

## Chapitre IV

# Sur les réseaux de communication temps-réel: contributions

### 1. Introduction

Le but de ce chapitre est de présenter les contributions significatives, à notre avis, que nous pensons avoir apportées au domaine des protocoles MAC temps-réel de classe 1 et classe 2, à la fois, en termes conceptuels et en termes de réflexions sur des protocoles existants.

Dans de nombreux protocoles de classe 1 [NFa, 90] [IEEE, 90] l'ordonnancement du trafic périodique (ordonnancement de base) est effectué hors-ligne et mis en oeuvre de manière centralisée dans une station particulière (nous avons indiqué au chapitre III tout l'intérêt de l'ordonnancement hors-ligne pour le trafic périodique). Ce n'est qu'ensuite, lorsque la séquence temporelle du trafic périodique a été spécifiée, que l'on essaie de voir comment on peut ordonnancer, dans les intervalles de temps vides, le trafic apériodique temps-réel (nous appelons cette technique la technique de l'essai d'ordonnancement à posteriori du trafic apériodique temps-réel). Notons que dans le contexte d'un ordonnancement hors-ligne, il est essentiel de définir des stratégies de changement de mode en-ligne afin de pouvoir prendre en compte l'évolutivité du fonctionnement d'un système distribué sur un réseau.

C'est dans ce double contexte (ordonnancement hors-ligne de trafic périodique et algorithme de changement de mode en-ligne; essai d'ordonnancement à posteriori du trafic apériodique temps-réel) que se situent deux types de travaux que nous avons effectués dans le domaine des protocoles de classe 1 et qui, à notre connaissance, représentent des résultats nouveaux:

- le premier type concerne la définition d'un algorithme de changement de mode en-ligne sachant que l'on a un ordonnancement hors-ligne basé sur l'algorithme RM (ces travaux ont été effectués lors d'études pour l'ordonnancement de trafic isochrone sur le réseau DQDB au moyen du mécanisme PA [IEEE, 90];
- le deuxième type concerne l'étude de l'ordonnançabilité du trafic apériodique temps-réel dans le réseau FIP et plus précisément ce qui est appelé dans la norme [NFa, 90] la scrutation déclenchée de variables apériodiques urgente et que nous appelons la scrutation indirecte d'une variable apériodique (c.f. III.3.3.5).

---

Dans le domaine des protocoles de classe 2, le protocole "jeton temporisé" [Gro, 82] est un protocole qui fait référence et dont plusieurs chercheurs ont analysé les caractéristiques et les propriétés.

Cependant, à notre connaissance, il n'y a pas eu de travaux faisant une analyse critique de ce protocole. Un mécanisme qui, à notre avis, mérite une analyse critique, est le mécanisme de la gestion du temps de cycle du jeton. On a en effet un temps de cycle irrégulier qui peut dépasser la valeur de  $T_{TRT}$  pour atteindre la valeur maximale de  $2 \cdot T_{TRT}$ . Ce fonctionnement a des conséquences fâcheuses pour le dimensionnement du protocole par rapport au trafic temps-réel car, à une prévision de temps de cycle maximal de  $2 \cdot T_{TRT}$ , ne correspond qu'un temps de cycle dont la valeur moyenne est  $T_{TRT}$ . Pratiquement, on a donc des ressources surdimensionnées. C'est précisément cet aspect du protocole du jeton temporisé qui a motivé notre réflexion et nous a amené à proposer une adaptation de ce protocole que nous appelons le protocole "jeton temporisé régulier" et dans lequel le temps de cycle ne subit pas des augmentations sporadiques jusqu'à  $2 \cdot T_{TRT}$ . De cette façon, le surdimensionnement est diminué.

Enfin, en ce qui concerne le protocole Profibus [DIN, 91], s'il présente des similitudes avec le protocole "jeton temporisé", il a cependant une différence fondamentale qui découle de la non allocation d'une bande passante bien définie, par station, pour le trafic temps-réel et qui a donc des incidences sur les performances de ce type de trafic. C'est précisément cet aspect du protocole Profibus qui a motivé notre réflexion et qui nous a amené à définir des conditions pour l'ordonnancement du trafic temps-réel dans le réseau Profibus.

En conséquence ce chapitre est divisé en deux grandes parties:

- la première partie est relative à des protocoles de la classe 1 et comprend deux parties: la première concerne la présentation d'un algorithme de changement de mode en-ligne sachant que l'on a un ordonnancement RM hors-ligne; la deuxième concerne la présentation de conditions pour l'ordonnancement du trafic aperiodique urgent (sur la base de la scrutation indirecte) dans le réseau FIP,
- la deuxième partie est relative à des protocoles de la classe 2 et comprend deux parties: la première concerne la définition du protocole "jeton temporisé régulier"; la deuxième concerne la présentation de conditions pour l'ordonnancement du trafic temps-réel dans le réseau Profibus.

## **2. Sur les protocoles de classe 1**

### **2.1. Algorithme de changement de mode de fonctionnement dans un protocole DQDB à ordonnancement RM hors-ligne [CVJ, 94] [JCV, 94]**

#### **2.1.1. Introduction**

Nous considérons un ensemble  $M = \{M_1, \dots, M_i, \dots, M_n\}$  de flux de slots à ordonnancer, avec  $\{C_i, P_i\} = \{1, P_i\}$  (slots PA de durée unitaire), chacun étant associé à un flux de messages isochrones (c.f. III.3.4).

L'algorithme de génération de la séquence temporelle étant basé sur l'algorithme RM, les slots sont alloués dans l'ordre inverse des périodes des flux. La séquence temporelle est évaluée sur l'intervalle  $t_i$ ,  $0 \leq t_i \leq ppcm\{P_1, P_2, \dots, P_n\}$ . Nous appelons  $ppcm\{P_1, P_2, \dots, P_n\}$  le cycle de la séquence temporelle.

Nous notons par la suite  $P_i^k$ , la  $k^{\text{ème}}$  occurrence de la période du flux  $M_i$  dans une séquence temporelle, et  $G$  comme l'intervalle correspondant à cette occurrence:  $k \cdot P_i \leq G \leq (k+1) \cdot P_i$ .

**Algorithme de génération de la séquence temporelle:**

Considérer l'ensemble  $M = \{M_1, \dots, M_i, \dots, M_n\}$  ordonné dans l'ordre croissant de leurs périodes:  $P_1 \leq P_2 \leq \dots \leq P_i \leq P_{i+1} \leq \dots \leq P_n$

Pour  $i = 1, 2, \dots, n$  faire: /\* pour chaque flux  $M_i$  \*/  
 Pour  $k = 0, 1, \dots, (\text{ppcm}/P_i - 1)$  /\* pour chaque  $P_i^k$  \*/  
 Allouer au flux  $M_i$ , le premier slot libre de l'intervalle  $G$

Exemple:

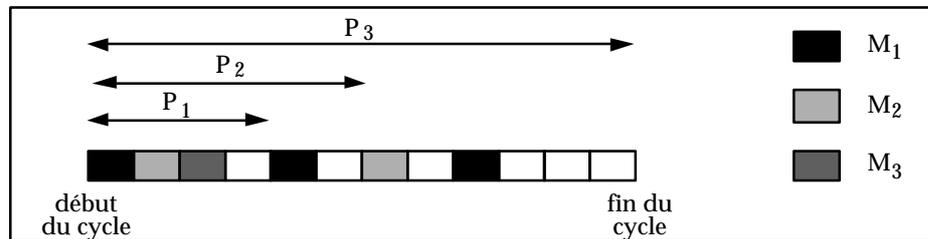


Figure 4.1: Exemple d'une séquence temporelle

Un exemple d'une séquence temporelle est présenté sur la figure 4.1. Nous avons 3 flux de slots:  $M_1(P_1 = 4)$ ,  $M_2(P_2 = 6)$  et  $M_3(P_3 = 12)$ . Alors  $\text{ppcm}(P_i) = 12$ .

### 2.1.2. Définitions

Nous définissons tout d'abord les ensembles de flux de slots des deux séquences temporelles: celle qui est en exécution et la nouvelle.

#### Définition 4.1: (Ensemble de flux de slots de la séquence temporelle en exécution)

Un ensemble de flux de slots  $M_{ste} = \{M_1, \dots, M_i, \dots, M_n\}$  avec des périodes  $P_1 \leq \dots \leq P_i \leq \dots \leq P_n$  est un ensemble où  $\forall M_i \in M_{ste}$ ,  $M_i$  est un flux de slots de la **séquence temporelle en exécution**.

#### Définition 4.2 (Ensemble de flux de slots de la nouvelle séquence temporelle):

Un ensemble de flux de slots  $M_{stn} = \{M_1, \dots, M_i, \dots, M_n\}$  avec des périodes  $P_1 \leq \dots \leq P_i \leq \dots \leq P_n$  est un ensemble où  $\forall M_i \in M_{stn}$ ,  $M_i$  est un flux de slots de la **nouvelle séquence temporelle**.

L'objectif de cet algorithme est de changer la séquence temporelle en exécution par une nouvelle séquence temporelle et d'assurer la *correction* de la génération de slots.

#### Définition 4.3 (Correction de la génération de slots):

La correction de la génération de slots est assurée si un et seulement un slot est attribué pour toute période  $P_i^k$ ,  $k = 0, 1, \dots, (\text{ppcm}/P_i - 1)$ ,  $\forall M_i$ ,  $i = 1, 2, \dots, n$ ,  $\{M_i \in M_{stn} \wedge M_i \in M_{ste}\}$ .

Le changement de séquence temporelle doit donc être transparent pour les flux de slots qui restent dans la séquence. Plus précisément, il faut éviter d'avoir des slots dupliqués ou perdus comme nous allons le montrer sur les deux exemples suivants.

### 2.1.3. Exemples

#### 1. Exemple 1: ajout d'un flux de slots

Considérons la séquence temporelle générée par un ensemble de 6 flux de slots  $M_{ste} = \{M_1, \dots, M_6\}$ , avec leurs périodes respectives  $P_1 = 5, P_2 \dots = P_6 = 20$  et la nouvelle séquence qui est obtenue après l'inclusion du flux  $M_7$  avec période  $P_7 = 5$  (figures 4.2.a et 4.2.b). Supposons que le changement de séquence soit fait après la génération du quatrième slot de la séquence temporelle en exécution. Les slots seront générés comme illustré sur la figure 4.2.c. Nous pouvons observer qu'un slot supplémentaire est attribuée au flux  $M_4$  pour cette occurrence de la période. **On a donc une duplication de slots.**

L'attribution d'un slot supplémentaire pour un flux  $M_i$  durant la  $k^{ème}$  occurrence de la période  $P_i$ , résulte du fait que la condition du flux de slots est différente dans les séquences temporelles lors du changement. Considérant l'exemple de la figure 4.2, la condition de  $M_4$  dans la séquence temporelle en exécution est: *le slot demandé pour  $P_4^1$  a été déjà attribué*; dans la nouvelle séquence temporelle sa condition est: *le slot demandé pour  $P_4^1$  n'a pas encore été attribué*.

