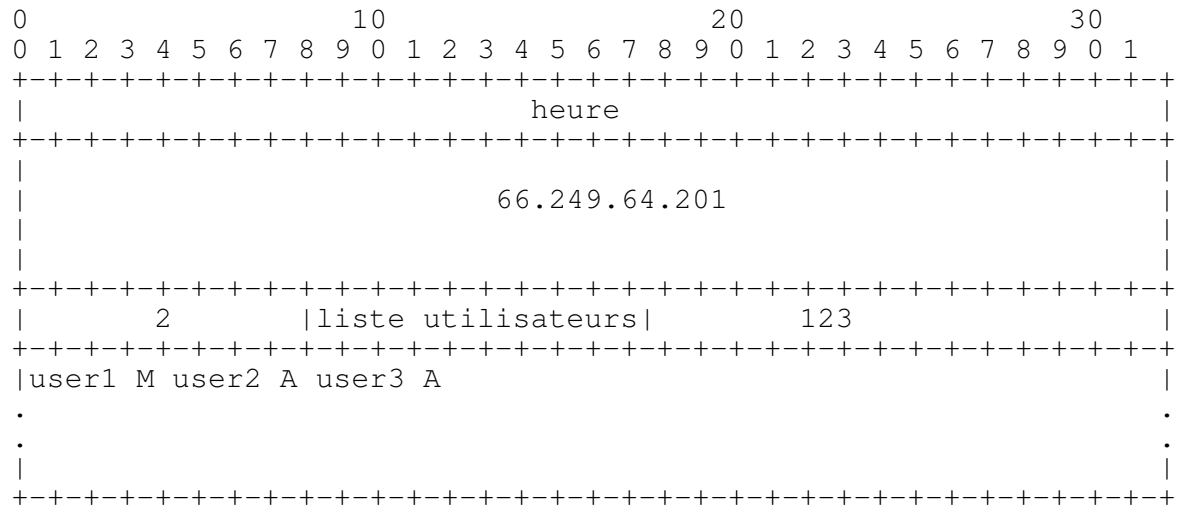


Figure 6

heure = 1000111100001101000110000110

66.249.64.201 = 1000010.11111001.1000000.11001001

numero de sequence = 10

type = 00001000

Taille message = 1111011

message = 01110101 01110011 01100101 01110010 00110001 00100000
01110101 01110011 01100101 01110010 00110010 00100000 01110101
01110011 01100101 01110010 00110011

