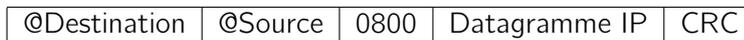


TD3 – Réseaux couche physique
 {Emmanuel.Godard, Yann.Esposito}@lif.univ-mrs.fr
 27 octobre 2005

☞ Internet peut-être vu comme un réseau de sous-réseaux hétérogènes. Le ciment qui permet à tous ces réseaux de coopérer, c'est le *protocole IP*. Une question fondamentale des communications entre réseaux est le **routage**. Il faut trouver un moyen de trouver un chemin entre deux sous-réseaux. Nous allons étudier les problèmes liés au routage et quelques solutions existantes.

0.1. Quelle est la différence entre *paquet* et *datagramme* ?

paquet : mode connecté comme par exemple une trame ethernet



datagramme : mode non connecté.

1 Routage

1.1. Supposons qu'un ordinateur A veuille communiquer avec un ordinateur B. L'adresse de A est 147.64.94.3, le masque du sous-réseau 255.255.255.0. A et B appartiennent tout deux à des sous-réseaux ethernet.

(a) Décrivez quelle trame ethernet A va envoyer et comment il la construit si l'adresse de B est 147.64.94.5.

D'abord A vérifie que B appartient au même sous-réseau :

$$147.64.94.3 \ \& \ 255.255.255.0 = 147.64.94.0$$

$$147.64.94.5 \ \& \ 255.255.255.0 = 147.64.94.0$$

Donc A sait que B appartient au même sous-réseau. Il vérifie alors si il possède l'adresse MAC de B dans sa table, si ce n'est pas le cas, il va faire une requête ARP. Une fois l'adresse MAC de B obtenue (@B) A envoie la trame



(b) Décrivez en détail les évènements qui se produisent lorsque l'adresse de B est 167.131.50.6.

A vérifie le réseau auquel appartient B. Il sait donc qu'il ne peut pas faire un envoie direct.

A possède au moins une adresse de routeur R. Il **encapsule** le datagramme IP dans la trame ethernet qu'il envoie au routeur de son sous-réseau.



R procède de la même manière que A

...

Jusqu'à arriver à un routeur F qui est connecté au sous-réseau de B. Ce dernier communique alors le datagramme IP à B via une encapsulation ethernet.

2 algorithmes de routage

2.1. Expliquez selon quels critères on peut classer un algorithme de routage. Donnez une explication de comment un algorithme peut améliorer chacun de ces critères.

Efficacité : l'algorithme trouve bien le meilleur chemin. Pour cela il faut avoir de nombreuses informations comme le nombre de noeuds, l'encombrement, la fiabilité...

Simplicité : la complexité de l'algorithme doit être la plus simple possible. Un façon est de ne considérer qu'un seul type de distance. En cela ce critère s'oppose au critère d'efficacité.

volume de trafic : l'algorithme doit généré peut de trafic sur le réseau. Par exemple en faisant des mises à jour de tables plutôt que des transferts entiers de tables.

convergence : vitesse à laquelle les routeurs apprennent l'indisponibilité ou plus généralement le changement de poids d'une arête d'un chemin.

robustesse : résistance de l'algorithme à des conditions anormales comme la surcharge ou la panne d'un routeur.

justice : il ne doit pas y avoir de sous-réseau isolé par ce que d'autres saturent certaines lignes.

2.2. Routage statique Soit deux routeurs R1 et R2. Le routeur R1 est connecté aux réseaux 147.64.94.0 avec l'adresse 147.64.94.1, 147.65.35.0 avec l'adresse 147.65.35.6. R2 est connecté aux réseaux 147.65.35.0 avec l'adresse 147.65.35.9, 167.131.0.0 avec l'adresse 167.131.1.14

(a) Donnez les tables de routages de R1 et R2 qu'un administrateur réseau choisirait.

R1		R2	
Réseau	Next hop	Réseau	Next hop
147.64.94.0	–	147.64.94.0	147.65.35.6
147.65.35.0	–	147.65.35.0	–
167.131.0.0	147.65.35.9	167.131.0.0	–

(b) Donnez une évaluation de ce type de routage en fonction des critères vu précédemment.