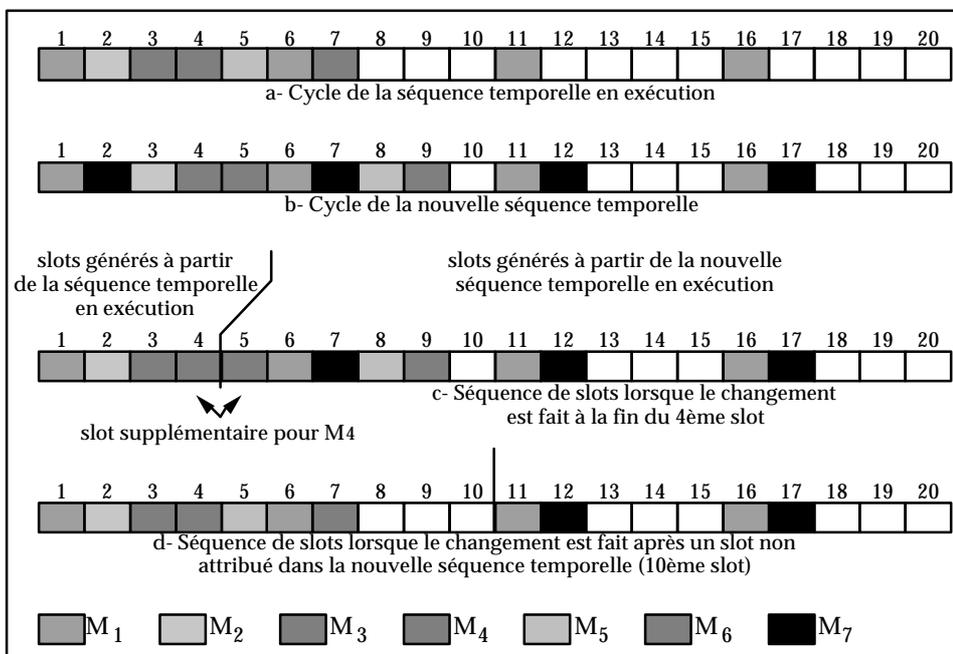


Figure 4.2: Le changement de la séquence temporelle lorsqu'un flux de messages est ajouté

#### 2. Exemple 2: retrait d'un flux de slots

Considérons le cas contraire au précédent, avec les mêmes paramètres. La nouvelle séquence temporelle et la séquence temporelle en exécution de l'exemple précédent sont respectivement la séquence temporelle en exécution et la nouvelle séquence temporelle dans cet exemple (figures 4.3.a, 4.3.b).

Supposons que le flux de slots doit être retiré de la séquence temporelle. Si le changement est fait après la génération du quatrième slot, un slot n'est pas attribué au flux  $M_4$  pour  $P_4^1$  (figure 4.3.c) et **nous avons donc une perte**. Comme dans l'exemple précédent, la condition de  $M_4$  est différent dans les séquences temporelles à l'instant du changement: *un slot n'a pas été attribué à  $M_4$  dans la séquence temporelle en exécution*; *un slot n'a pas été attribué à  $M_4$  dans la nouvelle séquence temporelle*.

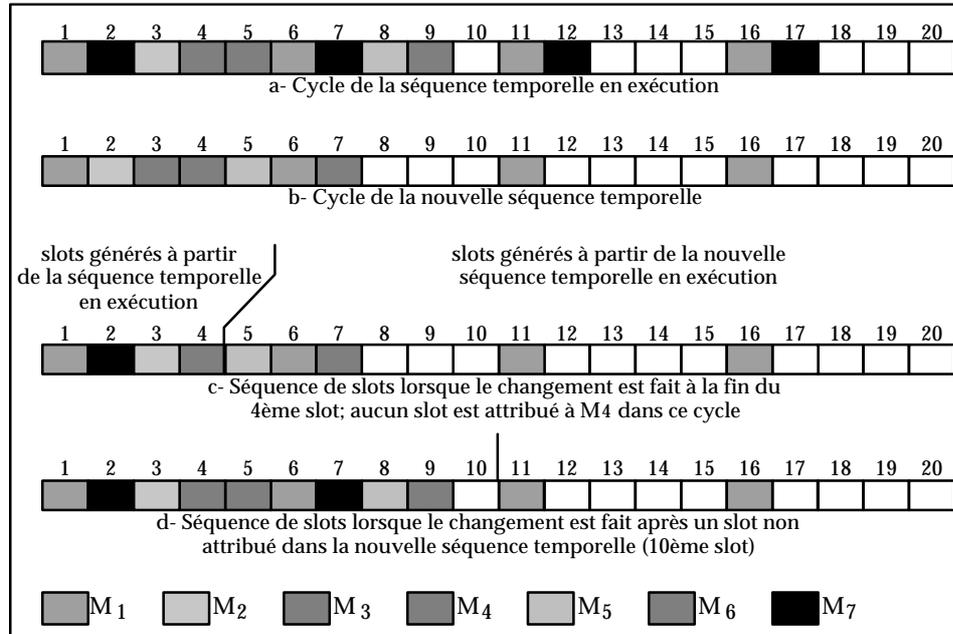


Figure 4.3: Le changement de la séquence temporelle lorsqu'un flux de messages est retiré

### 3. Conclusion

Les deux exemples précédents ont montré que la correction de la génération de slots ne peut pas être assurée lorsque le changement de la séquence temporelle est fait à un instant arbitraire.

#### 2.1.4. Concepts

Nous introduisons dans cette section deux types de concepts: des concepts sur la condition des flux de messages dans une séquence temporelle et des concepts sur l'instant de comparaison de deux séquences. Le premier type de concept sert à identifier la condition qui doit être respectée à l'instant du changement. Le deuxième type établit une correspondance entre les cycles des deux séquences.

##### 1. Concepts sur la condition des flux de messages dans une séquence temporelle

###### Définition 4.4 (Flux de slots en attente):

Un flux de slots  $M_i$  est en attente à l'instant  $t_l$  d'une séquence temporelle avec  $k \cdot P_i \leq t_l \leq (k+1) \cdot P_i$ ,  $k = 0, 1, \dots$  ppcm $\{P_1, P_2, \dots, P_n\}$ , si aucun slot n'a été attribué pour ce flux de slots pour la  $k^{\text{ème}}$  occurrence de la période.

###### Définition 4.5 (Ensemble de flux de slots en attente):

Un ensemble de flux de slots  $M_p = \{M_1, M_2, \dots, M_n\}$  avec des périodes  $P_1 \leq P_2 \leq \dots \leq P_n$  est en attente à l'instant  $t_l$ , si  $\forall M_i \in M_p$ ,  $M_i$  est un flux de slots en attente.

Nous notons  $M_{pe}$  et  $M_{pn}$  l'ensemble de flux de slots en attente à l'instant  $t_l$ , respectivement, de la séquence temporelle en exécution et de la nouvelle séquence temporelle.

---

**Lemme (4.1 Flux de slots en attente):**

Si  $t_l$  est l'instant initial d'un slot non attribué, alors l'ensemble  $M_p$  de la séquence temporelle est vide à l'instant  $t_l$ .

**Démonstration**

Supposons qu'il y a un flux de slots  $M_i$  en attente à l'instant  $t_l$ ,  $k \cdot P_i \leq t_l \leq (k+1) \cdot P_i$ ,  $k = 0, 1, \dots, ppcm\{P_1, P_2, \dots, P_n\}$ , et  $t_l$  est l'instant initial d'un slot non attribué. Ceci contredit l'algorithme de génération de la séquence temporelle qui attribue le premier slot libre de l'intervalle  $G$ ,  $k \cdot P_i \leq G \leq (k+1) \cdot P_i$ ,  $k = 0, 1, \dots, ((ppcm/P_i) - 1)$  au flux  $M_i$ .

**Fin démonstration****2. Concepts sur le changement entre deux séquences temporelles**

Les séquences temporelles peuvent avoir des durées différentes. Pour effectuer le changement entre deux séquences temporelles, nous devons, premièrement, définir le cycle à partir duquel les deux séquences doivent être comparées.

Considérons un instant  $t_{lc}$  arbitraire de la séquence temporelle en exécution. Nous identifions le cycle *correspondant* dans la nouvelle séquence temporelle à partir duquel les deux séquences peuvent être comparées. Notons les  $ppcm$  de la séquence temporelle en exécution et de la nouvelle séquence temporelle respectivement  $ppcm_e$  et  $ppcm_n$ . Il faut considérer deux cas:

- $ppcm_e \leq ppcm_n$ :  $t_{lc}$  appartient au premier cycle de la nouvelle séquence temporelle.
- $ppcm_e > ppcm_n$ : Notons  $X = (ppcm_e / ppcm_n)$ . Nous avons  $X$  cycles de la nouvelle séquence temporelle durant le cycle courant de la séquence temporelle en exécution. De façon générale,  $t_{lc}$  est telle que  $(b-1) \cdot ppcm_n \leq t_{lc} \leq b \cdot ppcm_n$ , avec  $b = 1, 2, \dots, X$ . L'instant  $t_{lc}$  appartient donc au  $b^{\text{ème}}$  cycle de la nouvelle séquence temporelle.

**Définition 4.6 (Instant de changement de la séquence temporelle):**

L'instant de changement de la séquence temporelle doit assurer la correction de la génération de slots.

**Théorème 4.2:**

Si le changement de la séquence temporelle est fait à l'instant  $t_{lc}$  et si à cet instant les ensembles  $C_{pe}$  et  $C_{pn}$  sont vides,  $C_{pn}$  étant l'ensemble de flux de slots en attente de la  $b^{\text{ème}}$  occurrence du cycle de la nouvelle séquence temporelle, alors la correction de la génération des slots est assurée.

**Démonstration (théorème):**

Puisqu'à l'instant  $t_{lc}$  il n'y a pas de flux de slots en attente dans les deux séquences temporelles, cela signifie que les slots ont été déjà attribués pour le  $P_i^k$  courant des deux séquences temporelles,  $\forall M_i, i = 1, 2, \dots, n$ . Alors, même si la séquence temporelle est changée à cet instant, comme il n'y a pas de flux de slots en attente dans les deux séquences temporelles, un et seulement un slot sera attribué pour tout  $P_i^{k+1}$  de la nouvelle séquence temporelle,  $\forall M_i, i = 1, 2, \dots, n$ . Par conséquent, la correction est assurée.

**Fin démonstration (théorème)**

Remarquons que l'approche la plus simple pour changer la séquence temporelle et garantir la correction de la génération de slots, serait d'effectuer le changement à la fin du cycle de la

séquence temporelle en exécution. Cependant, cette approche présente un inconvénient: elle entraîne un temps de réponse important aux demandes des utilisateurs dans le cas d'un cycle de longue durée.

### 2.1.5. Ajout de flux de slots

Supposons une séquence temporelle nouvelle générée après l'addition d'un nombre arbitraire de flux de slots à la séquence en exécution. Nous avons donc  $\#(M_{stn}) > \#(M_{ste})$ . Nous définissons une condition suffisante mais pas nécessaire pour identifier le premier instant  $t_{lc}$ , où les ensembles  $M_{pe}$  et  $M_{pn}$  sont vides.

#### Lemme 4.2 (Flux de slots en attente):

$\forall t_{lc}$ , si  $M_{pn}$  est vide, alors  $M_{pe}$  est aussi vide.

