

Etude du protocole du bit alterné

Objectifs pédagogiques :

- Appréhender la difficulté de raisonner sur des protocoles, liée à la répartition de l'algorithme, au parallélisme asynchrone
- Se familiariser avec des représentations simples : automates et diagrammes temporels comme outils de support au raisonnement.
- Faire le lien avec autres cours (algorithmique, théorie des langages pour les automates)
- Aborder des problèmes de sémantique de synchronisation.

On considère le protocole du bit alterné (on traduit « alternating » en alterné ou alternant), inséré en couche N entre une couche supérieure de niveau N+1 qui lui donne simplement un paquet de données M (« tableau de bits » de taille assez petit pour être transmis en une fois), et une couche inférieure qui achemine les bits qui lui sont fournis vers l'autre extrémité (communication point-à-point), mais peut les perdre (la corruption du contenu serait traitée par la couche supérieure avec une somme de contrôle dans les données M).

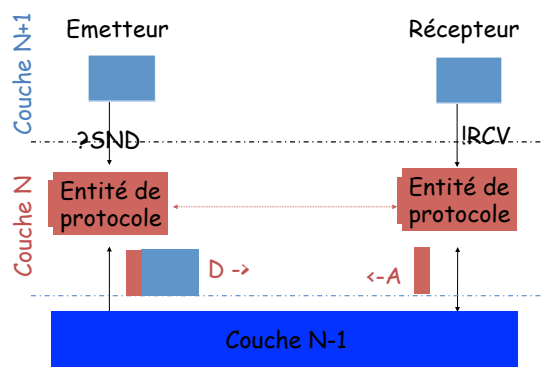
On considère pour simplifier un seul sens de communication¹, avec d'un côté un émetteur qui envoie des PDU de données D et de l'autre un récepteur qui renvoie des accusés A.

Format d'un PDU de type D (émetteur -> récepteur) : $\boxed{b} \boxed{M}$

Format d'un PDU de type A (récepteur -> émetteur) : \boxed{b} .

Comme on s'intéresse à représenter ce qui joue sur l'algorithme (le contrôle), on n'a pas besoin de tenir compte du contenu des données (M), qui n'influe pas sur le comportement du protocole. Donc on abstrait les PDU en D0 (qui représente tous les messages du type 0M), D1 (messages du type 1M) et A0, A1.

La couche supérieure offre du côté émetteur un SDU (Service Data Unit) : SND(M) permettant demander l'envoi du message M, et du côté récepteur un SDU : RCV(M) fournissant à la couche supérieure un message M correctement reçu.



Donc l'entité de protocole de l'émetteur a pour alphabet d'entrée {SND, A0, A1} et comme alphabet de sortie {D0, D1}. Du côté récepteur, l'alphabet d'entrée est {D0, D1} et l'alphabet de sortie {RCV, A0, A1}. Aux entrées peut s'ajouter une entrée « interne » spéciale, que l'on notera T, qui représente l'arrivée à échéance d'une temporisation.

¹ On rappelle qu'en fait le protocole est - comme TCP - bi-directionnel, permettant d'envoyer des données en retour en même temps que les accusés de réception (principe du « piggybacking »).

Chaque entité de protocole se comporte comme un automate, qui reste dans un état, tant qu'il ne reçoit pas un événement (une entrée venant d'un autre bloc, ou une temporisation interne). Lorsqu'il reçoit un événement, il réagit : il peut envoyer un message vers un autre bloc, et passer dans un autre état (ou rester dans le même).²

Question 1. *Représentez sous forme graphique l'automate à entrées et sorties associé à l'entité de protocole de l'émetteur. Les états seront différents selon la valeur du bit et l'existence ou non d'un temporisateur « armé ».*

Conventions habituelles de représentation :

- On fera figurer sur les transitions une étiquette correspondant à un couple entrée/sortie, couple noté E/S (séparé par une barre de fraction).
- Une entrée peut être : un SDU, un PDU ou l'échéance d'une temporisation (notée T). On soulignera les SDU pour mieux les distinguer des PDU.
- Une sortie peut être : un PDU, ou un SDU ou une combinaison des deux (deux sorties pour une même transition, séparées par une virgule).
- Les entrées (sauf la temporisation qui est interne) seront préfixées par un point d'interrogation « ? », les sorties par un point d'exclamation « ! ».

Question 2. *Représentez sous forme graphique avec les mêmes conventions l'automate récepteur.*

On suppose qu'il n'y a pas de borne sur le temps d'aller-retour.

Question 3. *Faites un diagramme montrant que si la connexion de réseau entre l'expéditeur et le destinataire ne préserve pas l'ordre (i.e. laisse s'intervertir) des différents messages qu'elle porte, ce protocole ne peut fonctionner correctement ; et donnez l'exemple d'une suite de caractères segmentée qui serait mal transmise.*

² Comme les machines sont très rapides, c'est comme s'il passait tout son temps à attendre dans un état ou un autre, et que le traitement d'un événement était instantané.