TP Interconnexions de Réseaux
Étude de la communication de groupe dans Internet: le Multicast

Pascal Sicard

11 février 2022

1 INTRODUCTION

Le concept du *multicast* est né dans les années 90. Steve Deering, dans le RFC 988, décrit cette nouvelle méthode de transmission de datagrammes IP vers des groupes de machines. Deux autres versions de ce RFC ont suivi dont le RFC 1112 datant du mois d'août 1989.
Le premier tunnel a été établi durant l'été 1988, entre deux universités américaines. Les premières expériences ont pu commencer. Il s'agissait de transmettre de l'audio grâce au multicast. Ne disposant pas de routeurs « multicast », ils se sont rapidement aperçus qu'il était plus simple, pour l'expansion du réseau, de mettre en place des *tunnels* (échange multicast via une liaison unicast) plutôt que de demander de modifier les fonctionnalités de l'ensemble des routeurs. Le nom de Mbone (Dorsale Multicast) date de juillet 1992 et a été attribué à la réunion de l'IETF (Internet Engineering Task Force).

Depuis, les expériences n'ont cessé de se multiplier, créant progressivement un réseau virtuel bâti au-dessus de l'Internet, appelé le MBone. Le mutlicast s'avère des plus intéressant pour toutes les applications gourmandes en bande passante : conversation audio à plusieurs, diffusion de vidéo-conférence, diffusion de radio, télévision... En effet le nombre de paquet circulant sur le réseau est largement diminué par rapport à des dialogues *unicast*.

Malheureusement le développement du Mbone ne s'est pas intensifié et son utilisation reste limité à quelques applications Universitaire et en interne pour des fournisseurs d'accès (visioconférence, Webradios, chaînes de télévision).

A savoir : tous les principes que nous allons étudier avec IPV4 existent de la même manière avec IPV6.

2 Définition du multicast

2.1 Rappel : l'unicast et le broadcast

Dans l'Internet, lorsque l'on désire échanger des données entre des machines, on utilise principalement l'***unicast*** pour communiquer entre deux machines. L'unicast permet à une machine, dite *source*, d'envoyer des paquets vers une seule machine destinataire. Chaque machine possède une adresse Internet (IP) et l'échange n'est possible entre applications que grâce au couple (adresse IP/numéro de port) des deux entités en jeu. Le numéro de port de la destination (le serveur) doit être un numéro connu (« well known port ») de l'expéditeur (le client), il est associé à un ***service*** réseau. Par contre le numéro de port de l'expéditeur peut être attribué dynamiquement par le système d'exploitation du client.
Rappel : Le ***broadcast*** permet à une machine d'envoyer des paquets à toutes les machines d'un réseau local.

2.2 Le multicast

Le concept du ***multicast*** permet d'envoyer des paquets non plus à destination d'une seule machine mais vers un groupe de machines. Pour cela on définit une adresse de groupe de machines. En IPV4 les adresses comprises entre 224.0.0.0 et 239.255.255.255 ont été réservées pour désigner des groupes. Concrètement, qu'est ce que cela signifie ?

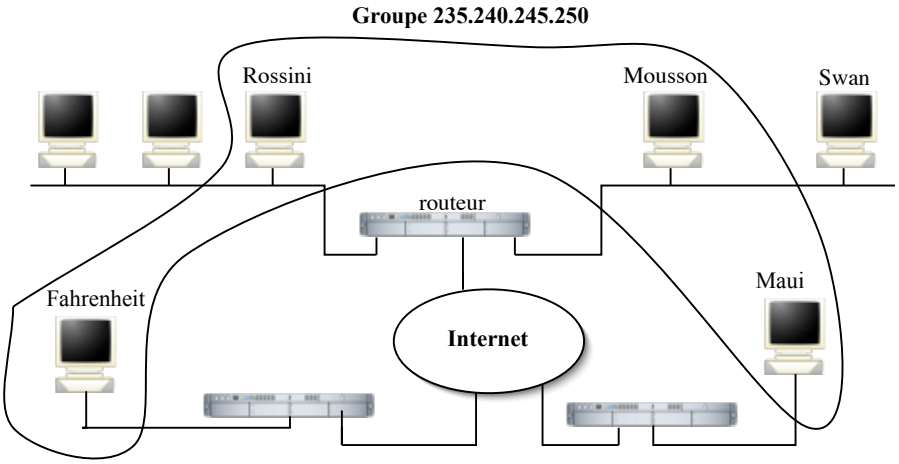

FIGURE 1 – Un groupe de machines

Prenons l'exemple de la figure 1 : Le groupe 235.240.245.250 comprend les machines *mousson, rossini* qui se trouvent sur un réseau local à l'IMAG mais séparées par un routeur, *maui.inria.fr* en France et *fahrenheit.nasa.gov* aux Etats-Unis.
Lorsque ***mousson*** envoie un paquet à destination de ce groupe, toutes les machines en faisant partie le recevront. Attention le paquet n'est envoyé qu'une seul fois . De plus la

machine ***swan*** bien que faisant partie du réseau local de ***mousson*** n'appartient pas au groupe et ne reçoit donc pas de paquet à destination de ce groupe.

3 La notion de groupe multicast

Un groupe est un ensemble de zéro ou plus de machines. Il est représenté/désigné par une adresse IP dite ***multicast***. L'ensemble des membres d'un groupe est entièrement **dynamique**.
Cela signifie que toute machine peut, à tout moment, se joindre au groupe ou bien le quitter. Il n'y a absolument aucune restriction sur le nombre de membres d'un groupe ainsi que sur leur localisation physique. Une machine peut être en même temps membre de plusieurs groupes.
Il n'est toutefois pas nécessaire d'être membre d'un groupe pour y envoyer des datagrammes.

