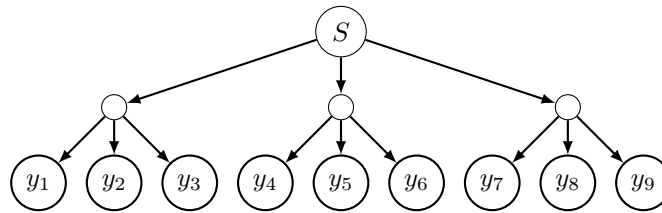


Protocoles réseaux

TD n° 10 : Multicast

Exercice 1 :

On considère le réseau de distribution ci-dessous. Le nœud S à la racine de l'arbre distribue des paquets à toutes les feuilles y_1, \dots, y_9 . Dans cet exercice, on définit le *coût de la distribution* comme étant le nombre de liens traversés par des paquets (si on envoie deux fois le même paquet sur un lien, on le compte deux fois).



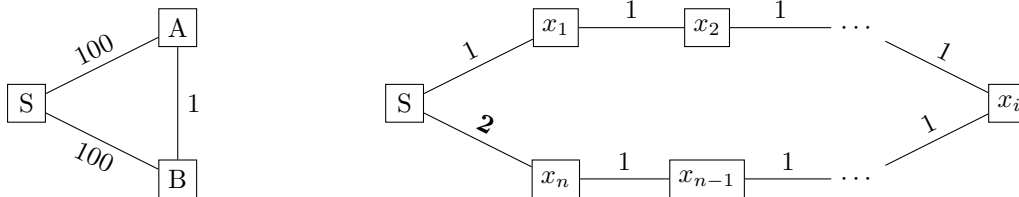
1. Si S fait de l'unicast, quel est coût total pour envoyer une copie d'un paquet à chaque feuille ?
2. Si S fait du multicast, quel est coût total pour envoyer une copie d'un paquet à chaque feuille ?
3. Refaire les calculs lorsque le réseau est maintenant un arbre de hauteur h (l'arbre ci-dessus est de hauteur 2) dont chaque nœud intermédiaire a 3 fils. On exprimera le résultat en fonction de n , le nombre de feuilles. Conclusion ?

S distribue maintenant un flux vidéo d'un débit constant de 5 Mb/s.

4. Quel est le débit maximal sur un lien, dans le cas de l'unicast et du multicast ?
5. Le nœud S est relié à l'Internet par de la fibre à 1Gb/s. Combien de clients peut-il desservir ?

Exercice 2 :

On considère les deux réseaux suivants, où S envoie des paquets en multicast à tous les autres nœuds. La métrique utilisée est un *coût* (plus ou moins financier) à minimiser. On veut construire un **arbre de distribution**, qui est un arbre couvrant (il touche tous les nœuds) enraciné en S .



On considère un arbre¹ de distribution construit à partir des **plus courts chemins** entre S et chaque nœud, par un algorithme tel que Dijkstra.

1. Calculez un tel arbre sur les deux réseaux ci-dessus. Que peut-on dire ?

On propose maintenant comme arbre de distribution un Arbre Couvrant Minimum² tel qu'obtenu par les algorithmes de Prim ou Kruskal (la somme des poids des arêtes de l'arbre est minimum).

2. Calculez un arbre couvrant minimal sur les deux réseaux ci-dessus. Que peut-on dire ?
3. Conclusion ?

1. pas forcément unique

2. pas forcément unique non plus

Exercice 3 :

Lorsqu'une application tournant sur l'hôte H s'abonne au groupe multicast G , H envoie un message $join(G)$ du protocole IGMP aux routeurs voisins. Lorsque plus aucune application sur un hôte n'est abonnée au groupe G , le hôte envoie un message IGMP $leave(G)$.

1. À quoi sert IGMP ?
2. Le message IGMP est envoyé en multicast ; on implémente donc le multicast à l'aide du multicast. Est-ce un problème ?
3. Que se passe-t-il si un hôte quitte le réseau avant d'avoir envoyé son *leave*, ou si le *leave* est perdu ? Comment imaginez-vous que IGMP résoud ce problème ?

Exercice 4 :

Pour chaque source de trafic S du réseau et pour chaque groupe multicast G , un *protocole à vecteur de distances multicast* naïf construit l'arbre enraciné en S (arbre entrant ou *intree*) des chemins les plus courts depuis S vers tous les autres routeurs du réseau à l'aide de l'algorithme à vecteurs de distances. Le plan de données obéit à l'algorithme *reverse path forwarding* (RPF) : lorsqu'un paquet avec pour source S et destination G arrive sur l'interface I d'un routeur R ,

- si l'interface I n'est pas l'interface par laquelle passe l'arbre de plus court chemin de S à R , alors le paquet est jeté et il n'y a plus rien à faire ;
- sinon, R duplique le paquet et l'émet à travers toutes les interfaces autres que I .

1. Que se passerait-il si les paquets qui arrivent par d'autres interfaces étaient propagés ?
2. Que doit contenir l'annonce d'un routeur R (relative à S et G) ? Que doit-on rajouter par-rapport au vecteur de distances unicast habituel ?
3. On suppose qu'il y a n routeurs dans le réseau et g groupes multicast. Quelle quantité d'état est stockée dans les routeurs du réseau ?
4. Quels sont les routeurs atteints par un paquet destiné à G dans cet algorithme naïf ? Proposez une solution à ce problème.

Exercice 5 :

On considère l'arbre de distribution de l'exercice 1. On suppose qu'on utilise le protocole d'émission 1-persistant suivant : chaque paquet P est émis par la source en multicast, et chacune des destinations d envoie un acquittement pour chaque paquet reçu. Pour chaque paire (d, P) , si un acquittement (pour P venant de d) n'est pas reçu depuis, la source réémet (une seule fois) le paquet P en unicast à d .

1. Ce protocole est-il fiable ? Est-il possible de le rendre fiable ?

On suppose que le taux de perte, c'est à dire la probabilité qu'un nœud ne reçoive pas un paquet, est p aussi bien pour le multicast que pour l'unicast³ et on suppose que les pertes sont indépendantes. Pour simplifier le calcul, on suppose que les ACK/NACK ne sont jamais perdus.

2. Calculez le coût (en moyenne) du protocole. Le coût dépend-il du taux de perte ? Conclusion ?

On considère maintenant un protocole basé sur les acquittements négatifs (NACK) : si la destination d reçoit un paquet P , elle n'envoie rien. Si par contre la destination ne reçoit pas le paquet P , elle envoie un NACK à la source. À chaque fois qu'elle reçoit un NACK, la source réémet le paquet en unicast vers d .

3. Ce protocole est-il fiable ? La modification que vous avez proposé plus haut marche-t-elle ?
4. Comment fait-on pour détecter si un paquet est perdu ? Que se passe-t-il en cas de déséquence des paquets par le réseau ?
5. Calculez le coût (en moyenne) du protocole. Comparez au protocole précédent.
6. Quelle est la latence maximale ajoutée par les réémissions de ces deux protocoles ? On supposera pour simplifier que les paquets multicast sont envoyés à intervalle régulier T .

3. Ce qui n'est sûrement pas réaliste. Je n'en sais ma foi rien.