

TP d'ordonnancement temps réel réparti

F. Singhoff

30 septembre 2003

L'objectif du TP est de vous montrer que les techniques et algorithmes d'ordonnancement vus en TD s'appliquent en fait à de nombreuses ressources partagées d'un système temps réel, et en particulier aux ressources réseaux. Dans ce TP, on se propose donc d'explorer deux types de problèmes :

1. On présente d'abord une technique permettant d'obtenir des délais de bout en bout.
2. On montre ensuite que l'ordonnancer des messages est similaire à l'ordonnancer des tâches.

Exercice 1 : prise en main d'un outil de simulation

L'objectif de cet exercice est de vous présenter un outil de simulation qui automatise certaines des techniques vues en cours. En particulier, il fournit un certain nombre d'ordonnanceurs de base accompagnés de leurs tests de faisabilité (ou ordonnançabilité).

L'outil est installé pour des machines Solaris uniquement. Afin de pouvoir le lancer, exécuter d'abord la commande shell suivante :

```
source ~singhoff/PUBLIC/cheddar.csh
```

Une fois l'outil lancé, on vous demande de le tester de la façon suivante :

1. Ajouter un processeur qui utilise un **ordonnanceur HPF/POSIX.4** grâce au menu *"Edit/Add/Add a processor"*.
2. Créer 3 tâches grâce au menu *"Edit/Add/Add a task"*. Les paramètres de ces tâches sont : $S_1 = S_2 = S_3 = 0, P_1 = 30, C_1 = 6, P_2 = 5, C_2 = 3, P_3 = 10, C_3 = 2$. Les délais critiques sont égaux aux périodes. N'oubliez pas de déterminer un niveau de priorité correcte vis-à-vis de la période des tâches : on souhaite appliquer la méthode d'affectation de priorité Rate Monotonic. Avec cet outil, les niveaux de priorité vont de 1 à 255 (255 est le niveau de priorité le plus élevé).
3. Ordonnancer le jeu de tâches en appuyant sur l'icone *"Scheduling simulation"*.
4. Appliquer les tests de faisabilité de base en appuyant sur l'icone *"Scheduling feasibility"*.

Exercice 2 : temps de réponse de bout en bout

L'objectif de cet exercice est de vous montrer comment il est possible de calculer un temps de réponse de bout en bout d'un ensemble de tâches réparties sur plusieurs processeurs.

On étudie le système décrit dans la figure 1. Le système est constitué de deux processeurs. Le processeur *a* héberge les tâches *T1*, *T2* et *T5*. Le processeur *b* héberge les tâches *T3* et *T4*. Les

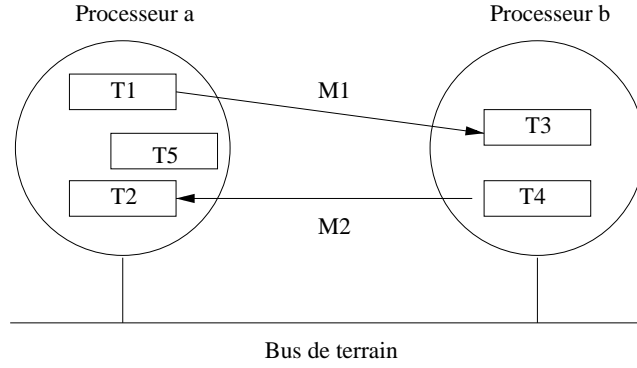


FIG. 1 – Une application répartie

tâches $T1$ et $T3$ communiquent grâce à un message périodique : le message $M1$. Les tâches $T4$ et $T2$ communiquent grâce à un message périodique : le message $M2$.

Les tâches et messages sont définis par les paramètres suivants :

Message	Période (en ms)	Temps de communication (en ms)
M1	100	6
M2	60	1

Le temps de communication inclut les temps de propagation, d'accès, de transmission et de traversée des couches. Le temps de réponse final d'un message i est obtenu par $TR(M_i) = J_i +$ temps de communication de i .