Un groupe multicast peut être permanent ou non. Un groupe permanent est en fait un groupe qui possède une adresse assignée par une autorité. Il faut bien comprendre que c'est l'adresse qui est permanente et non pas l'ensemble des membres d'un groupe : à tout moment un groupe permanent peut avoir n'importe quel nombre de membres y compris zéro !
Les adresses IP, non assignées à des groupes permanents peuvent être utilisées pour les groupes temporaires qui n'existent que tant qu'ils possèdent des membres.

Concrètement, les adresses IPv4 disponibles pour le multicast sont les adresses IP de classe D (4 bits de poids forts : 1110), en notation décimale, les adresses comprises entre 224.0.0.0 et 239.255.255.255.
Cela représente approximativement 250 millions d'adresses.
Remarque : les adresses entre 224.0.0.0 et 224.0.0.255 sont réservées pour les algorithmes de routage (RIP, OSPF ...).

4 Emission/réception de paquets multicast sur une machine utilisateur (hôte)

Le multicast est implémenté dans l'interface classique des *sockets*. A l'heure actuelle, seules les sockets AF_INET (pour IPv4) ou AF_INET6 (pour IPV6) de type SOCK_DGRAM (UDP) ou SOCK_RAW (IP) sont supportées.

Une fois une socket créee, on peut décider de ses paramètres multicast :

- **Limitation de la diffusion :**
  On peut limiter la diffusion des paquets multicast grâce au champ TTL(*Time To Live* ou durée de vie) de l'entête IP. A chaque socket multicast est associé un TTL,

qui par défaut prend la valeur 1 correspondant au sous-réseau local. Le TTL permet de contrôler la portée de l'émission.
Ceux sont les routeurs multicast qui donnent un sens à cette valeur, en décrémentant (d'un palier donné) ce champ au passage d'un paquet et en détruisant les paquets quand il passe en dessous de 0.
La figure 2 donne les limitations suggérées pour le TTL suivant la portée que l'on veut donner à un paquet.

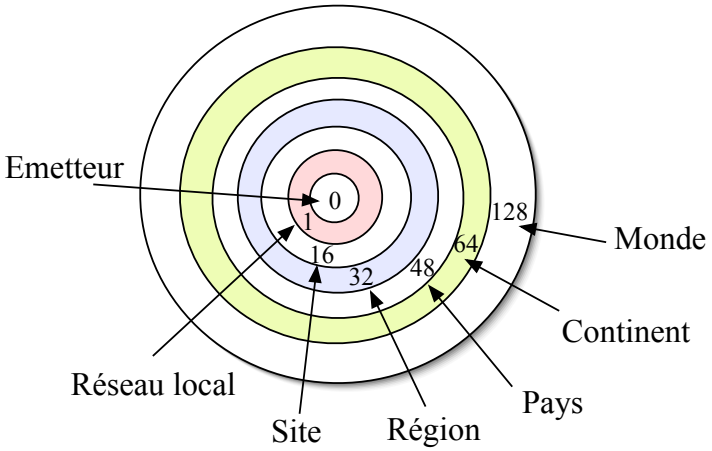

FIGURE 2 – Portée en fonction du TTL

La valeur initiale du TTL est configurable grâce à la fonction **setsockopt()** :

```
u_char ttl;
setsockopt(sock, IPPROTO_IP, IP_MULTICAST_TTL, &ttl, sizeof(ttl));
```

- **Se joindre à un groupe :**
  Pour qu'une application puisse recevoir des datagrammes multicast, il est nécessaire qu'elle devienne membre d'un groupe. Cela se fait avec la même fonction **setsockopt** :

```
struct ip_mreq (
      struct in_addr imr_multiaddr;    /*groupe multicast a joindre */
      struct in_addr imr_interface;     /*interface (INADDR\_ANY) */
                );
struct ip_mreq mreq;
setsockopt(sock, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(mreq));
```

  Lorsque l'on joint un groupe, on ne spécifie que l'adresse IP du groupe. Pour le numéro de port, il faut faire un traditionnel **bind()** sur ce port.
  Si l'on veut que plusieurs applications puissent utiliser la même adresse de groupe (et donc le même numéro de port) sur la même machine, il faut utiliser l'appel suivant :

```
int one=1;
setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &one, sizeof(one));
```

- **Pour quitter un groupe :**

On utilise à nouveau la même fonction.

```
setsockopt (sock, IPPROTO_IP, IP_DROP_MEMBERSHIP, &mreq, sizeof (mreq));
```

On peut noter toutefois que lorsque la socket est fermée, l'abandon du groupe est automatique. Dans un cas comme dans l'autre, si la machine possède d'autres sockets en liaison avec ce groupe, la machine reste membre de ce groupe.

Remarque : Ces différents appels sont implémentés dans **socklab (mode udp)** avec les primitives (msocket, mbind, mjoin, mleave ...).

5 Le multicast au niveau Ethernet

Lors qu'une machine se joint à un groupe, il faut aussi que le niveau liaison de données (Ethernet par exemple) est connaissance de se groupe afin de pouvoir réceptionner les paquets à destination d'un groupe.
Une adresse Ethernet est donc calculée à partir de l'adresse IP multicast (01 :00 :5E :(0, 23 bits de poids faible de l'adresse IP). Au moment ou une machine se joint à un groupe, la couche Ethernet sera informée et "remontera" par la suite les paquets à destination de cette adresse Ethernet multicast.

