

# “NetRadio”

## Projet de Programmation Réseaux

License Informatique - Université de Paris

2021

*Attention :* Prenez le temps de lire attentivement TOUT le document et ce dans ses moindres détails. Relisez plusieurs fois, même en groupe. Ne vous lancez pas immédiatement dans la réalisation à moins que vous ne soyez déjà à l’aise avec le développement réseau et la communication inter-applications. Faites donc des diagrammes, imaginez des scénarios de communication entre entités, essayez d’en analyser la portée, les problèmes, les informations dont les entités ont besoin au cours de leur vie, etc. Essayez d’abord de vous abstraire des problèmes techniques.

*Note :* Dans le document le signe `_` représentera un simple caractère d’espace (ASCII 32). De plus les messages circulant sont indiqués entre crochets, les crochets ne faisant pas partie du message. À la fin du document le format des parties constituant les messages est décrit de façon précise.

## Introduction

Le but de ce projet est de programmer différentes entités mettant en place un service de diffusion de messages de 140 caractères. Il y aura trois types d’entités :

1. Des diffuseurs de messages ;
2. Des clients capable de recevoir les messages diffusés ;
3. Des gestionnaire de diffuseurs de messages.

Les diffuseurs enverront en multi-diffusion des messages en continu, ils pourront aussi envoyer des messages d’utilisateurs et renvoyer les derniers messages à un utilisateur sur demande.

Les gestionnaires auront chacun une liste de diffuseurs (avec leurs caractéristiques) et pourront transmettre cette liste aux clients qui la demanderont. Ils devront aussi s’assurer que les diffuseurs présents dans la liste sont encore actifs.

Les clients devront être capable d’écouter les messages des diffuseurs, de leur envoyer des messages particuliers, de leur demander une liste des derniers messages et d’interroger les gestionnaires pour récupérer des informations sur les diffuseurs.

## Description détaillée

### Communication avec les diffuseurs

Les caractéristiques d’un diffuseur sont les suivantes :

- Un identifiant (d’au plus 8 caractères)

- Un port pour recevoir les messages d'utilisateurs (inférieur à 9999)
- Une adresse IPv4 de multi-diffusion
- Un port de multi-diffusion (inférieur à 9999)

Il est important de comprendre qu'un diffuseur gèrera sa liste de messages à diffuser. Parmi ces messages, il y aura les messages du diffuseur mais aussi les messages que les utilisateurs auront demandé à transmettre. Commençons tout d'abord par décrire comment un diffuseur diffuse ses messages. Il envoie sur son adresse et port de multi-diffusion des messages de la forme [DIFF\_num-mess\_id\_message] où num-mess est le numéro du message, qui va de 0 à 9999, id est l'identifiant de l'entité ayant rédigé le message, il peut s'agir soit du diffuseur, soit d'un utilisateur, et message est le message contenant au plus 140 caractères. Les numéros de message seront incrémentés à chaque envoi de message (et quand on arrive à 9999 on repart à 0). La fréquence d'envoi des messages n'est pas précisée, mais évitez d'envoyer des messages trop souvent ou trop rarement (à vous de déterminer la fréquence que vous souhaitez).

Un diffuseur peut de plus recevoir des messages d'utilisateurs qu'il pourra diffuser. La transmission d'un message par un utilisateur à un diffuseur se fait en mode connecté. Pour transmettre un message, un utilisateur se connecte au port de réception du diffuseur et lui envoie un message de la forme [MESS\_id\_message] où id est l'identifiant de l'utilisateur et message est le message à transmettre contenant au plus 140 caractères. Le diffuseur renvoie alors un message de la forme [ACKM] et ferme la connexion.

Un utilisateur peut aussi demander une liste des derniers messages diffusés. Ce mode de communication a également lieu en mode connecté. Pour ce faire un utilisateur se connecte au port de réception du diffuseur et lui envoie un message de la forme [LAST\_nb-mess] où nb-mess est un entier (compris entre 0 et 999). L'utilisateur demande ainsi au diffuseur les nb-mess derniers messages diffusés. À ce message, le diffuseur répond par une séquence d'au plus nb-mess messages de la forme [OLDM\_num-mess\_id\_message] où num-mess, id et message sont identiques aux champs des messages de type MESS. Cette séquence contient les derniers messages diffusés. Leur nombre peut être inférieur à nb-mess. Lorsque le diffuseur a fini de transmettre sa séquence de messages, il le signale à l'utilisateur en envoyant un message de la forme [ENDM] et il ferme ensuite la connexion.