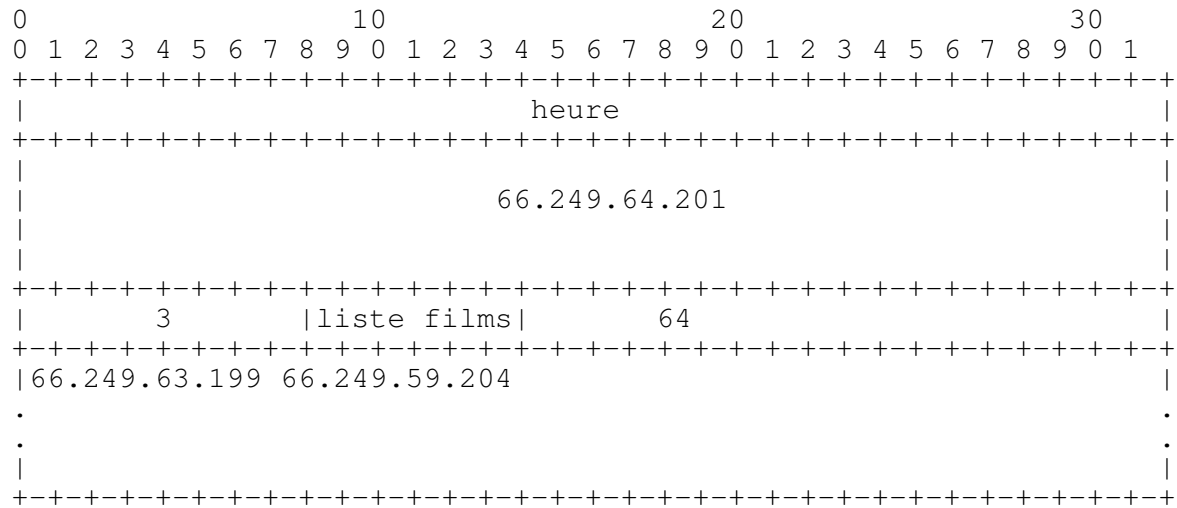


Figure 7

heure = 1000111100001101000110001010

66.249.64.201 = 1000010.11111001.1000000.11001001

numero de sequence = 11

type = 00010000

Taille message = 1000000

message = 1000010.11111001.1000000.11001001
1000010.11111001.111011.11001100

Le client envoie un acquittement

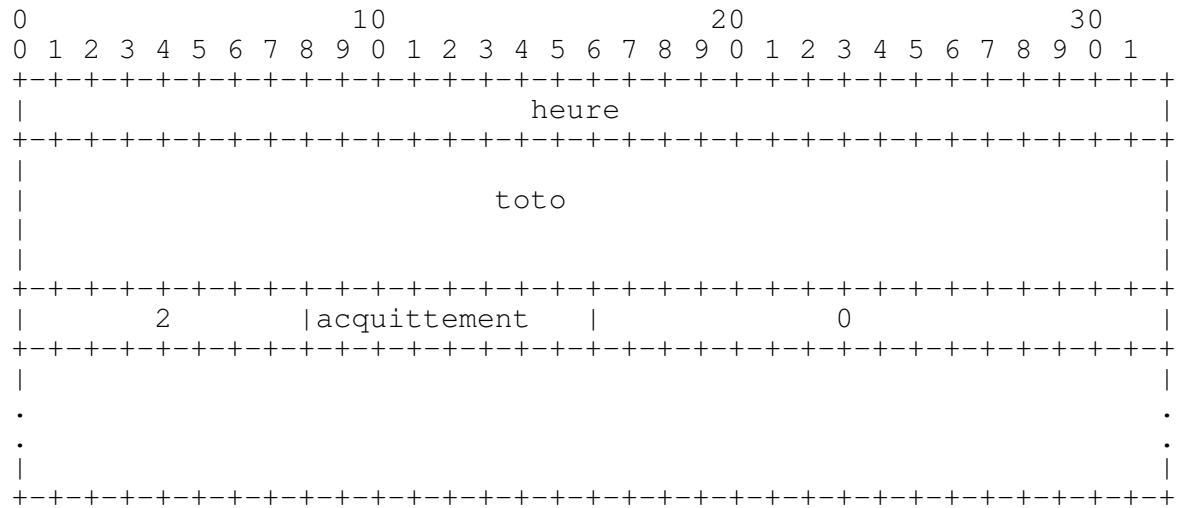


Figure 8

heure = 1000111100001101000110001010

identifiant = 01110100 01101111 01110100 01101111

numero de sequence = 10

type = 00000011

message = vide

On suppose maintenant que tous les messages sont envoyés correctement et on ne se préoccupe plus des acquittements. L'utilisateur choisit maintenant le film 1 et se connecte à la Movie Room

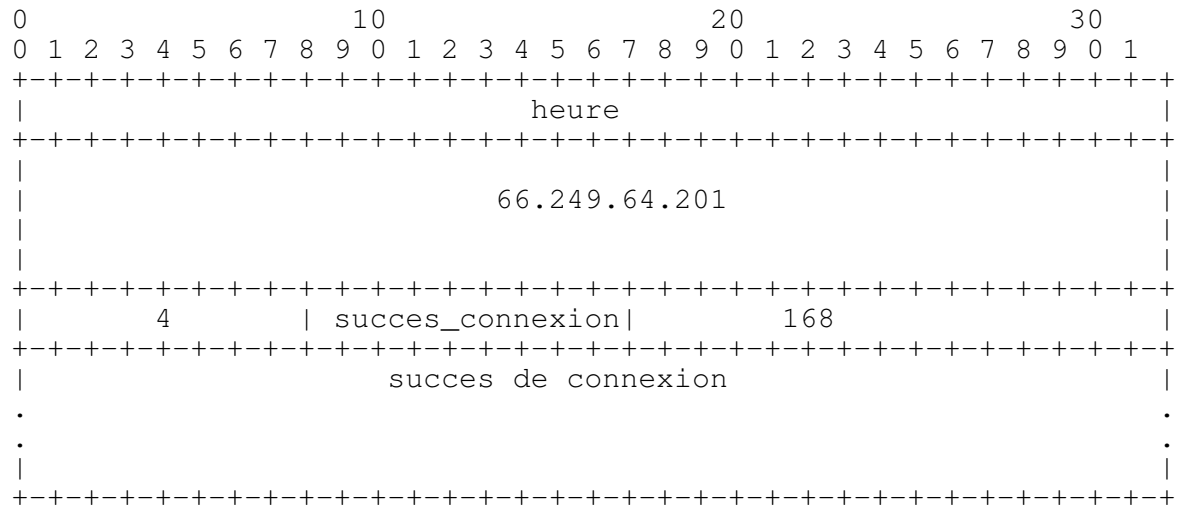


Figure 10

```
heure = 100011110000110100011010010
```

$$66.249.64.201 = 1000010.11111001.1000000.11001001$$

```
numero de sequence = 100
```

```
type = 00000010
```