Les switchs Ethernet actuels gèrent le mutlicast en maintenant une table port/adresse Ethernet multicast permettant de rediriger des paquets multicast vers les membres d'un groupe.

6 La gestion du multicast dans les routeurs

6.1 Le réseau multicast dans Internet : le MBone

MBone signifie "Multicast BackBone ". Il s'agit d'un sous-ensemble de routeurs d'Internet supportant le routage des paquets multicast, cette fonction n'étant pas implémentée dans tous les routeurs d'Internet.
En fait, ce réseau est composé d'îlots de réseaux pouvant supporter directement le multicast (typiquement des LANs multicast comme Ethernet) reliés par des liaisons virtuelles point-à-point que l'on appelle des ***tunnels***. L'extrémité d'un tunnel est constituée typiquement d'une machine de type hôte sur laquelle tourne le démon permettant de traiter le routage des paquets multicast ou d'un routeur traitant le multicast.

Le fonctionnement d'un tunnel est le suivant : les paquets IP multicast sont encapsulés avant la transmission dans des datagrammes UDP unicast classiques. L'adresse Internet destination de ce paquet est l'adresse unicast du prochain routeur traitant le multicast . A l'autre extrémité, le routeur multicast, fait l'opération inverse, il désencapsule le paquet unicast (cf figure 3). Il détermine ensuite à quels routeurs le paquet multicast doit être

envoyé au vu de l'adresse destination multicast et de sa table de routage (éventuellement via d'autres tunnels).

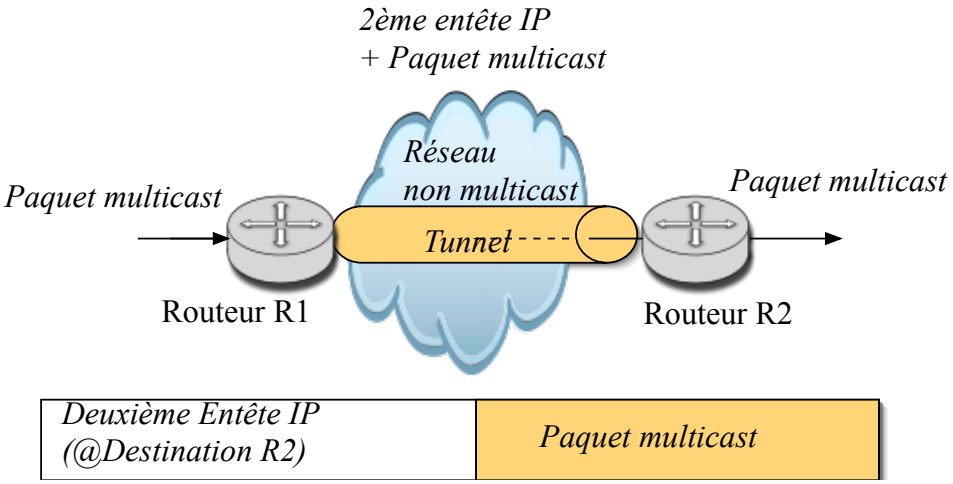

FIGURE 3 – Un tunnel entre deux routeurs

Il existe depuis 2001 un Mbone IPV6 : le M6Bone. Il reprend certains des algorithmes de routage utilisés dans IPV4 (voir $http://livre.g6.asso.fr/index.php/Main_Page$)

6.2 Dialogue machine utilisateur / routeur multicast

Une application peut se joindre à tout moment à un groupe. L'appel de **setsockopt()** avec **IP_ADD_MEMBERSHIP** fait que la machine devient membre du groupe spécifié (appelons le G).
Au moment où elle se joint à un groupe, la machine déclare au routeur multicast de son réseau local qu'elle fait partie de G. Ainsi les paquets multicast qui arrivent au routeur en provenance de G seront transmis jusqu'à elle. Lorsque l'on quitte un groupe, il faut aussi le déclarer au routeur pour minimiser le traffic (les paquets à destination du groupe ne seront plus envoyés à la machine). On peut dire qu'une machine fait partie d'un groupe tant qu'au moins une application tournant dessus possède une socket ouverte vers ce groupe.

C'est le protocole IGMP (Internet Group Management Protocol) qui est utilisé pour ce dialogue routeur/hôte.

Format des paquets IGMP version 2 :

- Champ *Type* (1 octet)
  - **0x11** : Demande d'appartenance à un groupe émis par le routeur multicast vers :
    1. Toutes les machines « utilisateurs » du LAN pour connaître l'ensemble des groupes existant sur le réseau (@destination= 224.0.0.1).
    2. Les machines d'un groupe pour savoir s'il existe encore des machines appartenant à un groupe donné (@destination= @du groupe en question).

Remarques : Dans le cas où plusieurs routeurs multicast sont sur le même réseau local, seul celui d'adresse IP la plus petite émettra ce type de paquet (les autres n'émettront pas au vu des paquets envoyés par ce dernier). Les paquets d'appartenance générale sont émis régulièrement pour parer aux pertes diverses de paquets IGMP.