Tâche	Période (en ms)	Capacité (en ms)	Priorité	Processeur
T1	100	4	1	a
T2	60	5	2	a
T3	100	3	2	b
T4	60	2	1	b
T5	90	3	3	a

On souhaite calculer les délais de bout en bout de chaque tâche du système. Pour ce faire, nous allons appliquer la méthode Holistique :

1. Avec Cheddar, définissez un projet comportant les processeurs a et b . Notez qu'ici, les priorités des tâches vous sont données.
2. On pose $\forall i : J_i = 0$. Quels sont les temps de réponse des tâches et des messages ?
3. Pour quels tâches/messages doit on modifier le paramètre *Jitter*. Donner les valeurs du *Jitter* pour les tâches/messages concernés.
4. Recalculer les temps de réponse après avoir modifié les *Jitter*. Quels sont les nouveaux temps de réponse ?
5. Répéter les opérations 3 et 4 jusqu'à convergence des temps de réponse. Quels sont les délais de bout en bout des chaînes de traitement $T1/T3$ et $T4/T2$?

Exercice 3 : ordonnancement de messages périodiques

Dans l'exercice précédant, nous avons supposé connus l'ordonnancement des messages périodiques. Nous regardons maintenant ce point sur un réseau offrant des garanties temporelles.

On rappelle que le temps de communication d'un message est constitué des éléments suivants (cf. [COT 00]) :

- Le délai de traversée des couches (fixe).
- Le délai de transmission (variable car dépendant de la quantité d'information à transmettre).
- Le délai de propagation (variable car dépendant de la distance à parcourir).
- Le délai d'accès qui dépend du protocole MAC (variable).
- Le délai de réception (fixe).

Message	Période (en ms)	Taille (en octets)	FIP (en μs)	CAN (en μs)
a	4	2	100	80
b	16	4	150	150
c	4	4	150	150
d	8	6	200	180
e	64	8	450	250

Dans cet exercice, on suppose que les délais de propagation, de réception et de traversée des couches du réseau sont négligeables. Les délais de transmission sont données dans le tableau ci-dessus dans les cas où un réseau FIP et un CAN sont utilisés. La taille des messages est donnée à titre indicatif.

Application au bus de terrain CAN

On rappelle qu'un bus CAN [ZEI 96] utilise le protocole CSMA/CA (évitement de collision). Chaque message est identifié uniquement. L'identificateur constitue en fait une priorité d'accès au médium. L'arbitrage s'effectue grâce à une émission bit à bit de l'identificateur (cf. bit dominant et bit récessif). En cas de contention, les coupleurs émettant un message de plus faible priorité passent en mode lecture.

1. Le protocole CSMA le plus courant est celui utilisé par Ethernet (CSMA/CD). Quelles sont les différences entre CSMA/CD et CSMA/CA en terme de fonctionnement et de délais de communication.
2. Proposer une solution permettant d'ordonnancer les messages. Les messages peuvent-ils respecter leurs contraintes temporelles ?
3. Déterminer le temps d'accès au réseau pour chaque message. En déduire le temps de communication des messages.

Application au bus de terrain FIP

L'allocation du médium dans un réseau FIP [CIA 99] est réalisé par un arbitre centralisé. L'arbitre possède un fonctionnement cyclique. Chaque cycle est constitué :

- D'une phase d'émission des informations périodiques (nommées variables dans FIP).
- D'une phase d'émission des informations aperiodiques.

Pour la phase cyclique l'arbitre exploite une table construite hors-ligne. L'unité de base utilisée dans cette table est la période minimale (appelé microcycle) du jeu de messages périodiques. La taille de la table est égale au PPCM des périodes des messages (macrocycle).

1. Donner la valeur du microcycle et du macrocycle.
2. Proposer une table d'arbitrage permettant d'ordonnancer correctement les différents messages. Les messages peuvent-ils respecter leurs contraintes temporelles ?

3. A quelle technique d'ordonnancement s'inspire la technique de la table d'arbitrage.
4. Quel(s) algorithme(s) aurions nous pu appliquer pour construire automatiquement cette table.
5. Déterminer le temps d'accès au réseau pour chaque message. En déduire le temps de communication des messages.

Références

Ces exercices sont inspirés de [COT 00]. Exercices complémentaires : [COT 00, KRI 97].

- [CIA 99] CIAME. *Réseaux de terrain*. Edition Hermès, 1999.
- [COT 00] F. Cottet, J. Delacroix, C. Kaiser, and Z. Mammeri. *Ordonnancement temps réel*. Hermès, 2000.
- [KRI 97] C. M. Krishna and K. G. Shin. *Real-Time Systems*. Mc Graw-Hill International Editions, 1997.
- [ZEI 96] H. Zeitwanger. « CAN in Industrial Applications ». *Real Time Magazine*, (2) :20-24, June 1996.