## Communication avec les gestionnaires

Les caractéristiques d'un gestionnaire sont les suivantes :

- Un port de communication (inférieur à 9999)
- Un nombre maximal de diffuseurs enregistrés

Un diffuseur a la possibilité de s'enregistrer auprès d'un gestionnaire, qui peut être vu comme un annuaire de diffuseurs. Un utilisateur peut ensuite interroger un gestionnaire pour connaître la liste de ses diffuseurs. Toutes les communications avec le gestionnaire ont lieu en mode connecté.

Pour s'enregistrer auprès d'un gestionnaire, un diffuseur lui envoie sur son port de communication un message de la forme [REGI\_id\_ip1\_port1\_ip2\_port2] où id est l'identifiant du diffuseur, ip1 et port1 correspondent à son adresse IPv4 et port de multi-diffusion, ip2 est l'adresse IPv4 de la machine où il se trouve et port2 est son port de réception des messages d'utilisateurs. Si le gestionnaire réussit à enregistrer le diffuseur (si il n'a pas encore atteint son nombre maximal de diffuseurs), il répond en envoyant le message [REOK] et il ne ferme pas la connexion. Si en revanche, il ne peut pas enregistrer le diffuseur, il répond en envoyant [RENO] et il ferme la connexion.

Afin de garantir que sa liste de diffuseurs est bien à jour et pour éventuellement libérer de la place, le gestionnaire peut tester si un diffuseur est toujours actif. Pour cela, il lui envoie un message de la forme [RUOK] auquel le diffuseur doit répondre avec un message [IMOK], au bout d'un certain délai que vous choisirez. Si le diffuseur ne répond pas ou renvoie une mauvaise réponse, le gestionnaire, peut le sortir de la liste et fermer la connexion correspondante. Il convient de faire attention à ne pas tester trop souvent la présence d'un diffuseur pour ne pas que celui-ci passe son temps à répondre au gestionnaire.

Un utilisateur peut interroger un gestionnaire pour connaître des diffuseurs. Pour cela il se connecte sur le port du gestionnaire et il lui envoie un message de la forme [LIST] auquel le gestionnaire répond

en envoyant la liste des diffuseurs dont il dispose. Le gestionnaire commence par envoyer un message [LINB\_num-diff] où num-diff est un entier compris entre 0 et 99 précisant le nombre de messages qu'il va envoyer, sachant qu'il enverra autant de messages qu'il dispose de diffuseurs dans sa liste. Chaque message envoyé ensuite aura la forme suivante [ITEM\_id\_ip1\_port1\_ip2\_port2] où les informations transmises sont similaires à celles contenues dans les messages de type REGI. Après avoir envoyé tous ces messages, le gestionnaire fermera la connexion.

## Les utilisateurs

Les caractéristiques d'un utilisateur sont les suivantes :

- Un identifiant (d'au plus 8 caractères)

Les utilisateurs communiquent avec les diffuseurs et les gestionnaires comme précédemment expliqué.

## Spécification de la forme des messages

Afin que vos projets puissent communiquer entre eux, il est important que tous les messages respectent une forme bien précise. Nous décrivons dans cette partie de façon plus détaillée le format de chacun des messages.

Tout d'abord chaque message terminera nécessairement par deux octets servant à encoder les caractères '\r' (retour chariot) et '\n' (nouvelle ligne).

Comme vous avez pu le remarquer, les quatre premiers octets d'un message serviront à encoder dans une chaîne de caractères son type. Les types des messages étant : DIFF, MESS, ACKM, LAST, OLDM, ENDM, REGI, REOK, RENO, RUOK, IMOK, LIST, LINB, ITEM.

Voilà maintenant les caractéristiques pour chacun des champs contenus dans les messages :

- num-mess sera codé sur 4 octets et contiendra la chaîne de caractères correspondant au numéro du message. Par exemple, pour le message 120, num-mess vaudra 0120.
- id sera codé sur 8 octets et contiendra une chaîne de caractères. Si l'identifiant contient moins de 8 caractères, alors on complètera la fin de la chaîne avec des caractères #. Par exemple si l'identifiant est RADIO, alors id vaudra RADIO###.
- mess sera codé sur 140 octets et contiendra une chaîne de caractères. Si le message contient moins de 140 caractères, mess sera complété comme id avec des caractères # à la fin.
- nb-mess sera codé sur 3 octets et contiendra la chaîne de caractères correspondant au nombre de messages. Si la chaîne de caractères contient moins de 3 caractères, on la complètera avec des 0 au début comme pour num-mess.
- ip1 et ip2 seront codés sur 15 octets et contiendront la chaîne de caractères correspondant à l'adresse IPv4. Par exemple, si l'adresse IP est 127.0.0.1, alors ip1 contiendra 127.000.000.001.
- port1 et port2 seront codés sur 4 octets et contiendront la chaîne de caractères correspondant au numéro de port.
- num-diff sera codé sur 2 octets et contiendra la chaîne de caractères correspondant au nombre de diffuseurs. Si la chaîne de caractères contient moins de 2 caractères, on la complètera avec des 0 au début comme pour num-mess.