- **0x16** : émis par les hôtes en réponse à une demande d'appartenance ou au moment où l'hôte se joint à un groupe donné (l'adresse destination IP est l'adresse du groupe en question).
  Pour limiter le trafic de ce type de paquet, les hôtes arment un temporisateur d'une durée prise aléatoirement entre 0 et Temps-de-réponse-max (voir champ suivant). L'hôte n'émet le paquet qu'à l'expiration du timer (un par groupe). S'il reçoit entre temps un autre paquet d'appartenance provenant d'un autre hôte pour ce groupe, il ne sert à rien qu'il émette ce paquet à son tour.

- **0x17** : Emis par un hôte lorsqu'il quitte un groupe à destination des routeurs du réseau (l'adresse destination= 224.0.0.2).

- Champ *Temps de réponse max* (1 octet) : Attente maximum avant l'envoi d'un rapport d'appartenance
- Champ *Checksum* (2 octets)
- Champ *Adresse de groupe* (4 octets) : Spécifie le groupe concernant le paquet IGMP. 0 pour une demande d'appartenance générale.

6.3 Gestion des groupes entre routeurs multicast

Les routeurs multicast doivent dialoguer aussi entre eux pour gérer dynamiquement ces groupes et maintenir à jour les tables de routage pour les adresses multicast.

La figure 4 nous donne un exemple de ce qui peut se passer avec un groupe. Nous avons un groupe G et trois machines M, P et Q membres.
Les routeurs A, B, C et D traitent le multicast ou sont reliés par des tunnels.
Aucune des machines en dessous du routeur D n'a déclaré appartenir à ce groupe.

La machine M envoie un paquet (trajet en pointillé). Celui-ci va passer par le routeur A, puis le B.
A ce niveau-là il ne va être transmis qu'au routeur C car B a la connaissance du fait qu'aucune machine ne fait partie du groupe après le routeur D. Puis finalement le paquet est délivré aux machines P et Q membres du groupe.

Il faut pour que cela puisse fonctionner ainsi, que les routeurs s'échangent des informations leur permettant de maintenir à jour les tables de routages pour les adresses multicasts.

Ces tables de routage multicast contiennent une liste (@groupe, liste des routeurs voisins). Plusieurs protocoles permettant de remplir ces informations existent :
Le protocole de diffusion de groupe implémenté dans les routeurs que vous allez utiliser

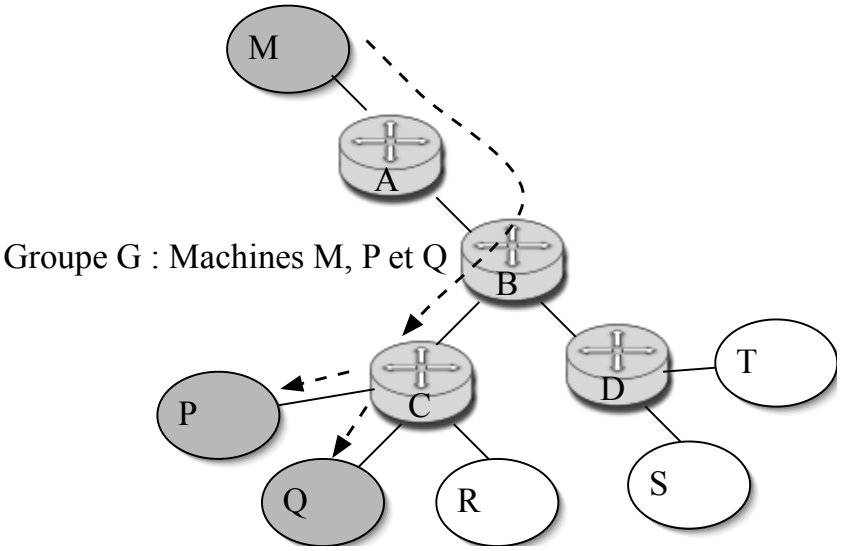

FIGURE 4 – Emission d'un paquet vers un groupe

(Cisco) est PIM (Protocol Independant Multicast).
Celui utilisé dans les routeurs du MBONE est DVMRP (Distant Vector Multicast Routing Protocol). Il n'est pas implémenté dans les routeurs Cisco, mais il est possible de créer un tunnel d'un routeur PIM vers un routeur DVRMP.

Ces différents protocoles reposent sur l'un ou l'autre des deux principes suivants :

- **Le *spanning tree* (arbre minimal recouvrant un graphe quelconque)** :
  Pour chaque groupe, un arbre de recouvrement minimal est construit et maintenu pour l'ensemble des routeurs du réseau. La liste des voisins à qui envoyer un paquet multicast est alors la liste des voisins dans l'arbre. Cet arbre est un sous-ensemble d'un arbre unique pour tous les flux multicast . Pour des questions de performance, cette technique est utilisée en fixant la racine de l'arbre (appelé RP : « *point de rendez vous* »).

- **Arbre déterminé par la source**
  Chaque routeur construit un arbre minimal au moment de l'émission des paquets de donnée multicast vers un groupe donné. Il y a *inondation* du réseau par les 1^{er} paquets multicast puis *élagage* si possible de l'arbre maximal. Les arbres obtenus sont donc différents pour chaque source de paquet multicast. Chaque routeur connaît l'ensemble de ces voisins amonts et avals dans l'arbre mais aucun (même pas la source) ne connaît vraiment la topologie de l'arbre dans son ensemble.

Dans le protocole PIM, il existe deux modes :

- **Dense :** beaucoup de routeurs sont reliés à des réseaux susceptibles d'avoir des membres d'un groupe. C'est la méthode basée sur les arbres depuis la source et le RPF (Reverse Path Forwarding) qui est utilisée.