Taille message = 10101000

```
message = 01110011 01110101 01100011 01100011 11101000 01110011
00100000 01100100 01100101 00100000 01100011 01101111 01101110
01101110 01100101 01111000 01101001 01101111 01101110 00001101
00001010
```

L'utilisateur envoie maintenant un message aux autres utilisateurs via le chat. Il envoie d'abord un message de type `requete_Chat` au serveur qui envoie ensuite un message de type `Response_Chat` aux autres utilisateurs

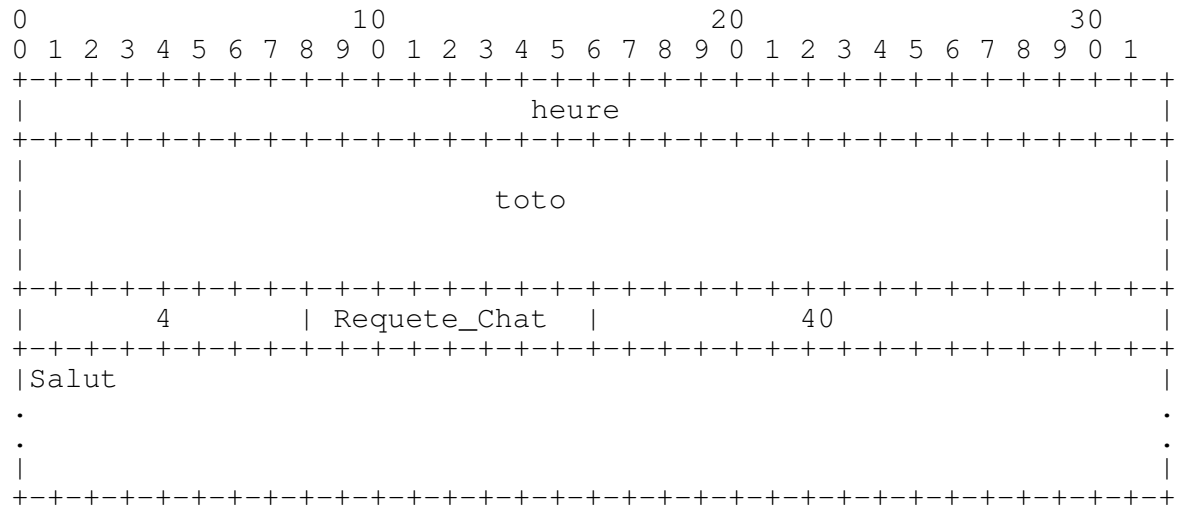


Figure 11

```

heure = 1000111100001101000111000010
toto = 01110100 01101111 01110100 01101111
numero de sequence = 100
type = 00100000
Taille message = 101000
message = 01010011 01100001 01101100 01110101 01110100