Efficacité : maximale si le réseau est stable peut être très mauvaise en cas de problème.

Simplicité : Maximale, pas d'algorithme.

volume de trafic : optimal, pas de communication entre routeurs.

convergence : on ne peut pas faire pire, c'est un administrateur qui doit s'en rendre compte.

robustesse : très faible, la présence de l'administrateur est requise au moindre changement de condition.

justice : bonne si l'administrateur possède une bonne idée de la topologie du réseau et sait à peu près à l'avance quels seront les ordinateurs qui génèrent le plus de trafic.

2.3. . Routage par inondation. Expliquez le fonctionnement. Comment éviter la duplication infinie des datagrammes? Évaluez ce type de routage.

Tous les routeurs envoient les paquets sur tous les sous-réseaux sauf sur le réseau d'entrée du paquet. Pour éviter la duplication infinie, il faut mettre en oeuvre soit un système de vieillissement des paquets, soit que chaque routeur se souviennent des paquets qu'il a déjà transmis pour ne plus les renvoyer.

Efficacité : maximale puisque l'on emprunte tous les chemins le meilleur est toujours trouvé.

Simplicité : très grande.

volume de trafic : maximal, le trafic est maximisé, la saturation devient donc plus courante.

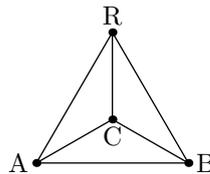
convergence : optimale.

robustesse : optimale, quelque-soit la nouvelle topologie du système, tout continue de marcher. Le seul problème provient de la saturation du réseau.

justice : optimale.

3 Mises à jour locales

3.1. Donnez les tables de routages de A,B et C du réseau suivant lorsqu'on considère que la distance est 1 par saut.



Décrivez ce qui se produit si R tombe en panne avec un algorithme de mises à jour locales ? C'est-à-dire que chaque routeur ne communique à son voisin direct que les changements dans sa table de routage.

Au départ $AR=BR=CR=1$. Lorsque R tombe en panne, A met à jour sa distance et en passant par B il est à une distance 2. B fait de même en passant par C. C fait de même en passant par A. un cas où $AR=BR=CR$. Les distances montent alors lentement vers ∞ et si A, B et C continuent de prendre de mauvaises décisions, les paquets entrent dans une boucle et encombrerent le réseau.

(a) En particulier déduisez en l'intérêt de l'information TTL (Time To Live) situé dans les datagrammes IP.

Si un paquet est perdu (il entre dans une boucle) alors le TTL permet de détruire ces paquets nuisibles qui encombrerent le réseau.

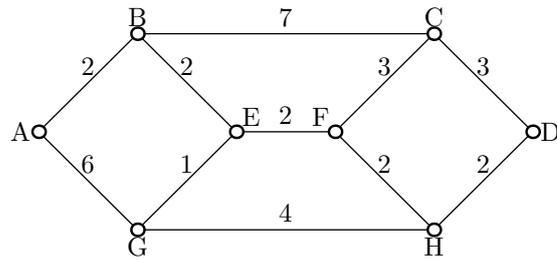
4 Mises à jour globales

4.1. Déduisez de la question précédente une méthode globale de routage. Donnez en les avantages et les inconvénients par rapport à la méthode précédente.

Il suffit d'imaginer un serveur unique qui récupère les tables de routage de tous les routeurs. Ce serveur connaît donc la topologie complète du réseau à un instant (court) donné. Il calcule les meilleurs chemins et envoie les mises à jour des tables de routages à tous les routeurs.

Attention, il ne faut pas oublier d'inclure une information temporelle pour éviter qu'une vieille mise à jour soit acceptée plutôt qu'une récente le parcours des paquets de mise à jour n'étant pas toujours identique.

4.2. Calculez les plus courts chemins à partir de A avec l'algorithme de Dijkstra.



4.3. Donnez l'arbre de recouvrement en partant de A qui correspondants aux chemins les plus courts partant de A.