- **Sparse :** peu de routeurs sont reliés à des réseaux susceptibles d'avoir des membres

d'un groupe. C'est la méthode d'arbre recouvrant avec point de rendez-vous qui est utilisée. L'arbre recouvrant (pour un groupe) est déterminé à partir de la table de routage Unicast. Tous les flots de données multicast seront donc concentrés sur les branches de cet arbre. C'est une solution peu coûteuse en terme de paquet « de gestion d'arbre » mais qui est beaucoup moins bonne si les flots multicast s'avèrent important. Tous les flux multicast passent par le point de rendez-vous.

Dans le protocole DVRMP, la méthode utilisée est la même (avec quelques nuances) que celle du mode dense de PIM (inondation puis élagage).

Dans ce TP, nous étudierons le mode **Dense** de PIM.

6.4 Principe du protocole PIM Dense

- **Conditions de fonctionnement** :
  Il faut tout d'abord que les tables de routages « Unicast » soient correctement remplies dans les routeur multicast. Il faut donc soit les remplir « à la main », soit qu'un algorithme de routage Unicast (RIP, OSPF ...) soit activé.
  Les tables de routages Unicast contiennent donc pour toutes les adresses de routeurs (susceptibles d'être rattachés à un réseau possédant un membre d'un groupe), l'adresse du routeur voisin permettant d'y accéder de « la meilleur façon possible ».

- **Diffusion générale et construction de l'arbre complet :**
  La définition de l'arbre des routeurs associé à un groupe se fait tout d'abord depuis la source, au moment de l'envoi d'un paquet multicast de donnée (comportant une adresse source **@S** et une adresse de groupe **@G**).
  Ce premier paquet de donnée multicast est diffusé par inondation (en broadcast) dans tous le réseau.
  A l'arrivée du paquet mutlicast, la relation (@S,@G, listes des autres voisins) est mémorisée dans la table de routage multicast. Le paquet de donnée mutlicast est ensuite re-diffusé sur toutes les autres interfaces du routeur.
  Dans l'exemple ci dessous B va noter (@A, @GROUPE, (A, D, E, C)) et réenvoyer le paquet vers D, E et C ; A la réception de ce paquet E va noter (@A, @GROUPE, (B, D, F)), etc

  Pour limiter le nombre d'émission de cette inondation (et créer un arbre de diffusion), les routeurs ne re-émettent un paquet de donnée multicast que s'il arrive par le chemin noté dans la table de routage Unicast (l'adresse du voisin qui a envoyé le paquet de source @S est dans la table de routage pour @S). Ce principe astucieux est appelé le **Reverse Path Forwarding (RPF).** Dans la figure 5 les paquets représentés par des flèches « barrées », ne sont pas réémis lors de l'inondation depuis A.

- **Principe d'élagage**
  Afin que les routeurs ne puissent pas recevoir de paquet de plusieurs voisins de l'arbre complet, on ajoute un mécanisme supplémentaire : l'élagage.
  L'élagage d'une branche se produit dans deux cas :

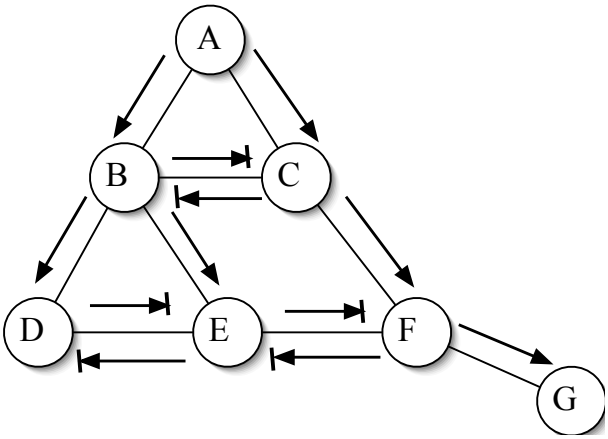

FIGURE 5 – Limitation de l'inondation

1. **Elagage dû au RPF :**
   Un routeur recevant un paquet de donnée multicast sur une interface (depuis un routeur voisin x) qui n'est pas donnée comme le plus court chemin à la source envoie un paquet spécial d'élagage au voisin x. A la réception de ce paquet d'élagage le routeur supprime de la liste des routeurs avals (dans l'arbre de diffusion) le routeur émetteur du paquet d'élagage.
   Dans la figure 5, toutes les flèches « barrées » correspondent à l'émission d'un paquet d'élagage : par exemple C envoit un paquet d'élagage à B, de même D et F envoient un paquet d'élagage à E.
   Donc on a ensuite :
   - dans la table de B : (@A, @GROUPE, (A, D, E))
   - dans la table de E : (@A, @GROUPE, (B)).

2. **Elagage par absence de machine appartenant au groupe :**
   Un routeur (recevant un paquet multicast) qui n'a pas de voisin à qui diffuser et qui ne possède pas de membre pour le groupe concerné (dans les réseaux qui lui sont rattachés) est donc une feuille de l'arbre. Il ne faut donc plus que le routeur amont lui envoie de paquet pour ce groupe. Dans ce cas, il émet un paquet d'élagage au routeur amont dans l'arbre. Dans la figure 5 si G n'a pas de membre rattaché au groupe visé, il émettra un paquet d'élagage vers F.