#### Démonstration

Considérons  $\#(M_{stn}) - \#(M_{ste}) = z$ . Nous montrons que pour  $z = 1$  la proposition est vraie. Considérons  $M_a$  le flux de slots ajouté à l'ensemble  $M_{pe}$  et  $P_a$  sa période. Nous considérons premièrement le cas où l'addition d'un nouveau flux ne modifie pas le cycle de la séquence temporelle. Après, nous généralisons la démonstration.

• Cas 1:  $ppcm_e = ppcm_n$ . Deux cas doivent être considérés:

- Cas 1.1: Le flux de slots  $M_a$  a une période  $P_a$  telle que  $P_1 \dots \leq P_n = P_a$ ;

Considérons la construction de la nouvelle séquence temporelle. Les slots sont attribués en premier aux flux  $M_1, M_2, \dots, M_n$  et un slot sera attribuée en dernier à chaque  $P_a^k$  du flux  $M_a$ . L'addition du flux de slots  $M_a$  ne va pas modifier la position des flux  $M_1, M_2, \dots, M_n$  par rapport à leurs positions dans la séquence temporelle en exécution: ils ont la même position dans les deux séquences temporelles. Par conséquent, tous les slots non attribués dans la nouvelle séquence temporelle sont aussi non attribués dans la séquence temporelle en exécution. Il découle du lemme 4.1 que  $\forall t_{lc}$ , si l'ensemble  $M_{pn}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pe}$  est aussi vide.

- Cas 1.2: Le flux de slots  $M_a$  a une période  $P_a$  telle que  $P_1 \dots \leq P_a \dots \leq P_n$ ;

Établissons une correspondance entre les positions occupées par les flux de slots dans les deux séquences temporelles. Selon l'algorithme de génération de la séquence temporelle, la position d'un flux de slots dans la nouvelle séquence temporelle sera la même, ou décalée à droite de  $m$  trames par rapport à sa position dans la séquence temporelle en exécution. La correspondance entre les deux positions dans les deux séquences est donc:

- Un flux de slots  $M_i$ , avec  $1 \leq i < a$ , occupe la même position dans les deux séquences.
- La position d'un flux de slots  $M_i$ , avec  $a < i \leq n$ , dans la séquence temporelle en exécution est occupé par un flux de slots  $M_j$  avec  $a \leq j \leq i$  dans la nouvelle séquence.

Tous les slots occupés dans la séquence temporelle en exécution sont donc occupés dans la nouvelle. Alors, tous les slots non attribués dans la nouvelle séquence temporelle sont aussi non attribués dans la séquence en exécution.  $\forall t_{lc}$ , si  $t_{lc}$  est l'instant initial d'un slot non attribué dans la nouvelle séquence temporelle, il en est de même aussi dans la séquence temporelle en exécution. Il découle du lemme 4.1 que  $\forall t_{lc}$ , si l'ensemble  $M_{pn}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pe}$  est aussi vide.

• Cas 2:  $ppcm_e < ppcm_n$ . Notons  $X = (ppcm_n / ppcm_e)$ . Deux cas doivent être considérés:

- Cas 2.1: Le flux de slots  $M_a$  a une période  $P_a$  telle que  $P_1 \dots \leq P_n < P_a$ ;

---

La démonstration est la même que dans le cas 1.1; la différence étant le nombre d'occurrences des périodes des slots dans les deux séquences temporelles. Dans le cas 1.1, le nombre d'occurrences est le même. Dans le cas présent le nombre d'occurrences dans la nouvelle séquence temporelle est  $X$  fois le nombre d'occurrences dans la séquence en exécution.

- *Cas 2.2: Le flux de slots  $M_a$  a une période  $P_a$  telle que  $P_1 \dots \leq P_a \dots \leq P_n$ ;*

La démonstration est la même que dans le cas 1.2; la différence étant le nombre d'occurrences des périodes des flux de slots dans les deux séquences temporelles. Dans le cas 1.2, le nombre d'occurrences est le même. Dans le cas présent le nombre d'occurrences dans la nouvelle séquence temporelle est  $X$  fois le nombre d'occurrences dans la séquence en exécution.

*Hypothèse inductive:* Nous supposons que la proposition est vraie pour  $z = x$  et nous montrons qu'elle est vraie pour  $z = x + 1$ .

Considérons l'ensemble  $M_{stn'}$  avec  $\#(M_{stn'}) = x$  et l'ensemble  $M_{stn}$  avec  $\#(M_{stn}) = x + 1$ . Nous supposons donc que  $\forall t_{lc}$ , si  $M_{stn'}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pe}$  est aussi vide. Il a été montré dans le cas de base que  $\forall t_{lc}$ , si  $M_{stn}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pn'}$  est aussi vide. Alors,  $\forall t_{lc}$ , si  $M_{pn}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pe}$  est aussi vide.

### **Fin démonstration**

Finalement, nous définissons l'instant de changement  $t_{lt}$  qui préserve la correction de la génération de slots.

### **Théorème 4.3:**

*L'instant de changement  $t_{lt}$  est celui qui correspond à l'instant initial du premier slot non attribué dans la nouvelle séquence temporelle.*

### **Démonstration:**

Il découle du lemme 4.1 que l'ensemble  $M_{pn}$  est vide à l'instant  $t_{lt}$ . D'après le lemme 4.2, l'ensemble  $M_{pe}$  est aussi vide. Alors, d'après le théorème 4.2, la correction de la génération des slots est assurée.

### **Fin démonstration**

Nous représentons dans la figure 4.2.d, le changement de séquence temporelle qui assure la correction de la génération de slots.

## **2.1.6. Retrait de flux de slots**

Nous avons montré (théorème 4.2) que lorsque les ensembles  $M_{pe}$  et  $M_{pn}$  sont vides, le changement peut être fait et la correction de la génération de slots est assurée. Analysons donc le cas où plusieurs flux de slots sont retirés.

Considérons le lemme 4.2. Nous pouvons facilement voir qu'il peut être adapté à travers une permutation entre la séquence temporelle en exécution et la nouvelle séquence.

### **Lemme 4.3:**

*$\forall t_{lc}$ , si  $M_{pe}$  est vide à l'instant  $t_{lc}$ , alors  $M_{pn}$  est aussi vide.*

Nous pouvons donc définir l'instant de changement  $t_{lt}$ .

**Théorème 4.4:**

*L'instant de changement  $t_{it}$  est celui qui correspond à l'instant initial du premier slot non attribué dans la séquence temporelle en exécution.*

**Démonstration**

Il découle du lemme 4.1 que l'ensemble  $M_{pe}$  est vide à l'instant  $t_{it}$ . D'après le lemme 4.3, l'ensemble  $M_{pn}$  est aussi vide. Alors, d'après le théorème 4.2, la correction de la génération de slots est assurée.

**Fin démonstration**

Nous représentons dans la figure 4.3.d, le changement de séquence temporelle qui assure la correction de la génération de slots.

**2.1.7. Temps maximum pour le changement de la séquence temporelle**

Le temps maximum pour remplacer la séquence temporelle en exécution par une nouvelle séquence temporelle, dépend du nombre maximum de slots consécutifs attribués aux flux de messages. Nous définissons par la suite la borne supérieure de ce nombre.

Nous allons, tout d'abord, caractériser l'ensemble de flux de slots périodiques qui entraîne le taux maximum d'utilisation du réseau. Pour ceci nous utilisons un résultat du domaine de l'ordonnancement des tâches périodiques sur un processeur.

Dans [LL, 73] (théorèmes 4 et 5, pages 181-183), les auteurs ont montré que la borne supérieure minimale obtenue pour le taux d'utilisation d'un processeur ( $U_x = n \cdot (\sqrt[n]{2} - 1)$ ) peut être atteinte si et seulement si  $C_i = P_{i+1} - P_i$  et  $P_n < 2 \cdot P_1$ . Ces deux relations caractérisent l'ensemble de tâches qui entraîne le taux maximum d'utilisation du processeur lorsqu'il est ordonné par l'algorithme RM.

Dans le contexte de notre étude, les slots sont de taille unitaire ( $C_i = 1$ ). L'ensemble de flux de slots périodiques  $M_M$  qui entraîne le taux maximum d'utilisation du réseau ( $n \cdot (\sqrt[n]{2} - 1)$ ) est donc caractérisé par:  $P_{i+1} = P_i + 1$  et  $P_n < 2 \cdot P_1$ . Si nous résolvons la première équation, nous obtenons  $P_n = P_1 + (n - 1)$ . En remplaçant ce résultat dans la deuxième équation, nous avons  $P_1 > (n - 1)$ . L'ensemble  $M_M$  est donc caractérisé par  $n$  flux de slots périodiques avec des périodes consécutives et  $P_1 > (n - 1)$ .

Considérons la séquence temporelle  $ST_M$  générée par l'ensemble  $M_M$  avec l'algorithme d'ordonnancement hors-ligne proposé. Nous montrons qu'il y a toujours un slot non attribué durant l'intervalle critique de cette séquence.

**Lemme 4.4**

*Il y a toujours un slot non attribué durant l'intervalle critique de la séquence  $ST_M$  générée par l'ensemble  $M_M$*

**Démonstration**

Nous montrons premièrement que, si  $P_1 = n$  le test d'ordonnabilité est faux; et deuxièmement que, si  $P_1 \geq (n + 1)$  il y a toujours un slot non attribué durant l'intervalle critique de la séquence  $ST_M$ .

- Cas 1:  $P_1 = n$

L'ensemble  $M_M$  a les caractéristiques suivantes:  $P_i = n$ ,  $P_{i+1} = P_i + 1$  et  $P_n < 2 \cdot P_i$ ; Le taux d'utilisation du réseau est donné par  $U = \sum_{k=n}^{2n-1} 1/k$  qui est toujours supérieur à  $n(2^{1/n} - 1)$ .

- Cas 2:  $P_i \geq (n+1)$

Dans l'intervalle critique, un slot sera attribué au flux  $M_n$ , au plus tard, dans l'intervalle  $[n-1, n[$ . Puisque le prochain slot pour le flux  $M_i$  sera attribué au plus tôt à l'instant  $(n+1)$ , il y a au moins un slot non attribué dans l'intervalle  $[n, n+1[$ .

### **Théorème 4.5**

*Le nombre maximum de slots consécutifs attribués dans la séquence temporelle générée avec l'algorithme d'ordonnancement hors-ligne est borné supérieurement par  $(P_n - 1)$*

### **Démonstration**

Il découle du lemme 4.4 qu'il y a toujours un slot libre dans l'instant critique de la séquence temporelle. Supposons que ce slot est le dernier de l'instant critique. Il y a donc au maximum  $(P_n - 1)$  slots consécutifs attribués dans la séquence temporelle.

## **2.2. Conditions d'ordonnancement du trafic aperiodique temps-réel dans le réseau FIP [VJ, 93] [VJ, 94]**

### **2.2.1. Cadre de l'analyse**

Nous présentons tout d'abord la contrainte de protocole, c'est-à-dire, la condition pour que la charge maximale du trafic aperiodique urgent pendant un macro-cycle puisse être écoulee pendant ce macro-cycle (garantie de non-surcharge). Dans le cas contraire, on aurait une accumulation de charge sur les macro-cycles suivants ce qui traduirait une instabilité du protocole. Nous évaluons ensuite la contrainte de l'échéance.