On remarque donc que chaque message a une unique taille fixe.

## Réalisation

La réalisation se fera nécessairement en C ou en Java, et possiblement dans les deux (ce qui serait un très bon exercice et un point pris en compte lors de l'évaluation). Un groupe devra réaliser au moins un utilisateur, un diffuseur et un gestionnaire.

Il est impératif de respecter scrupuleusement la spécification fournie dans le sujet ainsi que les formats de message. Toute violation sera jugée très défavorablement !

Il est vivement recommandé de proposer des extensions du projet, par exemple en proposant des nouveaux types de messages pour diffuser des fichiers, en demandant à un diffuseur de retransmettre un message particulier, en autorisant un utilisateur à envoyer un message à un gestionnaire qui devra le transmettre à tous ses diffuseurs ... etc. Sur ces points **à vous de jouer** ! Ces extensions devront être décrites de façon précise au moment du rendu du projet.

Nous mettrons en place un salon dédié sur Discord pour que vous puissiez communiquer entre vous et avec nous, par exemple pour faire part d'imprécisions dans le sujet, ou pour signaler que vous avez une entité qui tourne quelque part et ainsi donner l'opportunité à vos collègues de se connecter à elle. Ou encore pour préciser vos extensions. Toute demande sur le projet devra passer par le forum. Prenez garde à ne pas vous fier à la rumeur, la seule source d'information fiable sera le salon sur Discord. Au moindre doute, n'hésitez pas à y poster votre question !

La communication verbale entre groupes est non seulement autorisée mais encouragée, cependant il est **strictement interdit** d'échanger du code ; ceci serait considéré comme plagiat et par conséquent jugé sévèrement. Les discussions doivent donc seulement porter sur le fonctionnement du protocole et son interprétation, ou les formats des messages ; il vaut donc mieux éviter de donner trop d'indications sur la façon de coder les fonctionnalités.

La **notation finale** prendra en compte le fait que des groupes auront réussi à faire communiquer leurs applications entre elles. Il faudra donc penser à en faire la démonstration. D'une certaine manière, ceux qui collaborent dans les limites indiquées ci-dessus seront récompensés ; pour ceux qui décident de faire quelque chose dans leur coin et qui ressemblerait vaguement à ce qui est décrit, ce sera l'inverse, la notation sera revue à la baisse. Un **programme qui s'exécute n'est pas suffisant** ! Vous **devez** écrire un programme qui se comporte comme indiqué ; et s'il ne communique pas avec le programme écrit par d'autres, c'est que vous n'avez pas compris ce qu'est la programmation réseau.

Pour la soutenance, il sera **nécessaire de prévoir un mode de fonctionnement verbeux** dans lequel suffisamment d'informations seront disponibles à l'écran pour comprendre ce qui se passe (affichage des adresses utilisées, ports, états des listes internes, etc).

Vos programmes devront nécessairement pouvoir être exécutés sur les machines des salles de TP de l'UFR. Toute solution ne respectant pas ce critère sera jugée invalide.

Il n'est pas demandé de fournir une interface graphique ; il est même recommandé de s'abstenir d'en faire une (quoique vous en pensiez). Cela ne vous apportera rien, vous fera perdre du temps et n'impressionnera en rien les examinateurs ; et surtout **ce n'est absolument pas dans le programme de cet enseignement** par conséquent, nous insistons : **pas d'interface graphique**.

Votre projet devra bien entendu être robuste (c'est à dire sans erreur) et devra être capable de gérer des messages erronés sans planter (vous pouvez par exemple soit les ignorer, soit fermer la connexion correspondante ou décider d'un message à envoyer en cas de problème).

La réalisation du projet se fera par groupe **d'au moins deux** étudiants et **d'au plus trois** (cette règle ne souffrira aucune exception, un peu d'effort de la part de chacun pour ne pas prendre que des amis proches dans son groupe devrait aider). Et bien entendu, chacun dans un groupe devra travailler et il n'est pas exclu que des étudiants d'un même groupe n'aient pas la même note au final.

La composition des groupes devra être envoyée par mail à [sangnier@irif.fr](mailto:sangnier@irif.fr) avant **le Vendredi 26 Mars 2021 23h59**. Toute personne n'ayant pas soumis de groupe avant cette date prend le risque de ne pas avoir de note au projet.

Le projet devra être rendu sur Moodle la veille de la soutenance, dont la date vous sera fournie ultérieurement. Des informations sur la soutenance (ordre de passage et instructions) seront aussi fournies plus tard sur la page web du cours.