Ce processus d'élagage est récursif, un routeur devenant feuille par élagage de toutes ses branches avales, envoie alors un paquet d'élagage au routeur amont.
Par exemple si F reçoit un paquet d'élagage de G et n'a pas non plus de membre rattaché au groupe visé, il émettra un paquet d'élagage vers C.
A la disparition de tous les membres d'un groupe dans un réseau, l'arbre construit pour cette source doit être supprimé dans chaque routeur. Deux solutions sont envisageables : des timers de non-utilisation dans les routeurs, l'émission d'un paquet de suppression dans l'arbre par la source.

- **Principe de greffage** Un autre processus doit être mis en place dans le cas où

un nouveau membre d'un groupe apparaît sur une branche qui a été précédemment élaguée. Deux méthodes complémentaires résolvent ce problème :

- Un timer de re-greffage automatique est lancé au moment de l'élagage. A l'expiration de ce timer, un re-greffage est effectué automatiquement. Ce regreffage est inutile pour le 1^e cas d'élagage précédemment cité.
- Un routeur pour lequel apparaît un nouveau membre à un groupe envoie un paquet de greffage aux routeurs amonts pour tous les arbres dans lesquels des élagages ont eu lieu. Une branche élaguée doit donc être marquée « élaguée ». Un routeur doit savoir, si il a été précédemment élagué d'un groupe donné. Sinon ce n'est pas la peine qu'il essaie de se regreffer.
- Le processus de greffage est aussi récursif.

Il est à noter que ce processus d'élagage/greffage est effectué pour chaque arbre issu d'une source de paquet multicast différente.

- **Format d'un paquet PIM** : (dans l'ordre)

  - Pim Version (sur 4 bits) : (normalement 2)
  - Pim Type (sur 4 bits) :
    - Hello (ou Query en Pim v2)(permet de tester périodiquement la présence d'un voisin dans un arbre) : 0
    - Join (mode Parse) / Elagage(mode dense) : 3
    - Greffage : 6
    - Ack de greffage : 7
  - Checksum (sur 8 bits)

  La suite dépend du type, on devrait retrouver l'adresse de groupe que concerne le paquet.

  Les paquets PIM sont émis (sur les réseaux à diffusion) avec l'adresse destination spéciale 224.0.0.13 (routeurs traitants le multicast).
  Pour plus de détails voir la RFC : $http://tools.ietf.org/html/rfc3973$

- **Exemple commentée de table de routage multicast :**

  Router# show ip mroute
  IP Multicast Routing Table
  Flags : D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
  R - RP-bit set, F - Register flag, T - SPT-bit set
  Timers : Uptime/Expires
  Interface state : Interface, Next-Hop, State/Mode
  (*, 224.0.255.3), uptime 5 :29 :15, RP is 0.0.0.0, flags : DJCL
  Incoming interface : Ethernet1, RPF neighbor 0.0.0.0,
  Outgoing interface list :
  Ethernet0, Forward/dense, 5 :29 :15/0 :02 :57
  (198.92.46.0/24, 224.0.255.3), uptime 5 :29 :15, expires 0 :02 :59, flags : C

Incoming interface : Ethernet1, RPF neighbor 0.0.0.0,
Outgoing interface list :
Ethernet0, Forward/dense, 5 :29 :15/0 :02 :57

- (*,224.0.255.3) indique l'entrée de la table de routage pour le groupe 224.0.255.3, * indique tous les machines sources.
- Incoming interface : indique l'nterface par laquelle les paquets multicast doivent arrivés pour être pris en compte (sinon ils sont détruits).
- Outgoing interface list : indique les interfaces sur lesquelles les paquets multicast doivent être re-émis à destination du groupe en question.
- Dans l'exemple précédent, un paquet à destination du groupe 224.0.255.3 provenant de l'interface Ethernet1 doit être réemis vers l'interface Ethernet0.
- Remarque : les lignes (@S,@G) sont construites à partir de la ligne dont la source est marqué par une * (en fonction du RPF).

- **Administration des switchs :**
  - Connexion par un câble série au port com1 du PC
  - Commande sur le PC : *minicom*
    Puis login : *manager* et password : *friend*
  - Retour à une configuration « minimale » :
    Dans Menu « *System Config* » : « *Reset Factory Defaults* »

7 Expérimentations

ATTENTION : Les routeurs 2500 ne traitent pas le multicast, utilisez les 2600 ou 892.

Conseils : Mettez au départ les routeurs dans une configuration minimale :
erase startup-config puis *reload* (à la question « . . .initial configuration dialog » réponde no)
De même mettez le Switch « administrable » dans une configuration « minimale ».

Configurez la plate-forme comme indiquée sur la figure 6.

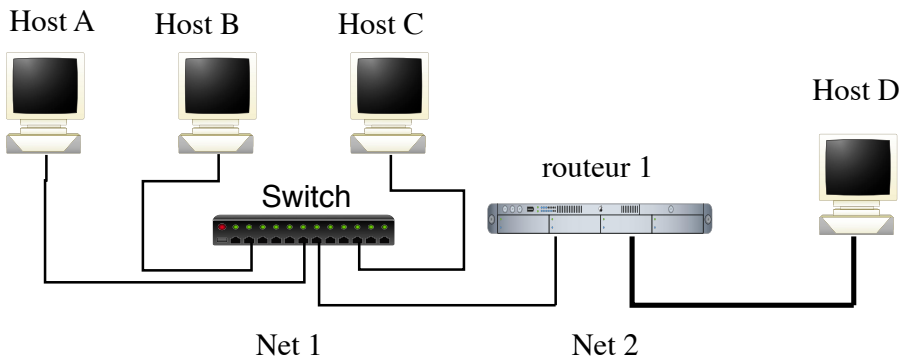

FIGURE 6 – Plateforme d'expérimentation

Donnez des adresses IPv4 à vos machines et aux interfaces du routeur afin qu'elles puissent toutes communiquer entre elles.
Testez à l'aide de ping.