Considérons les notations suivantes:

- $t_r$  est le temps de retournement d'un modem;
- $C_{\{id\_dat\}}$ ,  $C_{\{rp\_dat\}}$ ,  $C_{\{id\_rq\}}$  et  $C_{\{rp\_rq\}}$  sont les durées maximales d'utilisation du réseau associées respectivement aux PDU's  $id\_dat$ ,  $rp\_dat$ ,  $id\_rq$  et  $rp\_rq$ ; nous ne distinguons pas ici les durées associées aux PDU's  $rp\_dat$  et  $rp\_dat\_rq$  (nous parlons toujours de  $C_{\{rp\_dat\}}$ ).

### **2.2.2. Contrainte de protocole**

Afin d'évaluer cette contrainte, on doit tout d'abord calculer la durée maximale associée au transfert d'un message dans chaque type de transfert (c.f. III.3.3: figures 3.7, 3.8 et 3.9):

- pour un message d'un flux périodique (scrutation périodique de variables périodiques):

$$C_i = C_{\{id\_dat\}} + C_{\{rp\_dat\}} + 2 \cdot t_r \quad (4.1)$$

- pour une variable d'un flux aperiodique, dans le cas du transfert par scrutation directe (à condition qu'une seule variable soit demandée):

$$C_j^{(dir)} = C_{\{id\_rq\}} + C_{\{rp\_rq\}} + C_{\{id\_dat\}} + C_{\{rp\_dat\}} + 4 \cdot t_r \quad (4.2)$$

- pour une variable d'un flux apériodique, dans le cas où le transfert est effectué par scrutation indirecte:

$$C_j^{(indir)} = 2 \cdot (C_{\{id\_dat\}} + C_{\{rp\_dat\}}) + C_{\{id\_rq\}} + C_{\{rp\_rq\}} + 6 \cdot t_r \quad (4.3)$$

La contrainte du protocole (afin que l'on ait une garantie de non-surcharge) est:

$$\sum_{i=1}^n C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \sum_{\substack{j=1 \\ Ma_j \text{ scruté} \\ \text{directement}}}^s C_j^{(dir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil + \sum_{\substack{j=1 \\ Ma_j \text{ scruté} \\ \text{indirectement}}}^s C_j^{(indir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil \leq T_{MC} \quad (4.4)$$

### 2.2.3. Contrainte de l'échéance

Le calcul de la contrainte de l'échéance nécessite d'introduire deux notions (figure 4.4):

- le *retard d'avertissement*  $Da_j$  de l'arbitre qui représente le temps écoulé entre l'instant d'occurrence d'une requête d'une variable apériodique d'un flux  $Ma_j$  dans une station et l'instant où l'arbitre est informé de cette requête; l'arbitre en sera informé dans la fenêtre périodique suite à une scrutation périodique de variables périodiques  $Mp_i$  de cette station (fin de  $C_i^{(ind).1}$  (paragraphe III.3.3.3: figure 3.9)).
- le *retard d'ordonnancement*  $Do_j$ , qui représente le temps écoulé entre l'instant où l'arbitre est informé de la demande de transfert et l'instant où l'arbitre a fini l'ordonnancement du transfert de cette variable (fin de  $C_j^{(ind).2}$  (paragraphe III.3.3.3: figure 3.9)).

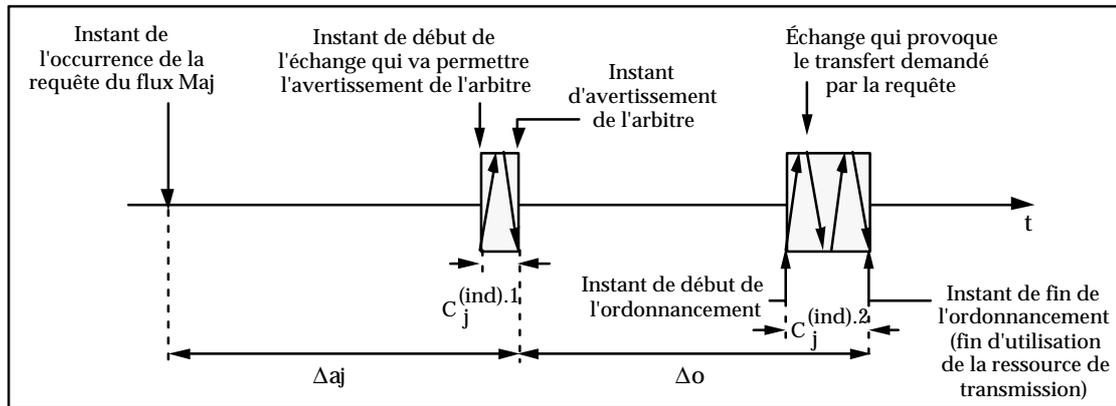


Figure 4.4: Retards d'ordonnancement et d'avertissement dans la scrutation indirecte d'une variable apériodique

Ces deux notions sont visualisées sur la figure 4.4. Pour bien comprendre cette figure, il est important de se reporter au scénario de la technique de la scrutation indirecte de variables apériodiques donnée au chapitre III (paragraphe III.3.5; figure 3.9).

Le retard d'avertissement  $Da_j$ , est conditionné par les périodes  $P_i$  des variables des flux périodiques  $Mp_i$  de la station  $k$  à laquelle appartient le flux apériodique  $Ma_j$ . On a donc la relation suivante:

$$Da_j \leq \min_{Mp_i \in \text{nœud}_k} \{P_i\} + (C_{\{rp\_dat\}} + t_r) \quad (4.5)$$

où  $\text{nœud}_k$  représente l'ensemble des flux temps-réel de la station  $k$  et  $(C_{\{rp\_dat\}} + t_r)$  représente le temps écoulé entre l'instant de début de l'envoi de la trame  $rp\_dat\_rq$  (que nous notons  $rp\_dat$ ; c.f. début du paragraphe) par la station  $k$  qui va permettre l'avertissement de l'arbitre

et l'instant où l'arbitre est informé. Notons que, lorsqu'il est informé, l'arbitre enregistre cette requête dans la "file des demandes apériodiques urgentes".

Le retard d'ordonnancement dépend de l'état de la "file de demandes apériodiques urgentes" (file FIFO) à l'instant où l'arbitre est averti de cette requête. Nous considérons le pire cas, c'est-à-dire l'arbitre n'est averti de cette requête que juste après avoir été averti de toutes les autres requêtes possibles. En conséquence de la considération du pire cas, le retard d'ordonnancement  $Do_j$  est le même quel que soit le flux  $Ma_j$  considéré; on le note donc  $Do$ .

Comme l'ordonnancement du trafic apériodique par scrutation indirecte n'est effectué qu'après l'ordonnancement du trafic périodique, et compte tenu de la satisfaction de la contrainte du protocole, on a donc la relation suivante:

$$Do \leq \sum_{i=1}^n C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \sum_{\substack{j=1 \\ Ma_j \text{ scruté} \\ \text{directement}}}^s C_j^{(dir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil + \sum_{\substack{j=1 \\ Ma_j \text{ scruté} \\ \text{indirectement}}}^s C_j^{(indir)} \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil \leq T_{MC} \quad (4.6)$$

**Contrainte de l'échéance:** Pour garantir le respect des échéances du trafic apériodique urgent, il faut garantir que, quel que soit le flux apériodique  $Ma_j$ , la somme des retards d'avertissement et d'ordonnancement soit inférieure à l'échéance  $d_j$ , c'est-à-dire:

$$Da_j + Do \leq d_j, \quad 1 \leq j \leq s \quad (4.7)$$

## 2.2.4. Exemple

Considérons un système avec 6 flux de messages périodiques  $Mp = \{Mp_1 \dots Mp_6\}$ , caractérisés par  $\{C_i, P_i\} = \{(4, 40), (4, 40), (4, 80), (4, 80), (4, 80), (4, 160)\}$ , et 6 flux de messages apériodiques urgents identiques  $Ma = \{Ma_1 \dots Ma_6\}$ , scrutés indirectement et caractérisés par  $\{C_i, P_i\} = \{(4, 155), \dots, (4, 155)\}$ .

Les flux de messages sont divisés parmi 3 stations de la façon suivante:  $nœud_1 = \{Mp_1, Mp_2, Ma_1, Ma_2\}$ ,  $nœud_2 = \{Mp_3, Mp_4, Ma_3, Ma_4\}$  et  $nœud_3 = \{Mp_5, Mp_6, Ma_5, Ma_6\}$

Les micro et macro cycles sont respectivement:

$$T_{mc} = p \text{gcd}\{40, 80, 160\} = 40 \quad \text{et} \quad T_{MC} = p \text{pcm}\{40, 80, 160\} = 160$$

La contrainte du protocole impose que:

$$\sum_{i=1}^n C_i \cdot \left\lceil \frac{T_{MC}}{P_i} \right\rceil + \sum_{j=1}^s C_j \cdot \left\lceil \frac{T_{MC}}{P_j} \right\rceil = 60 + 48 \leq T_{MC} = 160 \quad (\text{condition vraie})$$

Le retard d'avertissement dépend des durées  $C_{\{rp\_dat\}}$  et  $t_r$  qui ne sont pas connues; néanmoins, sachant que  $C_i > C_{\{rp\_dat\}} + t_r$ , ce retard est borné par:

$$Da_j = \{44, 44, 84, 84, 84, 84\}$$

Le retard d'ordonnancement est:

$$Do = 108$$

La contrainte de l'échéance impose que:  $Da_j + Do \leq d_j$ ,  $1 \leq j \leq s$ , ce qui n'est vrai que pour les flux apériodiques appartenant à la station 1 ( $152 < 155$ ); pour les flux apériodiques appartenant aux stations 2 et 3, la contrainte de l'échéance n'est pas respectée ( $192 < 155$ ).

On voit que, quoique le taux d'utilisation du réseau dû au trafic périodique ne dépasse pas 37.5%, nous ne pouvons ordonnancer que des flux apériodiques supplémentaires appartenant

à la première station; il n'est pas possible d'ordonnancer des flux à faible taux d'utilisation du réseau (2.6%) dans les stations 2 et 3.

### 2.2.5. Conclusion

Les expressions relatives à l'ordonnançabilité du trafic aperiodique urgent (par la technique de scrutation indirecte) montrent que l'efficacité de ce service, dans l'hypothèse du pire cas, est réduite, car:

- on a besoin d'échanges de PDUs supplémentaires pour informer l'arbitre des requêtes en attente dans les stations (séquence d'échanges  $id\_dat/rp\_dat\_rq$  et  $id\_rq/rp\_rq$ );
- le délai d'avertissement est élevé, à cause de l'asynchronisme entre la requête aperiodique urgente et l'échange de PDUs  $id\_dat/rp\_dat\_rq$  ;
- le caractère FIFO de la file de demandes aperiodiques urgentes génère des inversions de priorité non contrôlées et donc le délai d'ordonnancement doit être évalué avec la file FIFO pleine à l'instant de l'échange de PDUs  $id\_rq/rp\_rq$

La considération du pire cas induit évidemment un test d'ordonnancement pessimiste.

## 3. Sur les protocoles de classe 2

### 3.1. Le protocole "jeton temporisé régulier"

#### 3.1.1. Principe

L'idée est de considérer  $T_{TRT}$  non comme le temps cible pour le cycle du jeton mais comme le temps maximal.