```

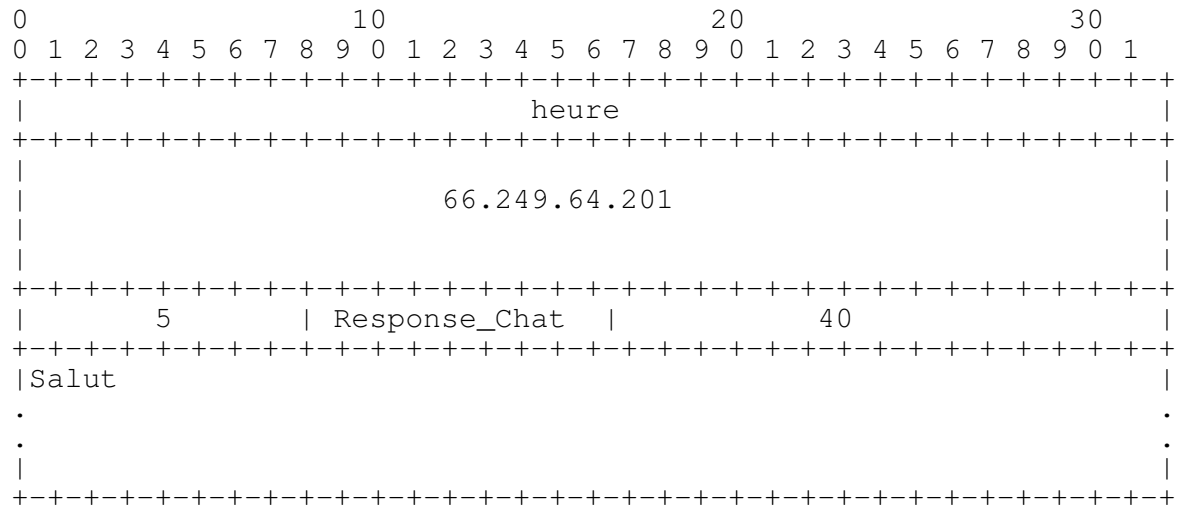


Figure 12

```

heure = 1000111100001101000111000010
toto = 01110100 01101111 01110100 01101111
numero de sequence = 101
type = 00100000
Taille message = 101000
message = 01010011 01100001 01101100 01110101 01110100
L'utilisateur se deconnecte.
    
```

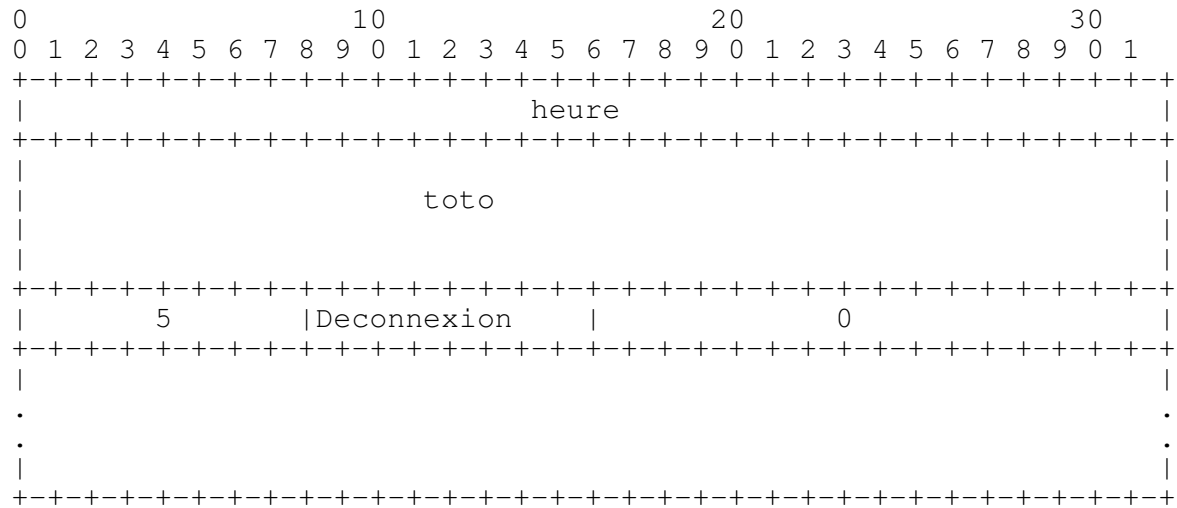


Figure 13

heure = 1000111100001101000111111010

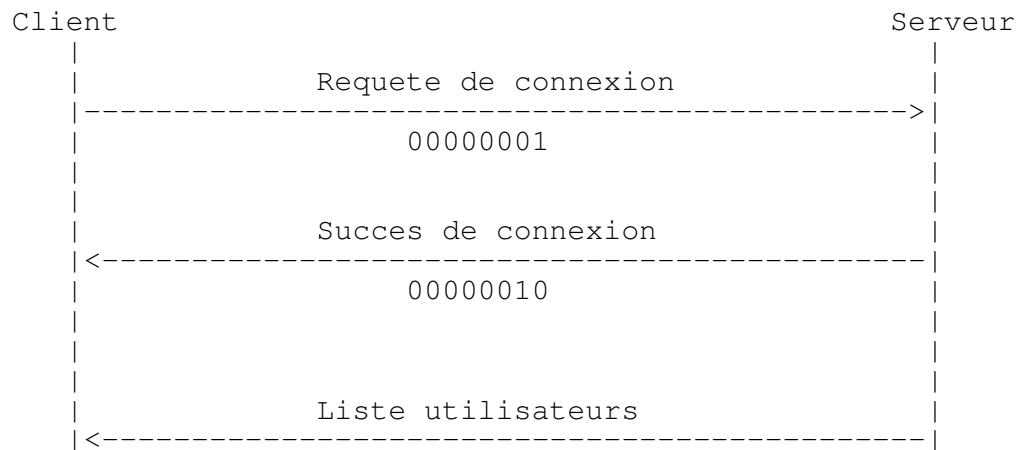
toto = 01110100 01101111 01110100 01101111

numero de sequence = 101

type = 10000000

Taille message = 0

Le scenario correspond au diagramme de sequence suivant:



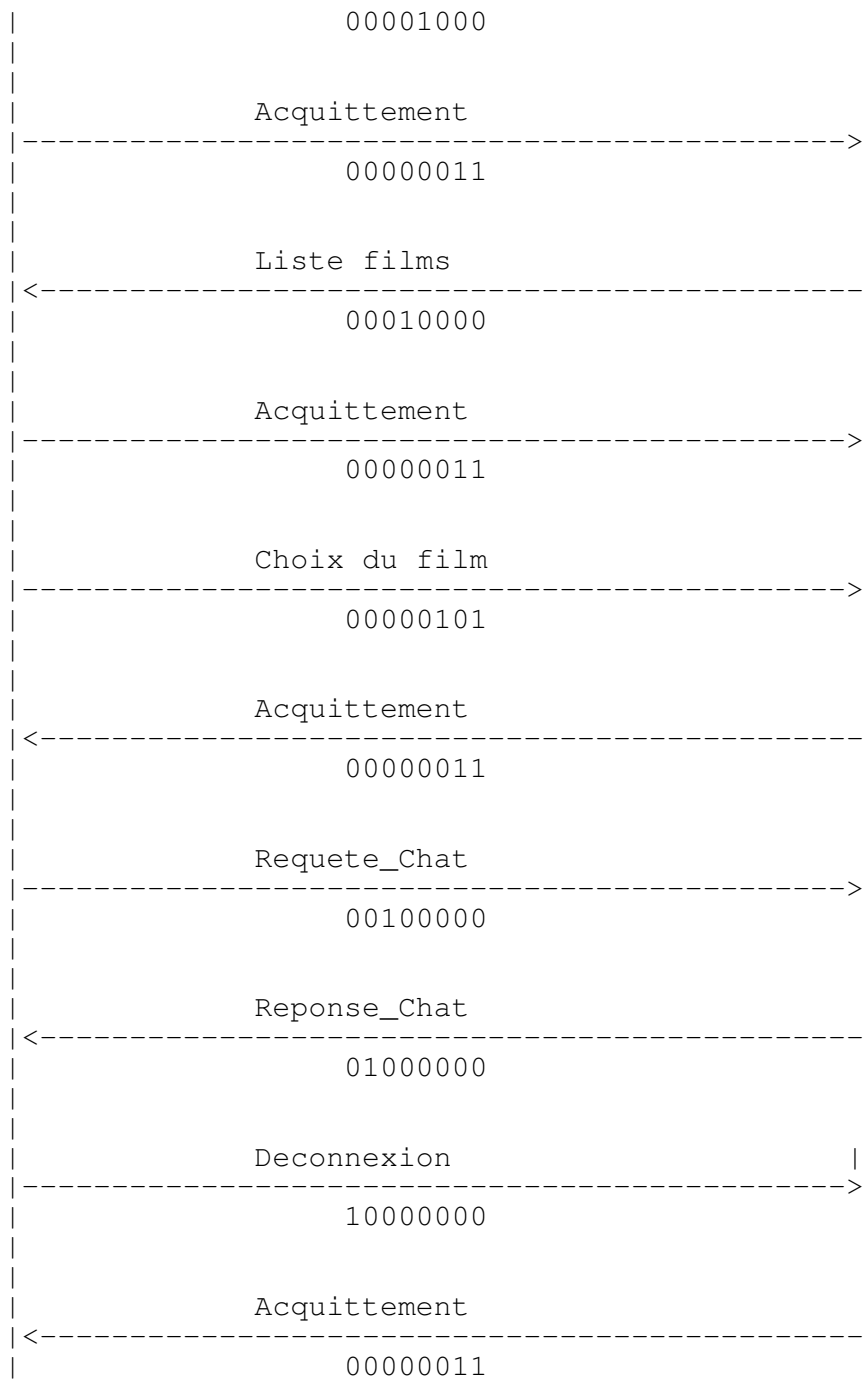


Figure 14

9. Conclusion

Ce document repertorie donc l'ensemble des specifications du protocole, necessaire au bon fonctionnement de l'application c2w.

Authors' Addresses

AUFFRAY--PLESSIX Anne-Sibylle
Telecom Bretagne

Email: as.auffray--plessix@telecom-bretagne.eu

DIEYE Souleymane
Telecom Bretagne

Email: souleymane.dieye@telecom-bretagne.eu

LESAGE Vincent
Telecom Bretagne

Email: vincent.lesage@telecom-bretagne.eu

TOBITT Eric
Telecom Bretagne

Email: eric.tobitt@telecom-bretagne.eu