Téléinformatique & Protocoles

TD 9 : Protocole liaison

Corrigé

Cette représentation de protocole est un *résea d'évaluation*. Le seul état stable de ce protocole est l'état *émission*. Ensuite, trois types d'événement peuvent sortir le protocole de cet état :

- l'événement *"Il y a un bloc à émettre et la fenêtre d'émission n'est pas pleine"* ; auquel cas on exécute la fonction *E-émis* en envoyant le bloc de numéro de séquence i.
- l'événement *"Timeout expiré"* ; auquel cas on exécute la fonction *E-réémis* en renvoyant le bloc de numéro de séquence j.
- l'événement *"Un accusé de réception est arrivé"*. Auquel cas on exécute la fonction *"E-ACK"*.

1. Le récepteur n'accepte les trames que dans l'ordre donc il n'y a pas de fenetre de réception, donc ce n'est pas de la retransmission sélective.

   L'émetteur peut ne pas avoir reçu l'acquittement d'une trame pour envoyer la trame suivante. Il y a un mécanisme de fenêtre d'émission de taille K. Donc c'est bien un protocole de type retransmission continue ($\approx$ protocole 5 $\approx$ GoBack-N).

   La variable i n'est autre que le $PTAE$ (ProchaineTrameAEnvoyer) du Tannenbaum, c'est à dire 1 + la borne supérieure de la fenêtre d'émission. C'est le numéro de séquence

   La variable j n'est autre que le $AckAttendue$ c'est à dire la borne inférieure de la fenètre de réception.

   Enfin la variable a n'est autre que la variable TA (TrameAttendue) du Tannenbaum.

2. Dans le cas de perte de trames, il n'y a pas de problème, car le récepteur refuse toute trame qui ne soit pas exactement celle qu'il attendait. Toutes les suivantes sont refusées jusqu'à ce que la bonne trame soit réémise.

   Dans le cas de perte d'acquittement, on retrouve un problème qu'on a plus ou moins vu pour l'arrêt et attente : si le numéro de séquence est incorrecte, alors il faut tout de même envoyer un acquittement, sinon, l'émetteur va renvoyer la même trame. Or ici, si le numéro de séquence est incorrecte, on ne renvoie rien -> problème.

   Plus précisément, c'est lorsque toute une fenêtre a été émise et que tous les acquittements sont perdus que le problème se pose : Le récepteur n'acquitte rien et l'émetteur en déduit qu'il faut encore renvoyer les mêmes trames et ainsi de suite -> boucles sans progrès.

   Correction :

$$\begin{array}{ll} R-recu: & si(n \leq a) \\ & \{envoyerACK(n+1) \\ & si(n == a) \\ & \{a \leftarrow a+1; \\ & accepterbloc \\ & \} \\ & \} \end{array}$$

3. Dans le cas d'un déséquencement, tant que les numéros de séquence ne sont pas bornées, tout se passe bien, en cas de déséquencement de trame d'info ou d'acquittement.
4. L'intérêt des acquittements négatifs est simplement de ne pas attendre le timeout pour réémettre.
5. Il faut simplement prendre des comparaisons circulaires : $si(j < n \leq i)ou(i < j < n)ou(n \leq i < j)$. D'autre part dans $E-ACK$, la condition pour laquelle il faut réarmer le temporisateur n'est plus $j < i$ mais $j \neq i$.

6. En présence de déséquencement, il y a un problème car une trame i qui a été déséquencée prend du retard, mais elle n'est pas perdue. Le timeout expire et on réémet un clone de la trame i déséquencée et qui est acceptée, ce qui permet au récepteur de continuer son travail.

   Mais il suffit qu'on ait émis N trames et qu'on attende de nouveau la trame i. A ce moment, si la trame déséquencée dont nous avons parlé tout à l'heure surgit, elle peut être pris pour la nouvelle trame i.

   Pour éviter ce problème il faut simplement que le retard que peut prendre une trame soit inférieure à la durée minimale nécessaire pour émettre N trames. (N est le modulo).