Les principes de base du protocole sont:

- 1 on alloue à une station  $k$  une bande passante  $H_k$  pendant laquelle la station  $k$  transfère son trafic temps-réel (tant qu'elle en a) et ensuite, tant que l'allocation de bande passante  $H_k$  n'est pas terminée, transfère du trafic non-temps-réel (si elle en a); notons la différence fondamentale avec la définition de  $H_k$  dans le protocole jeton temporisé.
- 2 on contraint les allocations de bande passante aux  $m$  stations d'un réseau par la relation suivante qui définit la contrainte de protocole:

$$\sum_{k=1}^m H_k \leq T_{TRT} - t \quad (4.8)$$

Le temps de cycle est donc compris entre  $t$  et  $\left( t + \sum_{k=1}^m H_k \right)$  qui est borné par  $T_{TRT}$ .

Compte tenu du principe 1, on peut dire que ce protocole ne favorise pas le trafic non-temps-réel: d'une part, une station  $k$  ne peut en transmettre que si le trafic temps-réel de cette station n'a pas utilisé toute la bande  $H_k$  et, d'autre part, on n'a plus le mécanisme défini dans le protocole "jeton temporisé" (si le jeton arrive dans une station avant la fin du  $T_{TRT}$ , c'est-à-dire si le jeton est en avance, ce qui veut dire que les autres stations n'ont pas utilisée leur bande passante, la station peut transférer du trafic non-temps-réel).

La mise en oeuvre de ce protocole dans une station  $k$  est basée, d'une part, sur l'utilisation de deux compteurs ( $TRT_k$  et  $THT_k$  qui ont la même sémantique que dans le protocole "jeton temporisé") et, d'autre part, sur le temps de référence  $T_{TRT}$ .

L'algorithme du protocole est indiqué sur la procédure suivante:

```

Algorithme "jeton temporisé régulier"

Dans chaque station  $k$ ,  $k = 1, 2, \dots, m$ :
   $THT_k \leftarrow 0$ ;  $TRT_k \leftarrow T_{TRT}$ ; /*procédure d'initialisation */
  Démarrer  $TRT_k$  (compteur décrémental)

Dans chaque station  $k$ ,  $k = 1, 2, \dots, m$ :
  si  $TRT_k = 0$ : procédure "récupération de l'anneau" /* erreur*/
  À l'arrivée du jeton, faire: /* transfert de données */
     $THT_k \leftarrow H_k$ ;
     $TRT_k \leftarrow T_{TRT}$ ;
  Démarrer compteurs (décrémentaux)  $TRT_k$  et  $THT_k$ 
  tant que  $THT_k > 0$  et (messages temps-réel en attente):
    envoi messages temps-réel
  tant que  $THT_k > 0$  et (messages non-temps-réel en attente):
    envoi messages non-temps-réel
  passage du jeton à la station  $(k+1)$  (modulo  $k$ )

```

Notons que ici le rôle du compteur  $TRT_k$  est limité uniquement à un rôle de vérification que la contrainte de protocole est assurée (dans le cas contraire, on déclenche une procédure de "récupération de l'anneau").

### 3.1.2. Prise en compte des caractéristiques des flux de messages temps-réel

La garantie d'un délai d'accès borné à une station est une condition nécessaire mais non suffisante pour garantir le respect des échéances associées aux flux de messages de la station. Afin de garantir ce respect pour tout flux temps-réel  $M_i$  (période  $P_i$  et durée d'utilisation maximale du réseau  $C_i$ ), il faut que les deux contraintes suivantes (exprimées déjà pour le protocole "jeton temporisé") soient satisfaites:

- 1 le jeton passe dans la station pendant la période  $P_i$ ;
- 2 le temps minimum  $X_i$  dont dispose la station pour transmettre le message du flux  $M_i$  sur la totalité des passages du jeton pendant la période  $P_i$ , est suffisant pour assurer la durée nécessaire d'utilisation du réseau.

La satisfaction de ces contraintes dépend évidemment des spécificités du protocole "jeton temporisé régulier" et donc s'exprime par des relations différentes par rapport au cas du protocole "jeton temporisé".

La condition 1 implique que la période  $P_i$  soit supérieur au temps maximum de cycle ( $T_{TRT}$ ):

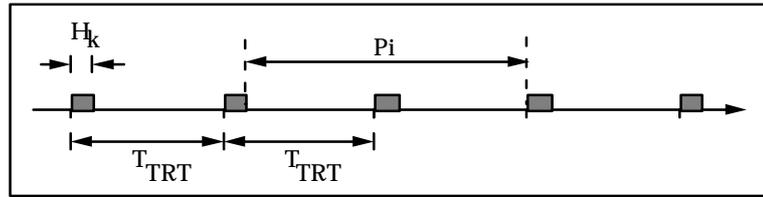
$$P_i \geq T_{TRT}, \quad \forall i: i = 1 \dots n \quad (4.9)$$

La condition 2 (qui traduit la contrainte de l'échéance) nécessite de déterminer le nombre minimum  $v_i$  de passages du jeton dans une station pendant une période  $P_i$  (on doit considérer l'asynchronisme entre les requêtes et les passages du jeton dans le cas pire à la fois de déphasage et de passage du jeton; c.f. figure 4.5).

On obtient:  $v_i = \lfloor P_i / T_{TRT} \rfloor$ , ce qui donne  $X_i = v_i \cdot H_i = \lfloor P_i / T_{TRT} \rfloor \cdot H_i$ , où  $H_i$  est l'allocation de bande pour le flux de messages  $M_i$  à chaque passage de jeton.

La contrainte de l'échéance est donc:

$$\lfloor P_i / T_{TRT} \rfloor \cdot H_i \geq C_i \quad \forall i=1 \dots n \quad (4.10)$$

Figure 4.5: Évaluation de  $v_i$  dans la station  $k$ 

### 3.1.3. Algorithmes d'allocation de bande passante

Nous proposons l'adaptation des algorithmes d'allocation de bande passante "durée complète" et "proportionnel adapté" définis pour le protocole "jeton temporisé" (c.f. III.4.4.1).

Cependant, nous considérons ici que l'on peut avoir plusieurs flux de messages dans une station  $k$  (la bande passante nécessaire à la station  $k$  ( $H_k$ ) lors d'un passage de jeton est la somme des bandes passantes nécessaires aux différents flux de la station  $k$  lors de ce passage:  $H_k = \sum_{n \in \text{neud}_k} H_i$ ) et nous généralisons donc les analyses faites avec le protocole "jeton temporisé" [ACZD, 92] [ACZ, 93] [ACZD, 94].

Concernant l'algorithme "proportionnel adapté", qui nécessite une fragmentation (contrairement à l'algorithme "durée complète"), nous l'analysons tout d'abord en négligeant le coût de la fragmentation (nécessite des en-têtes pour les fragments) et ensuite nous considérons ce surplus. A notre connaissance, une telle étude sur les aspects de la fragmentation n'a jamais été entreprise (voir les travaux présentés dans [ACZD, 92], [ACZ, 93], [ACZD, 94] et [MZ, 95]).

#### 1. Algorithme "durée complète"

On prend  $H_k = \sum_{n \in \text{neud}_k} C_i$ , c'est-à-dire, on transmet à l'arrivée d'un jeton tous les messages temps-réel en attente. Notons que la contrainte de l'échéance est automatiquement satisfaite pour chaque flux: en effet, il y a au moins un passage de jeton par période d'un flux ( $T_{TRT} \leq \min_i \{P_i\}$ ) et le message est transmis entièrement sur un passage.

#### Théorème:

*La borne minimale de l'utilisation maximale  $U_x$  est asymptotiquement nulle*

#### Démonstration

Considérons, réparti sur l'ensemble des stations, l'ensemble de flux de messages temps-réel  $M = \{M_1, M_2, \dots, M_n\}$  suivant:

- on a 1 flux  $M_1$  avec une période  $P_1 = T_{TRT}$  et une durée  $C_1$  infinitésimale:  $C_1 = \epsilon$  ( $\epsilon \rightarrow 0$ );
- on a  $n-1$  flux avec des périodes  $P_i$ , telles que  $P_i = T_{TRT}/\epsilon$  ( $\epsilon \rightarrow 0$ ), c'est-à-dire de longues périodes, et des durées identiques pour tous les flux  $C_i = C$ ;

Cet ensemble de flux, où le flux  $M_1$  sert à fixer la valeur de  $T_{TRT}$  ( $T_{TRT} \leq \min P_i$ ), représente un cas extrême de mauvaise efficacité pour l'algorithme (les  $n-1$  flux de période  $P_i = T_{TRT}/\epsilon$  (période très grande) n'utilisent qu'un seul passage de jeton sur un grand nombre de passages).

Le taux d'utilisation  $U$  du réseau pour cet ensemble de flux de messages est:

$$U = \sum_{i=1}^n \frac{C_i}{P_i} = \frac{\mathbf{e}}{T_{TRT}} + \frac{(n-1) \cdot C \cdot \mathbf{e}}{T_{TRT}} = \frac{\mathbf{e}}{T_{TRT}} \cdot (1 + (n-1) \cdot C) \quad \mathbf{e} \rightarrow 0 \quad (4.11)$$

La contrainte du protocole est:

$$\sum_{k=1}^m H_k = \sum_{k=1}^m \left\{ \sum_{n \in \text{eud}_k} C_i \right\} = \sum_{i=1}^n C_i = \mathbf{e} + (n-1) \cdot C \leq T_{TRT}(1-a)$$

Intégrons la contrainte du protocole dans l'expression du taux d'utilisation. Nous obtenons:  $U \leq \frac{\mathbf{e}}{T_{TRT}} (1 + (T_{TRT}(1-a) - \mathbf{e}))$  et, lorsque  $\mathbf{e} \rightarrow 0$ , on a  $U \rightarrow 0$ . Ceci montre que quand  $\mathbf{e} \rightarrow 0$ , la condition d'ordonnançabilité  $U \leq U_x$  n'est vraie que pour  $U_x = 0$ .

## Fin démonstration

Commentaire: En conséquence de l'utilisation maximale qui peut devenir asymptotiquement nulle, on est obligé pour ordonnancer toute configuration de messages de vérifier la validité des contraintes temporelles.

## 2. Algorithme "proportionnel adapté" sans prise en compte du coût de fragmentation

On prend  $H_k = \sum_{n \in \text{eud}_k} \frac{C_i}{v_i} = \sum_{n \in \text{eud}_k} \frac{C_i}{\lfloor P_i / T_{TRT} \rfloor}$ , c'est-à-dire on répartit la transmission de chaque message sur les  $v_i$  arrivées successives du jeton (donc on envoie des fragments de message). Notons que la contrainte de l'échéance est automatiquement satisfaite pour chaque flux car tous les fragments sont transmis au moyen du nombre minimal de passages de jeton par période  $P_i$  d'un flux: en effet ce nombre minimal est  $\lfloor P_i / T_{TRT} \rfloor$  et on transmet un fragment de durée  $\frac{C_i}{\lfloor P_i / T_{TRT} \rfloor}$  par passage de jeton.

### Théorème:

La borne minimale de l'utilisation maximale  $U_x$  est:  $U_x = (1-a)/2$

Afin de démontrer ce théorème, nous devons considérer le lemme suivant:

### Lemme

Avec le protocole "jeton temporisé régulier", compte tenu de la contrainte  $T_{TRT} \leq \min_i P_i$ , on a:

$$\frac{\lfloor P_i / T_{TRT} \rfloor}{P_i / T_{TRT}} \geq \frac{1}{2} \quad (4.12)$$

Nous démontrons tout d'abord le lemme et ensuite le théorème.