7.1 Diffusion sur un réseau local

Lancer " socklab udp " sur les stations hostA, hostB et hostC.

1. Sur hostA et hostB, créez (**msocket**) des sockets multicast. Associez un numéro de port local à cette socket (**mbind**).
   Attention ce numéro de port devra être le même pour toutes les sockets d'un groupe donné.
2. Joignez les (**mjoin**) à un groupe multicast, dont vous aurez choisi l'adresse de classe D dans l'intervalle 224.0.1.0 - 239.255.255.255 (224.0.0.0 à 224.0.0.255 étant réservées au routage).
3. Mettez ensuite les stations en réception de paquets multicast (**mrecv**)
   Sur hostA, lancez Wireshark.
4. Sur hostC, créez une socket multicast (**msocket**). Associez un numéro de port local à cette socket (**mbind**) puis émettez (**msend**) un paquet à destination du groupe

muticast constitué par les stations hostA et hostB.

Remarque : lors du **msend** il est demandé de préciser l'adresse source du paquet au cas ou la machine possèderait plusieurs interfaces.
ATTENTION Suivant les versions de BSD, il peut être nécessaire qu'une route par défaut existe sur la machine (bug du noyau BSD).

5. Après cette émission, vérifiez tout d'abord que les stations du groupe ont effectivement reçu le paquet émis, puis analyser ce qu'il s'est passé sur le réseau.
6. Regardez les adresses Ethernet des paquets « multicast ». Expliquez comment est réalisé l'adressage multicast au niveau Ethernet.
7. Recommencez l'expérience en utilisant une socket non multicast côté émetteur, cela fonctionne t-il encore ? pourquoi ?

7.2 Utilisation d'un routeur multicast

1. **Ajout d'un membre à un groupe**
   (a) Configurez le routeur pour qu'il gère les paquets multicast et utilise le protocole IGMP (Internet Group Multicast Protocol).

   (b) En mode configuration terminal :
   - **ip multicast-routing**
     Puis en mode configuration interface (sur les 2 interfaces) :
     - **ip pim dense**
       Cette commande lance le démon du protocole IGMP (version2) et du protocole PIM sur chaque interface.
     - On peut vérifier l'état des interfaces concernant ces protocoles par : **sh ip igmp interface.**

   (c) Refaites l'expérience précédente en observant le réseau lors des *joins* sur hostA et hostB. A quoi servent les paquets circulant à ce moment-là ?

   (d) Lors d'une émission vers le groupe (hostA, hostB), est ce que le paquet circule aussi sur le réseau net2 ?

   (e) Regarder et expliquez la table de routage concernant le multicast sur le routeur (**sh ip mroute**). On peut avoir la liste des groupes connus du routeur par **sh ip igmp groups**.
   Remarque : Le groupe 224.0.1.40 est réservé à la découverte de point de rendez vous de PIM mode parse. On n'en tiendra pas compte ici.

   (f) Joignez le hostD au groupe précédemment créé entre hostA et hostB.
   Que se passe t il sur le réseau net2 ?

   (g) Lors d'une émission vers le groupe (A, B, D), est-ce que le paquet circule aussi sur le réseau net2 ?

(h) Vérifiez le champ TTL de l'entête IP des paquets multicast.

(i) Observez à nouveau la table de routage multicast du routeur. Expliquez précisément son contenu.

2. **Suppression d'un membre à un groupe**

(a) Sur hostA enlever la socket du groupe auquel elle appartient (**mleave**). Observez ce qu'il se passe alors sur le réseau.

(b) Observez la table de routage multicast du routeur. Comment le routeur sait-il s'il existe encore des machines hôtes appartenant à un groupe donné ?
Quel est le dialogue effectué par un hôte et le routeur lorsque :
- un hôte se joint à un groupe
- un hôte fait partie d'un groupe
- un hôte quitte un groupe

(c) Essayez d'estimer le timer de « demande d'appartenance » utilisé dans les routeurs.

(d) Au vue des différents paquets IGMP circulant sur les réseaux pendant ces expérimentations donnez l'algorithme du protocole IGMP dans un hôte et dans un routeur.

(e) Expliquez comment la table de routage multicast est construite en fonction des informations données par le protocole IGMP.

7.3 Gestion du multicast par les switchs

Les switchs que vous utilisez peuvent gérer le Multicast. C'est à dire qu'ils sont capable de connaître les groupes en cours et les machines appartenant à ces groupes. Ce type de switch analyse donc les paquets IGMP et filtre la propagation des paquets multicasts.

Dans les expérimentations que vous venez d'effectuer ce n'était pas le cas, les paquets multicast arrivent à toutes les machines du réseau Ethernet. Vérifiez le.

Configuration du switch multicast
- Menu *System Config*
- Puis A *Advanced Config*
- Puis *IGMP Snooping Config*
- Puis *IGMP Snooping Status : Enable*

Refaites une expérience permettant de vérifier que le switch *filtre* les paquets mutlicast. C'est à dire qu'il ne fait plus du *broadcast* en cas d'adresse multicast.
On peut observer la table multicast du switch par : « *view multicast hosts list* »

A quel moment cette table est elle remplie ou modifiée ?

Conclusions.

