

## Communication multicast

La communication en multicast est une communication entre un ou plusieurs émetteurs et un ou plusieurs récepteurs, l'ensemble représentant ce que l'on appelle un "groupe multicast". Les adresses IP ayant leur premier octet à une valeurs supérieure ou égale à 224 sont des adresses multicast. Pour réaliser un groupe multicast, il faut que les émetteurs et récepteurs s'abonnent à une même adresse IP multicast. A partir de ce moment, tout ce qui envoyé par un émetteur est reçu par tous les récepteurs (et également les autres émetteurs s'ils désirent écouter). En plus de l'adresse IP multicast, les récepteurs doivent tous attendre sur un même numéro de port et les émetteurs doivent envoyer leurs informations vers ce numéro de port (il n'y a pas de numéro de port réservés pour le multicast).

La classe `MulticastSocket` est utilisée pour envoyer ou recevoir des données d'une adresse multicast. Un socket multicast est un `DatagramSocket` avec des propriétés supplémentaires pour se connecter à un groupe multicast sur l'internet. Un groupe multicast est représenté par une adresse IP de classe D, ce qui fait que les adresses multicast sont comprises entre 224.0.0.1 et 239.255.255.255, et un numéro de port.

- Pour créer un socket multicast, il est nécessaire d'instancier la classe `MulticastSocket` grâce à son constructeur `MulticastSocket(int port)`. Si l'on est récepteur, il faut préciser dans le constructeur le numéro de port sur lequel on écoute. Si l'on est client, ce numéro de port sera nécessaire pour construire et envoyer les datagrammes vers les récepteurs.
- Une fois un socket multicast créé, il est possible d'exécuter quatre méthodes sur lui qui sont :
  - Joindre un groupe multicast. Cela est nécessaire pour recevoir des données d'un groupe multicast et se réalise à l'aide de la méthode `joinGroup(InetAddress i)` à laquelle on passe en paramètre une `InetAddress` représentant l'adresse multicast.
  - Envoyer des données aux membres du groupe multicast grâce à la méthode `send(DatagramPacket dp, byte ttl)` à laquelle on passe en paramètre un `DatagramPacket` contenant la donnée à envoyer et son Time-To-Live afin d'empêcher les boucles de routage.
  - Recevoir des donnée des membres du groupe multicast grâce à la méthode `receive(DatagramPacket dp)` à laquelle on passe en paramètre un `DatagramPacket` destiné à recevoir la donnée envoyée par un des autres membres du groupe.
  - Quitter le groupe multicast à l'aide de la méthode `leaveGroup(InetAddress i)` à laquelle on passe en paramètre l'adresse multicast à quitter.

## Programme de discussion interactive

Un programme de discussion interactive (un *chat*) permet à plusieurs utilisateurs de s'envoyer des messages. Avec le multicast, faire en sorte que chaque message soit reçu par tous les utilisateurs.