### Démonstration du lemme

Considérons le diagramme de la figure 4.6 qui visualise l'expression d'une période  $P_i$  d'un flux en fonction du nombre minimal d'arrivées du jeton pendant la période  $P_i$  ( $\lfloor P_i / T_{TRT} \rfloor$ ) et du  $T_{TRT}$ , en se plaçant dans le pire cas de déphasage entre les occurrences des messages de ce flux et les passages de jeton.

On voit que l'on a  $P_i = \lfloor P_i / T_{TRT} \rfloor \cdot T_{TRT} + d_i$ , avec  $0 \leq d_i < T_{TRT}$  et  $\lfloor P_i / T_{TRT} \rfloor$  étant un entier  $x$  supérieur ou égal à 1, soit  $\left\lfloor \frac{P_i}{T_{TRT}} \right\rfloor = \frac{P_i}{T_{TRT}} - \frac{d_i}{T_{TRT}}$ . On obtient:

$$\frac{\lfloor \frac{P_i}{T_{TRT}} \rfloor}{P_i/T_{TRT}} = \frac{\frac{P_i}{T_{TRT}} - \frac{d_i}{T_{TRT}}}{\frac{P_i}{T_{TRT}}} = 1 - \frac{d_i}{P_i} = 1 - \frac{d_i}{x \cdot T_{TRT} + d_i}$$

Le minimum de  $\frac{\lfloor \frac{P_i}{T_{TRT}} \rfloor}{P_i/T_{TRT}}$  est obtenu pour  $x = 1$  et vaut  $\frac{1}{2}$ .

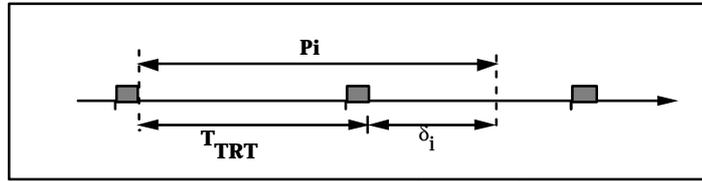


Figure 4.6: Relation entre  $P_i$ ,  $\lfloor P_i/T_{TRT} \rfloor$  et  $T_{TRT}$

### Fin démonstration (lemme)

### Démonstration (théorème)

Considérons la contrainte du protocole  $\sum_{k=1}^m \left\{ \sum_{n \in \text{eud}_k} \frac{C_i}{\lfloor \frac{P_i}{T_{TRT}} \rfloor} \right\} \leq T_{TRT} - t$ , qui peut (en considérant seulement l'ensemble des flux) encore s'écrire:  $\sum_{i=1}^n \frac{C_i}{\lfloor \frac{P_i}{T_{TRT}} \rfloor} \leq T_{TRT} - t$ .

En intégrant le lemme  $\frac{\lfloor \frac{P_i}{T_{TRT}} \rfloor}{P_i/T_{TRT}} \geq \frac{1}{2}$  dans la contrainte du protocole, on obtient la relation suivante:

$$\sum_{i=1}^n \frac{C_i}{P_i} \cdot 2 \cdot T_{TRT} \leq T_{TRT} - t$$

Soit comme  $U = \sum_{i=1}^n \frac{C_i}{P_i}$ , la relation précédente devient:  $U \leq \frac{1-a}{2}$ . Donc on a:  $U_x = \frac{1-a}{2} \rightarrow 50\%$  ( $a$  étant voisin de zéro).

### Fin démonstration (théorème)

Commentaire: ce résultat ( $U_x \rightarrow 50\%$ ) traduit le principal intérêt de ce protocole en terme de garantie d'ordonnabilité de configuration de messages temps-réel, et montre sa supériorité (du point de vue du trafic temps-réel) par rapport au protocole du jeton temporisé ( $U_x \rightarrow 33.3\%$ ). Dans le cas où  $a$  n'est pas voisin de zéro, il y a une dégradation de la borne  $U_x$ , quelque soit le protocole.

Intuitivement, on peut prévoir cette valeur de 50% pour  $U_x$  car le plus grand retard du jeton étant  $T_{TRT}$ , si la "périodicité" de tous les flux est de "presque  $2 \cdot T_{TRT}$ ", dans le pire cas chaque message n'aura qu'une arrivée du jeton pour être émis; Donc, comme la relation entre la "périodicité" des flux et le temps de cycle de  $\frac{1}{2}$ , l'utilisation maximale ne peut pas dépasser les 50%.

## 3. Algorithme "proportionnel adapté" avec prise en compte du coût de fragmentation

### Caractérisation de la fragmentation

Appelons  $j$  la durée de l'en-tête d'un fragment (nous considérons que la durée  $j$  ne dépend que de l'implémentation du protocole, c'est-à-dire  $j$  a une valeur constante qui ne

dépend pas de la longueur du fragment). Compte tenu qu'un message de durée d'utilisation du réseau  $C_i$  est fragmenté en  $v_i$  fragments, on peut définir la durée effective d'utilisation  $C_i^*$  du réseau pour ce message: on a  $C_i^* = C_i + v_i \cdot j$  avec  $v_i = \lfloor P_i/T_{TRT} \rfloor$ . La composante  $v_i \cdot j$  de  $C_i^*$  représente le surplus dû à la fragmentation.

### **Théorème:**

La borne minimale de l'utilisation maximale  $U_x$ , en considérant le coût de fragmentation est

$$U_x = \frac{1-a}{2} - \frac{n \cdot j}{2 \cdot T_{TRT}}, \text{ où } n \text{ représente le nombre de flux de messages}$$

### **Démonstration**

Reprenons la contrainte du protocole exprimée dans le paragraphe précédent:  $\sum_{i=1}^n \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} \leq T_{TRT} - t$  et, en remplaçant  $C_i$  par  $C_i^*$ , on obtient la contrainte du protocole avec la prise en compte du coût de fragmentation:

$$\sum_{i=1}^n \frac{C_i^*}{\lfloor P_i/T_{TRT} \rfloor} \leq T_{TRT} - t$$

Le membre de gauche de cette relation peut encore s'écrire en tenant compte, d'une part, de l'expression  $C_i^* (C_i + \lfloor P_i/T_{TRT} \rfloor \cdot j)$  et, d'autre part, du lemme  $\frac{\lfloor P_i/T_{TRT} \rfloor}{P_i/T_{TRT}} \geq \frac{1}{2}$ :

$$\sum_{i=1}^n \frac{C_i^*}{\lfloor P_i/T_{TRT} \rfloor} = \sum_{i=1}^n \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} + \sum_{i=1}^n \frac{\lfloor P_i/T_{TRT} \rfloor \cdot j}{\lfloor P_i/T_{TRT} \rfloor} = \sum_{i=1}^n \frac{C_i}{\lfloor P_i/T_{TRT} \rfloor} + n \cdot j \geq \sum_{i=1}^n \frac{C_i}{P_i} \cdot 2 \cdot T_{TRT} + n \cdot j$$

La contrainte du protocole peut donc être exprimée par la relation suivante:

$$2 \cdot T_{TRT} \cdot \sum_{i=1}^n \frac{C_i}{P_i} + n \cdot j \leq T_{TRT}(1-a), \text{ soit } U \leq \frac{1-a}{2} - \frac{n \cdot j}{2 \cdot T_{TRT}}. \text{ Donc on a: } U_x = \frac{1-a}{2} - \frac{n \cdot j}{2 \cdot T_{TRT}}$$

### **Fin démonstration**

#### *Commentaire*

La borne  $U_x$  devient nulle lorsque  $n \cdot j = T_{TRT}(1-a)$  (soit  $T_{TRT} \approx n \cdot j$ ,  $a$  étant voisin de zéro) ce qui est logique: ça traduit que les longueurs des en-têtes de tous les fragments occupent tout le temps du  $T_{TRT}$ . Cette expression est intéressante car elle montre bien, d'une part, l'influence de la fragmentation et, d'autre part, la nécessité de ne pas trop fragmenter si on veut avoir des valeurs de  $U_x$  supérieures à zéro.

## **3.2. Conditions d'ordonnançabilité du trafic temps-réel dans le réseau Profibus [VJ, 93] [VJ, 94]**

### **3.2.1. Réflexions sur l'algorithme d'arbitrage d'accès**

L'algorithme d'arbitrage d'accès du protocole Profibus est représenté dans la procédure suivante (nous remplaçons la notation utilisée dans [DIN, 91] par une notation équivalente à celle du jeton temporisé: les variables  $T_{TR}$ ,  $T_{TC}$ ,  $T_{TH}$  et  $T_{RR}$  [DIN, 91] sont remplacées par, respectivement,  $T_{TRT}$ ,  $t/m$ ,  $THT_k$  et  $TRT_k$ ):

**Algorithme d'arbitrage d'accès de Profibus**

```

Dans chaque station  $k$ , ( $k=1,2,\dots,m$ ):
   $THT_k \leftarrow 0$ ;                               /*procédure d'initialisation */
   $TRT_k \leftarrow 0$ ;
  Démarrer  $TRT_k$  (compteur incrémental)
Dans chaque station  $k$ , ( $k=1,2,\dots,m$ ), à l'arrivée du jeton, faire:
   $THT_k \leftarrow T_{TRT} - TRT_k$ ;
   $TRT_k \leftarrow 0$ ;
  Démarrer  $TRT_k$  (compteur incrémental)
  Si  $THT_k > 0$ :
    Démarrer  $THT_k$  (compteur décremental)
  (fin si)
  transfert "1" message "haute-priorité"
  Tant que  $THT_k > 0$  et (messages "haute-priorité" en attente)
    transfert messages "haute-priorité"
  Tant que  $THT_k > 0$  et (messages "basse-priorité" en attente)
    transfert messages "basse-priorité"
  passage du jeton à la station ( $k+1$ )

```

Ce protocole utilise des concepts que l'on retrouve dans le protocole du "jeton temporisé", à savoir le concept de  $T_{TRT}$  et les concepts de jeton en avance ( $THT_k > 0$ ) et de jeton en retard ( $THT_k < 0$ ).

La différence fondamentale avec le protocole du "jeton temporisé" est qu'il n'y a pas d'allocation de bande passante fixant les possibilités du trafic temps-réel (pas de spécification de  $H_k$ ), et par conséquent il n'y a pas de contrainte du protocole, c'est-à-dire, une relation entre les  $H_k$  et le  $T_{TRT}$ . Plus précisément, on peut voir que lorsque le jeton qui arrive dans une station  $k$  est en avance ( $THT_k > 0$ ), et si le trafic non-temps-réel dans cette station n'est pas contraint, on peut avoir des valeurs de  $THT_x < 0$  pour les stations suivantes ( $x = (k+1), \dots, m, 1, \dots, (k-1)$ ). Cela peut affecter les possibilités de transfert de trafic temps-réel dans ces stations, qui ne vont pouvoir transmettre qu'un seul message temps-réel. On a donc une influence du trafic non-temps-réel d'une station sur les possibilités en trafic temps-réel des stations suivantes. En conséquence, il est important de noter que si le trafic non-temps-réel n'est pas contraint, les possibilités en trafic temps-réel sont toujours réduites à un seul message temps-réel par arrivée du jeton dans une station et ceci quel que soit le choix du  $T_{TRT}$ . Donc, du point de vue strictement de la mise en oeuvre du trafic temps-réel, sur la base de 1 seul message par passage de jeton, on n'a pas de contrainte sur la valeur associée au  $T_{TRT}$ . Cette modalité de fonctionnement définit ce que nous appelons le profil originel ou profil 1 du réseau Profibus.