7.3.1 Gestion du multicast dans le switch lors de l'apparition d'un routeur multicast

1. Comment le switch connaît il le port sur lequel est branché le routeur multicast ? Quel problème cela poserait il si il n'avait pas cette connaissance ?

2. Quels sont les changements (informations multicast au niveau du routeur en particulier) par rapport à la configuration précédente (où le switch ne gère pas le multicast) ?

3. Est ce le routeur ou le switch qui effectue « les demandes d'appartenance » ?

4. Quel intérêt ?

Normalement vous arrivez par là à la fin de la 1ère séance....

7.4 Propagation des groupes entre routeurs

Configurez la plate-forme comme indiquée sur la figure 7.
Attention utilisez sur Net4 un Hub et non un switch pour pouvoir capturer sur C tous les paquets circulant sur Net4

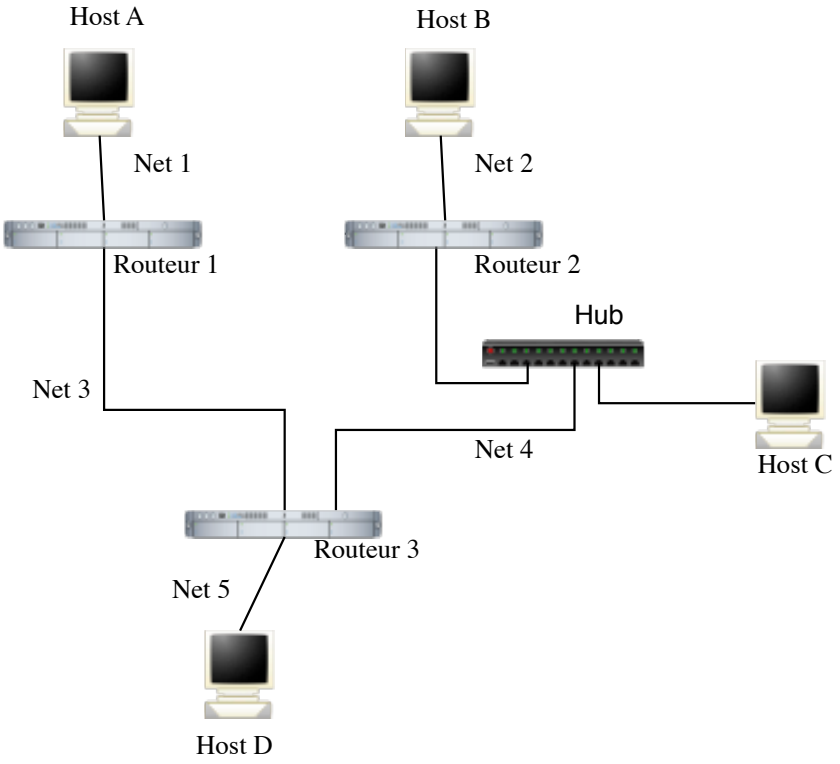

FIGURE 7 – Plateforme d'expérimentation pour l'étude de PIM

On pourra repartir de la configuration précédente pour gagner du temps.

1. Lancer RIP sur les hôtes (***routed [-q|-s] -P ripv2 -P no_rdisc***) et sur chaque interface des 3 routeurs .
2. Lancer PIM sur les interfaces des routeurs.
   Observez les paquets circulant sur le réseau. Les routeurs sont-ils conscients de leur présence réciproque ?
3. Effectuer en utilisant Socklab sur D, l'émission d'un paquet de donnée multicast vers un groupe G ne possédant aucun membre. Capturez les paquets circulant alors sur net1, net2 et net4.

   Pourquoi le paquet multicast de donnée circule sur **net4** (et sur **net3**) et non sur **net1** et **net2**. Quels sont les paquets PIM qui ont circulé sur net4 ?
4. Faîtes une figure similaire à la 5 montrant les paquets d'inondation circulant sur le réseau et les paquets d'élagage dû au RPF et ceux dû à la non existence d'un membre au groupe en jeu.
5. Observez (sans trop attendre) la table de routage multicast des 3 routeurs. Quelles sont les lignes qui sont marquées élaguées (*Pruned*) ? Pourquoi ?
   Attention le timer de greffage automatique est assez court (˜30s).
6. Refaites l'expérience précédente en ayant au préalable joint A au groupe G. Quels changements dans les tables de routages multicast cela induit-il ? Refaire une figure. Expliquez.
7. Sur B à l'aide de socklab, joignez vous au groupe G dont l'arbre vient d'être créé depuis D.

   Est ce que cela engendre un paquet PIM de greffage ?
   Envoyez à nouveau un paquet multicast à destination de G. Arrive t-il à B ?

   Attention dans PIM l'état élaguée" des branches précédemment élaguées (mais mémorisées) est supprimé sur timer. C'est équivalent à un greffage mais sans paquet PIM. Donc au bout de ce timer, si on reenvoie un paquet vers le groupe il arrive de nouveau aux routeurs feuilles.
8. Sur B, quittez le groupe G.
   - Est ce que des paquets IGMP circulent au moment du ***mleave*** ? Pourquoi ?
   - Est ce que des paquets PIM circulent au moment du ***mleave*** ? Pourquoi ?
   - Quand est effectué l'élagage dans ce cas ?
9. Sur A, quittez le groupe G.
   - A quel moment et comment un arbre est-il complètement détruit ?
   - Il y a t-il un paquet spécial pour cela ? Est ce que des timers sont utilisés pour cela ?

A partir de ces observations donnez un résumé de l'algorithme du protocole PIM en mode « dense ».