

Exemples de protocoles élémentaires



- Modèle**
 - Les couches réseau, liaison et physique sont indépendantes et communiquent par messages
 - Les transmissions de données se font de A vers B
 - Les seules erreurs sont des erreurs de transmission (pas de panne des ordinateur notamment)

Diapo
Edité le 27/03/08

© M.BESSON

vendredi 3 octobre 2008

10



Protocole simplex non restreint

- Pas d'erreur de transmission
- Les données sont traitées sans délai par les couches réseau
- Pas de problème de contrôle de flux: on dispose d'autant de mémoire tampon que nécessaire

Diapo
Edité le 27/03/08

© M.BESSON

vendredi 3 octobre 2008

11

Algo d'émission (couche liaison de A)

→ While (true) {

- ◆ Récupérer des données à envoyer de la couche réseau de 1
- ◆ Créer une trame contenant ces données
- ◆ La transmettre à la couche physique pour émission

→ }

Algo de réception (couche liaison de B):

→ While (true) { /* boucle infinie)

- ◆ Attendre un événement /* seul choix : arrivée d'une trame */
- ◆ Récupérer le LPDU auprès de la couche physique de B
- ◆ Transmettre le LSDU à la couche réseau

→ }

Protocole simplex arrêt et attente

- Canal bidirectionnel
- Pas d'erreur de transmission
- Les données sont traitées sans délai par les couches réseau
- contrôle de flux pour éviter de saturer le récepteur :
 - Le récepteur acquittera chaque trame reçue
 - L'émetteur attendra l'accusé de réception pour émettre la trame suivante
- Si l'aller/retour est long, on perd beaucoup en performances car rien n'est émis pendant l'attente de l'accusé de réception. Exemple: ligne rapide longue distance: vitesse élevée **mais latence importante**

vendredi 3 octobre 2008

12

- Algo d'émission (coucher liaison de A)
- While (true) {
 - ◆ Récupérer des données à envoyer de la couche réseau de 1
 - ◆ Créer une trame contenant ces données
 - ◆ La transmettre à la couche physique pour émission vers B
 - ◆ Attendre un événement /* on attend le ack */
- ➔ }
- Algo de réception (couche liaison de B):
 - ➔ While (true) { /* boucle infinie
 - ◆ Attendre un événement /* seul choix : arrivée d'une trame */
 - ◆ Récupérer le LPDU auprès de la couche physique de B
 - ◆ Transmettre le LSDU à la couche réseau
 - ◆ Créer l'accusé de réception
 - ◆ Le transmettre à la couche physique pour émission vers A

Protocole simplex sur canal bruité

- Canal bidirectionnel
- Canal bruité: erreurs de transmission
- Les données sont traitées sans délai par les couches réseau
- contrôle de flux pour éviter de saturer le récepteur :
 - Le récepteur acquittera chaque trame reçue
 - L'émetteur attendra l'accusé de réception pour émettre la trame suivante
 - Utilisation d'un numéro de séquence pour détecter les réémissions
 - Utilisation d'un timer pour réémettre une trame sans ack

vendredi 3 octobre 2008

13

- Solution incorrecte:
 - ➔ La machine A déclenche un timer à l'émission d'une trame
 - ➔ En cas d'erreur d'émission, la machine B n'émet de ack
 - ➔ Si le timer de A expire, A réémet la trame
- Pb: si l'émission est correcte et si l'acquittement se perd, la machine B reçoit deux fois la trame sans le savoir et la transmet deux fois à sa couche réseau
- Solution: un numéro de séquence sur les trames pour détecter les réémissions de trame.
- Pb: quelle taille pour le numéro de séquence ? Le protocole arrête et attende fait que la confusion ne peut porter que sur une trame et la suivante. Un numéro de séquence à deux valeurs (1 bit) suffit

Protocole simplex sur canal bruité

Algo d'émission (couche liaison de A)

- NoTrameSuivante=0;
- Récupérer des données à envoyer de la couche réseau de A
- While (true) {
 - Construire une trame contenant les données et le NoTrameSuivante
 - La transmettre à la couche physique pour émission vers B
 - Lancer un timer
 - Attendre un événement /* on attend le ack ou la fin du timer */
 - Si l'événement est l'arrivée d'une trame:
 - Récupérer le LPDU auprès de la couche physique de A
 - Si le NoAck du LPDU est égal à NoTrameSuivant+1
 - Désactiver le timer
 - Récupérer des données à envoyer de la couche réseau de A
 - Incrémenter(NoTrameSuivante)
 - finSi
 - FinSi
- }

Protocole simplex sur canal bruité

Algo de réception (couche liaison de B):

- NoTrameAttendue=0
- While (true) { /* boucle infinie */
 - Attendre un événement /* arrivée d'une trame ou erreur de checksum */
 - Si l'événement est l'arrivée d'une trame
 - Récupérer le LPDU auprès de la couche physique de B
 - Si le No de séquence du LPDU est égal à NoTrameAttendue
 - Transmettre le LSDU à la couche réseau
 - Incrémenter NoTrameAttendue
 - Créer l'accusé de réception avec NoTrameAttendue comme No de séquence
 - Le transmettre à la couche physique pour émission vers A
- }

- **Le timer doit être suffisamment long pour permettre**
- ◆ **le transfert d'une trame à destination**
- ◆ **Son traitement**
- ◆ **Le transfert et la réception de l'acquittement**