On peut cependant considérer un autre profil de fonctionnement, en supposant que toute station contrôle le débit associé à ses flux de messages non-temps-réel de façon à ce qu'elle puisse toujours transmettre tous les messages des flux temps-réel présents lors de l'arrivée d'un jeton. Cette modalité de fonctionnement, qui est basée sur la satisfaction d'une contrainte de protocole (le temps de cycle doit être tel que tous les besoins de trafic temps-réel, présents lors de l'arrivée d'un jeton, puissent être satisfaits (valeur de  $THT_k$  suffisante et donc temps de cycle inférieur à  $T_{TRT}$ )) définit ce que nous appelons le profil 2 du réseau Profibus. Il est important de remarquer que, par rapport au profil 1, on a maintenant une contrainte sur la valeur de  $T_{TRT}$ .

Nous évaluons maintenant ces deux profils par rapport aux possibilités de trafic temps-réel.

### 3.2.2. Évaluation du profil 1 (trafic non-temps-réel non contraint)

Considérons le cas pire des conséquences de ce profil, c'est-à-dire, une station qui utilise tout le temps défini par le  $T_{TRT}$ , ce qui amène ensuite les autres stations à ne pouvoir transmettre qu'un seul message temps-réel à l'arrivée du jeton.

Nous supposons que la durée maximale d'utilisation du réseau nécessaire pour chaque flux temps-réel est égale à la durée permise par le transfert du seul message temps-réel qu'une station peut envoyer quand elle reçoit le jeton (appelons  $C$  cette durée).

#### 1. Contrainte de l'échéance

Appelons  $T_{CYCLE_{max}}$  la valeur maximale du temps de cycle du jeton. Afin que la contrainte de l'échéance d'un flux de messages temps-réel  $M_{p_i}$  (période  $P_i$ ) appartenant à une station  $k$  soit satisfaite, il faut que le nombre de passages du jeton dans la station  $k$ , pendant la période  $P_i$ , soit supérieur ou égal à 1, c'est-à-dire on doit avoir la relation suivante:  $\lfloor P_i / T_{CYCLE_{max}} \rfloor \geq 1$ . Cette relation peut encore s'écrire comme  $1 / \lfloor P_i / T_{CYCLE_{max}} \rfloor \leq 1$ , où  $1 / \lfloor P_i / T_{CYCLE_{max}} \rfloor$  représente le taux d'utilisation des passages du jeton pendant la période  $P_i$ , ce qui traduit encore les besoins du service du jeton pour le flux  $P_i$  appartenant à la station  $k$ .

En conséquence, les échéances de tous les flux appartenant à une station  $k$  seront satisfaites si les besoins de service du jeton sont inférieurs ou égaux à la capacité de service des passages du jeton dans la station  $k$ , c'est-à-dire:

$$\sum_{M_i \in \text{nœud}_k} \frac{1}{\lfloor P_i / T_{CYCLE_{max}} \rfloor} \leq 1, \quad \forall k, 1 \leq k \leq m \quad (4.13)$$

Cette formule traduit une contrainte sur la valeur de  $T_{CYCLE_{max}}$ , valeur qui est fixée par la station  $k$  qui possède le flux avec la plus petite période  $P_i$  et qui dépend du nombre total de flux que possède cette station  $k$ . Plus précisément, la valeur de  $T_{CYCLE_{max}}$  est d'autant plus faible par rapport à la plus petite période  $P_i$  qu'il y a un nombre important de flux dans la station  $k$ . Par voie de conséquence, on aura dans le cas le plus général, des passages de jetons dans des stations qui fonctionneront "à vide" (pas de message à transmettre).

#### 2. Autre contrainte sur la valeur $T_{CYCLE_{max}}$

##### Théorème:

La valeur maximale du temps de cycle  $T_{CYCLE_{max}}$  est:

$$T_{CYCLE_{max}} = T_{TRT} + t + (m - 1) \cdot C \quad (4.14)$$

où  $t$  est le temps de propagation du jeton sur l'anneau logique,  $C$  est la durée maximale d'un message temps-réel et  $m$  est le nombre de stations.

##### Démonstration:

Supposons que dans le cycle  $(l+1)$  le jeton arrive au plus tôt à la station  $k$ , c'est-à-dire, aucun trafic n'est transmis par aucune station pendant le cycle de jeton précédent (figure 4.7); donc,  $TRT_k = t$  et  $THT_k$  est initialisé à  $T_{TRT} - t$ . Dans ce cas, la station peut transférer du trafic pendant  $T_{TRT} - t$  unités de temps, jusqu'à ce que son  $THT_k$  soit écoulé.

Cependant, si la station  $k$  laisse écouler son  $THT_k$ , la station  $k+1$  recevra le jeton à l'instant où  $TRT_{k+1} = T_{TRT}$  et donc  $THT_{k+1}$  sera initialisé à zéro (car si  $t_{k+1}(l) - t_k(l) = t/m$  et  $t_{k+1}(l+1) - t_k(l) = T_{TRT} + t/m$ , alors  $TRT_k = t_{k+1}(l+1) - t_{k+1}(l) = T_{TRT}$  et donc la valeur de  $THT_{k+1}$  est nulle).

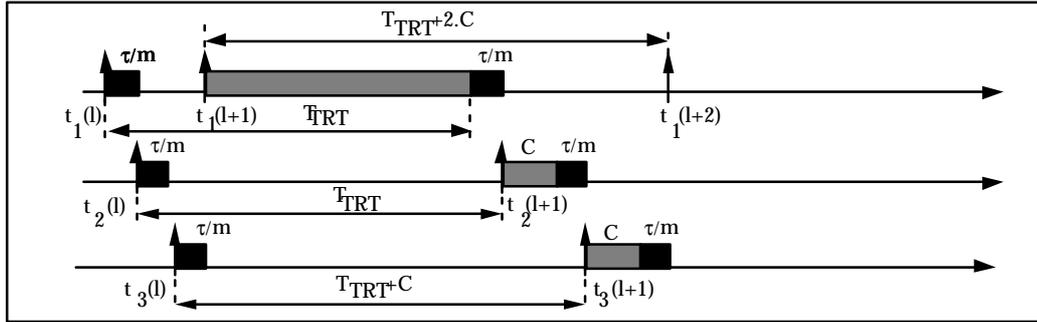


Figure 4.7: Évaluation du temps maximal de cycle du jeton

Si, à l'arrivée du jeton à la station  $k+1$ , la valeur de  $THT_{k+1}$  est nulle, la station n'a le droit de transférer qu'un seul message; par conséquent, à l'arrivée du jeton sur la station suivante:  $TRT_{k+2} = T_{TRT} + C$ , et ainsi de suite. Dans ce cas,  $T_{CYCLE_{max}} = T_{TRT} + (m-1) \cdot C$ .

Dans la figure 4.7, nous présentons le cas où  $k=3$  et donc  $TRT_{max} = T_{TRT} + 2 \cdot C$ .

Cependant, ceci n'est pas le temps de cycle maximal car, dans le cas particulier du démarrage du système, les  $TRT_k$ 's démarrent simultanément et donc le temps de cycle maximal est  $T_{CYCLE_{max}} = T_{TRT} + t + (m-1) \cdot C$  (figure 4.8).

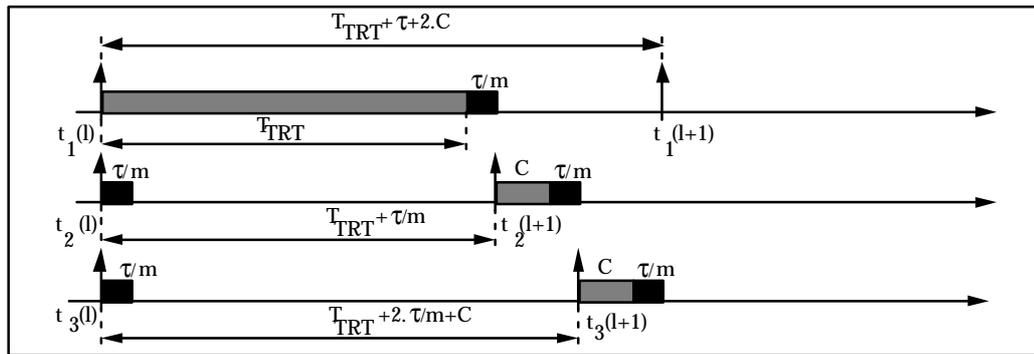


Figure 4.8: Évaluation du temps maximal de cycle du jeton

### 3.2.3. Évaluation du profil 2 (trafic non-temps-réel contraint)

Considérons  $m$  stations,  $n$  flux de messages temps-réel,  $m$  flux de messages non-temps-réel et  $T_{CYCLE}$  la durée d'un cycle. Nous exprimons, tout d'abord, la contrainte de protocole à partir de laquelle nous obtenons la valeur maximale du temps de cycle et le délai d'accès. Ensuite, nous donnons la condition pour que la contrainte de l'échéance soit satisfaite.

#### 1. Contrainte de protocole

Afin que tous les messages présents lors de l'arrivée du jeton dans les stations puissent être transmis, on doit avoir la relation suivante:

$$\sum_{i=1}^n \left\{ C_i \cdot \left\lceil \frac{T_{CYCLE}}{P_i} \right\rceil \right\} + \sum_{j=1}^m \left\{ C_j \cdot \left\lceil \frac{T_{CYCLE}}{P_j} \right\rceil \right\} + t + \max_{k=1 \dots m} \left\{ \sum_{n \in \text{eud}_k} C_i \right\} \leq T_{TRT} \quad (4.15)$$

où le premier terme et le deuxième terme représentent les bornes maximales, respectivement, du trafic temps-réel et du trafic non-temps-réel sur un cycle; la somme de ces deux termes représente la charge maximale sur un cycle. Le troisième terme représente toujours le temps de propagation du jeton sur l'anneau logique plus la durée du jeton. La somme de ces trois premiers termes représente la durée du temps de cycle. Le quatrième terme représente la

borne maximale du trafic temps-réel qu'une station doit transmettre lorsque le jeton lui revient après un cycle (ce terme représente la valeur maximale que doit avoir le  $THT$  de la station lorsque le jeton lui revient).

## 2. Contrainte de l'échéance

Comme à l'arrivée du jeton dans une station, on a toujours le temps suffisant pour transférer tous les messages temps-réel en attente, la garantie de l'échéance d'un flux de période  $P_i$  est obtenue si le jeton arrive à la station au moins une fois pendant la période  $P_i$ .

$$P_i \geq T_{CYCLE_{max}}, \quad \forall i, 1 \leq i \leq p \quad (4.16)$$

C'est donc le flux de plus petite période ( $\min_i \{P_i\}$ ) qui représente la borne supérieure de  $T_{CYCLE_{max}}$ .

