Protocoles réseaux
TD n° 10 : Diagrammes temporels et protocole ND pour IPv6

I) Lien avec fenêtre de taille 1

Dans cette section, on se place au niveau de la couche lien entre deux machines. Ce lien bidirectionnel peut *perdre* des paquets : un paquet envoyé soit n'arrive pas de tout (perte), soit arrive correctement (pas d'erreur, ni de création de paquet). On suppose ce lien suffisament court pour que ne pas plus d'un paquet en vol (fenêtre de 1 paquet) soit pertinent.

On suppose que la couche supérieure de la machine A doit transmettre à celle de la machine B un flux de données (suite d'octets sans fin, une éventuelle structuration de ce flux étant le travail de cette couche supérieure). Au niveau de la couche lien, on transmet des paquets de 1 ko maximum. Le lien est *synchrone* de la façon suivante : un paquet envoyé est soit reçu 1ms plus tard au maximum, soit perdu.

On dit que la transmission est *fiable* si pour toute donnée émise par A :
- B reçoit la donnée correctement,
- ou A reçoit une indication d'erreur

Exercice 1 : acquittement simple

On considère le protocole suivant. Les messages sont de deux types :
- Données. Entête constituée du caractère `'D'`, puis de la longueur du message (sur 2 octets) suivi des données elles-même (un segment d'au plus 1021 octets du flux)
- Acquittement. Message constitué d'un seul octet : `'A'`.

Le protocole est ainsi constitué :
- A envoie un paquet de données (au plus 1021 octets pris dans le flux) à B
- À chaque paquet de données reçu, B enfile son contenu dans son flux et envoie un acquittement
- A attend un des deux événements suivants
  - Soit un timeout a lieu : au bout de 2ms, A suspecte une perte et renvoie *le même* paquet de données que précédemment
  - Soit un acquittement est reçu : A pourra alors envoyer le segment *suivant* de données

Montrer que ce protocole n'assure pas une transmission fiable du flux de données.

Exercice 2 : bit alterné

On considère maintenant quatre types de messages : deux types de messages de données, `'D'` et `'d'`, et deux types d'acquittements `'A'` et `'a'`.

1. Spécifier un protocole utilisant judicieusement ces quatre types (A n'envoie que des données et B que des acquittements).
2. Montrer qu'il transmet de façon fiable le flux.
3. Pourquoi n'est-il pas utilisé en couche 4 par des protocoles tels que TCP ?

Exercice 3 : réseau asynchrone

On enlève l'hypothèse qu'un paquet non reçu au bout de 1ms est nécéssairement perdu.

1. Montrer que les protocoles des deux exercices précédents n'assurent pas une transmission fiable.
2. Rajouter une hypothèse pour que le protocole de l'exercice 2 devienne correct. Cette hypothèse ne doit pas être un délai maximum au-delà duquel on peut supposer quelque chose (le canal est *asynchrone*) ni une certitude que certains paquets arrivent (le canal est *avec pertes*).
3. Montrer que le protocole de transmission est fiable sous cette hypothèse.

II) Le protocole Neighbor Discovery pour IPv6

Le protocole Neighbor Discovery (RFC 4861) est un protocole qui en fait réunit trois protocoles distincts :

- un protocole qui permet à un hôte de découvrir les routeurs résidant sur le même lien que lui,
- un protocole qui permet à un hôte de déterminer les adresses de couche lien des voisins connus résidant sur le même lien que lui (équivalent du protocole ARP en IPv4),
- un protocole de redirection ; lorsque l'hôte envoie un paquet à son routeur par défaut, si ce dernier s'aperçoit que l'hôte a une meilleure route qui ne passerait pas par lui, il envoie le paquet vers sa destination et en parallèle envoie une notification à l'hôte pour lui annoncer cette meilleure route. Ainsi l'hôte pourra envoyer les paquets suivants pour la même destination par là.

Exercice 4 : accessibilité d'un voisin

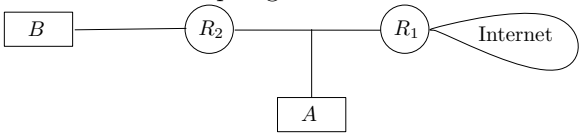
En IPv6, le protocole ARP d'IPv4 est remplacé par le protocole de découverte de voisins (ND), décrit dans la RFC 4861. Ce protocole est plus raffiné que ARP, et permet notamment de découvrir qu'un voisin est devenu inaccessible, ce qui permet par exemple de choisir un routeur différent. Les entrées du cache de voisins de ND sont équipées d'un champ *état*, qui peut avoir une des valeurs suivantes :

- INCOMPLETE : l'hôte connaît l'adresse IP de son voisin mais l'adresse de couche lien n'a pas encore été déterminée,
- REACHABLE : le voisin a été joint il y a moins de t_1 secondes (en pratique t_1 est inférieur à quelques dizaines de secondes),
- STALE : le voisin n'a plus été joint depuis un temps supérieur à t_1,
- DELAY : le voisin n'a plus été joint depuis un temps supérieur à t_1, puis des paquets ont été envoyés au voisin depuis un temps inférieur à t_2, mais le voisin n'a pas encore confirmé la bonne réception de ces paquets,
- PROBE : le voisin n'est plus considéré comme joignable.

Les transitions entre ces différents états sont décrits dans l'annexe C de la RFC 4861. Téléchargez cette RFC, et dessinez l'automate décrit.

Exercice 5 : redirection

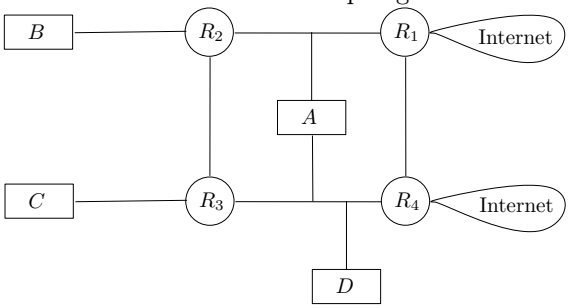
1. On considère la topologie suivante où R_1 et R_2 sont des routeurs :



On suppose que l'hôte A route tous les paquets qui ne sont pas destinés vers le préfixe local P vers le routeur R_1. Décrire la table de routage de A.

Il envoie maintenant des paquets destinés à la machine B. Donnez la route suivie par les premiers paquets envoyés, expliquez le méchanisme de redirection dans ce cas et donner la table de routage de A après exécution.

2. On considère maintenant la topologie de multi-homing suivante où les R_i, $1 \leqslant i \leqslant 4$, sont des routeurs :



On suppose que l'hôte A route tous les paquets qui ne sont pas destinés vers l'un des deux liens locaux vers le routeur R_1.

Il envoie des paquets destinés à la machine C. Donnez la route suivie par les premiers paquets envoyés, expliquez en quoi le méchanisme de redirection échoue. et donner la table de routage de A après exécution.

Comment régler ce problème manuellement ?