## 3. Expression de $T_{CYCLE_{max}}$

Compte tenu de la contrainte de l'échéance (qui implique que  $\lceil T_{CYCLE}/P_i \rceil = 1, \forall i$ ) et en considérant également que  $\lfloor T_{CYCLE}/P_j \rfloor = 1, \forall j$ , on obtient l'expression de la durée de  $T_{CYCLE_{max}}$  :

$$T_{CYCLE_{max}} = \sum_{i=1}^n C_i + \sum_{j=1}^m C_j + t \quad (4.17)$$

Étant donnée la contrainte de l'échéance et l'expression de  $T_{CYCLE_{max}}$ , on doit encore avoir l'expression suivante:

$$\sum_{i=1}^n C_i + \sum_{j=1}^m C_j + t \leq \min_i \{P_i\} \quad (4.18)$$

Il est important de remarquer, à partir de cette dernière relation, que si presque tous les besoins de trafic doivent être transmis sur un temps borné par  $\min_i \{P_i\}$ , on aura ici aussi des passages de jetons dans des stations qui fonctionnent "à vide" (stations où on a des flux de périodes  $P_i$  grandes par rapport à la valeur de la plus petite période).

## 4. Contrainte sur la valeur de $T_{TRT}$

En considérant, à la fois, la contrainte de protocole et la valeur de  $T_{CYCLE_{max}}$ , on obtient:

$$T_{CYCLE_{max}} + \max_{k=1..m} \left\{ \sum_{n \in \text{œud}_k} C_i \right\} \leq T_{TRT} \quad (4.19)$$

### 3.2.4. Exemple d'application

Nous considérons le système suivant: on a  $m$  stations ( $m = 10; t = 1$ ), qui ont chacune 3 flux temps-réel identiques  $\{C_j, P_j\} = \{1, 100), (1, 100), (C^*, 400)\}$ , la durée  $C^*$  du dernier flux étant variable mais borné inférieurement par 1, c'est-à-dire  $C^* > 1$  (notre objectif étant de voir les bornes maximales admissibles pour la durée  $C^*$  dans chaque profil); on a aussi, sur deux stations, 2 flux non-temps-réel:  $\{C_j, P_j\} = \{(5, 500), (5, 500)\}$ .

#### **Profil 1:**

Contrainte de l'échéance:  $\sum_{M_i \in \text{œud}_k} \frac{1}{\lfloor P_i/T_{CYCLE_{max}} \rfloor} \leq 1, \quad \forall k, 1 \leq k \leq m$ ; en développant, nous avons

que pour chaque station,  $\frac{2}{\lfloor 100/T_{CYCLE_{max}} \rfloor} + \frac{1}{\lfloor 400/T_{CYCLE_{max}} \rfloor} \leq 1$ , et donc il faut que

$\lfloor 100/T_{CYCLE_{max}} \rfloor > 2$ , c'est-à-dire  $T_{CYCLE_{max}} \leq 33.3$ .

Temps de cycle:  $T_{CYCLE_{max}} \leq T_{TRT} + t + (m-1) \cdot C$ , c'est-à-dire:  $T_{CYCLE_{max}} \leq T_{TRT} + 1 + 9 \cdot C^*$

Sachant que le temps de cycle est borné par  $T_{CYCLE_{max}} \leq T_{TRT} + 1 + 9 \cdot C^*$ , pour que la contrainte de l'échéance soit elle aussi satisfaite il faut que  $T_{CYCLE_{max}} \leq (T_{TRT} + 1 + 9 \cdot C^*) \leq 33.3$ , c'est-à-dire, il faut que  $T_{TRT} \leq 32.3 - 9 \cdot C^*$ .

Si on diminue  $T_{TRT}$  jusqu'à la valeur nulle, la durée  $C^*$  peut donc, augmenter jusqu'à 3.6; néanmoins, si on choisi  $T_{TRT} = 0$ , le trafic non-temps-réel ne pourra plus utiliser la largeur de bande laissée disponible par le trafic temps-réel.

### **Profil 2:**

Contrainte de l'échéance:  $P_i \geq T_{CYCLE_{max}}$ ; donc,  $T_{CYCLE_{max}} = 100$ ;

Contrainte de protocole:  $\sum_{i=1}^n \left\{ C_i \cdot \left\lceil \frac{T_{CYCLE}}{P_i} \right\rceil \right\} + \sum_{j=1}^m \left\{ C_j \cdot \left\lceil \frac{T_{CYCLE}}{P_j} \right\rceil \right\} + t + \max_{k=1, \dots, m} \left\{ \sum_{n \in \text{eud}_k} C_i \right\} \leq T_{TRT}$ ;

sachant que  $T_{CYCLE} \leq P_i$ , la contrainte de protocole impose  $T_{TRT} \geq 11 \cdot (2 + C^*) + 11$ .

$T_{CYCLE_{max}}$ : Il faut que  $\sum_{i=1}^n C_i + \sum_{j=1}^m C_j + t \leq \min_i \{P_i\}$ , c'est-à-dire:  $10 \cdot (2 + C^*) + 10 + 1 \leq 100$ , ce qui impose  $C^* \leq 6.9$ .

La valeur de  $T_{TRT}$  découle de la contrainte du protocole:  $T_{TRT} \geq 108.9$ .

On peut donc, sur cet exemple, dire que pour le profil 1, l'ensemble de flux de messages est ordonnançable tant que  $C^* \leq 3.6$  (pour  $T_{TRT} = 0$ ), ce qui correspond à un taux d'utilisation des flux de messages temps-réel qui ne dépasse pas les 29%. Par contre, le profil 2 peut ordonnancer des flux de messages avec  $C^* \leq 6.9$  ( $T_{TRT} \geq 108.9$ ), ce qui correspond à un taux d'utilisation pour les flux de messages temps-réel nettement supérieur:  $U = 37.25\%$ .

En ce qui concerne le délai d'accès au réseau, dans les cas du profil 1 et 2, celui-ci est borné à, respectivement,  $T_{CYCLE_{max}} = 33.3$  et  $T_{CYCLE_{max}} = 100$ , ce qui correspond à un délai d'accès plus intéressant dans le cas du profil 1.

### **3.2.5. Conclusion**

Nous avons proposé deux profils de fonctionnement, qui correspondent à deux types de comportements des sources de trafic non-temps-réel (comportement non contraint (profil 1) et comportement contraint (profil 2)) et nous avons donné pour chaque profil des conditions d'ordonnançabilité du trafic temps-réel.

Le profil 1 est caractérisé par un temps de cycle faible (ce qui permet des temps d'accès très courts pour les messages les plus prioritaires) et impose des durées courtes pour les messages.

Le profil 2 a un temps de cycle plus long et permet des durées de messages plus élevées.

## **4. Conclusion**

Nous voulons souligner les points suivants:

- concernant les contributions sur les protocoles de classe 1:
  - nous avons conceptualisé un algorithme de changement en ligne de la séquence temporelle de flux périodiques ordonnancés par l'algorithme RM pour le réseau DQDB et donné, d'une part, les conditions pour effectuer des changements corrects (ni perte de message, ni duplication de message) et, d'autre part, le temps maximum

---

pour rendre effectif un changement de mode (aspect très important dans un contexte temps-réel);

- nous avons exprimé des conditions pour l'ordonnançabilité, dans le pire cas, du service du trafic aperiodique urgent dans le réseau FIP en termes, d'une part, d'une contrainte de protocole afin de garantir une non surcharge du macro-cycle et, d'autre part, d'une contrainte de l'échéance basée sur des notions de retard d'avertissement de l'arbitre du bus et de retard d'ordonnement. Ces expressions montrent que l'efficacité de ce service est faible (ce qui découle de la prépondérance accordée au trafic périodique dans le réseau FIP);
- concernant les contributions sur les protocoles de classe 2:
  - nous avons défini un protocole appelé "jeton temporisé régulier" qui, par rapport au protocole du "jeton temporisé" présente une régularité de cycle (le temps de cycle est borné par  $T_{TRT}$ , alors que la borne est de  $2 \cdot T_{TRT}$  dans le "jeton temporisé") ce qui permet une meilleure borne minimale de l'utilisation maximale avec l'algorithme d'allocation de bande passante proportionnel adapté (50% par rapport à 33.3%) et donc présente un intérêt supérieur pour le trafic temps-réel; nous avons également évalué le coût de la fragmentation inhérente à l'algorithme proportionnel adapté.
  - nous avons défini deux profils de fonctionnement pour le réseau Profibus, à savoir le profil 1 (on considère que le trafic non-temps-réel n'est pas contraint) et le profil 2 (on considère que le trafic non-temps-réel est contraint), et nous avons défini des conditions d'ordonnançabilité pour ces deux profils; le profil 1 est caractérisé par un temps de cycle faible (ce qui permet des temps d'accès très courts pour les messages les plus prioritaires) et impose des durées courtes pour les messages; le profil 2 a un temps de cycle plus long mais contraint moins les durées des messages.

## 5. Références

- [ACZD, 92] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines in High-Speed Token-Ring Networks with the Timed Token Protocol; in Proc. of 12th IEEE Distributed Computing Systems, Yokohama, pages 468-475, 1992;
- [ACZ, 93] G. Agrawal, B. Chen and W. Zhao; Local Synchronous Capacity Allocation Schemes for Guaranteeing Message Deadlines with the Timed Token Protocol; Proc. of INFOCOM'93
- [ACZD, 94] G. Agrawal, B. Chen, W. Zhao, S. Davari; Guaranteeing Synchronous Messages Deadlines with the Timed Token Medium Access Control Protocol; in IEEE Transactions on Computers, vol. 43, no 3, pages 327-339, March 1994;
- [CVJ, 94] R. Carmo, F. Vasques, G. Juanole; Real-Time Communication Services in a DQDB Network; IEEE RTSS'94, pages 249-258, S. Juan, 1994
- [JCV, 94] G. Juanole, R. Carmo, F. Vasques; Connection Oriented Isochronous Services in a DQDB Network: Specification of a Service-Protocol pair and a Bandwidth Allocation Scheme; Proc. of International Conference on Local and Metropolitan Communication Systems, pages 377-396, Kyoto, Japan, 1994
- [DIN, 91] DIN 19245 Part 1 & 2: Process Field Bus (English translation of German version); April 1991

- [Gro, 82] R. Grow; A Timed Token Protocol for Local Area Networks; IEEE Electro'82, Token Access Protocols, 17/3, 1982
- [IEEE, 90] Distributed Queue Dual Bus (DQDB) Subnetwork of Metropolitan Area Network, IEEE Std 802.6 - 1990
- [LL, 73] C. Liu, J. Layland; Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment; Journal of the ACM 20 (1), pages 46-61, 1973;
- [MZ, 95] N. Malcolm, W. Zhao; Hard Real-Time Communication in Multiple-Access Networks; The Journal of Real-Time Systems, 8(1), pages 35-77, 1995;
- [NFa, 90] FIP Bus for Exchange of Information between Transmitters, Actuators and Programmable Controllers, Data Link Layer; NF C 46-603, June 1990
- [VJ, 93] F. Vasques, G. Juanole; Fieldbus MAC Mechanisms for Hard Real-Time Data Communication Support; proc. of OBS'93: Open Bus Systems, pages 225-232; Munich, Germany, November 1993.
- [VJ, 94] F. Vasques, G. Juanole; Pre-Run-Time Schedulability Analysis in Fieldbus Networks; proc. of IECON'94, pages 1200-1204; Bologna, Italy, September 1994.

