

## Chapitre III

# Réseaux de communication temps-réel: les principaux protocoles MAC

### 1. Introduction

Le but de ce chapitre est de présenter les principaux protocoles MAC qui peuvent être utilisés dans un contexte temps-réel (c'est-à-dire, qui permettent la mise en oeuvre d'un ordonnancement de transfert de flux de messages contraints temporellement).

En ce qui concerne ces protocoles, les présentations traditionnelles se focalisent principalement sur les "techniques d'accès à la ressource de transmission" et passent généralement sous silence l'aspect "ordonnancement du transfert de flux de messages à travers la ressource de communication". Ce dernier aspect est absolument essentiel, et plus particulièrement dans un contexte temps-réel, si on veut spécifier et concevoir de manière rationnelle un protocole MAC. C'est ce souci qui va nous guider dans la présentation proposée, afin de bien mettre en évidence les différences entre les protocoles existants.

Ce chapitre comprend cinq parties:

- La première partie concerne une classification de protocoles MAC temps-réel en deux grandes classes: classe 1 et classe 2;
- La deuxième partie présente les principaux protocoles de classe 1;
- La troisième partie présente les principaux protocoles de classe 2;
- La quatrième partie concerne la présentation d'un protocole intégrant les caractéristiques de la classe 1 et de la classe 2;
- La cinquième partie présente une comparaison des protocoles de classe 1 et de classe 2;

## 2. Classification des protocoles MAC temps-réel

### 2.1. La problématique de l'ordonnancement du transfert de flux de messages temps-réel

Une classification des protocoles MAC temps-réel a été proposée dans [MZ, 95]. Dans ce travail les auteurs considèrent d'abord que l'ordonnancement du transfert des messages doit être vu sous l'angle d'une séquence de deux processus:

- un processus *d'arbitrage d'accès* qui détermine *quand* la station a le droit d'utiliser la ressource de transmission;
- un processus de *contrôle de transmission* qui détermine *combien de temps* la station a le droit d'utiliser la ressource de transmission;

Ensuite, une classification des protocoles MAC temps-réel est proposée selon la prépondérance de chacun de ces processus sur l'ordonnancement des messages; deux approches sont définies:

- une approche basée sur l'arbitrage d'accès, où l'accent est mis sur le processus d'arbitrage d'accès;
- une approche basée sur le contrôle de transmission, où l'accent est mis sur le processus de contrôle de transmission.

Un problème évident associé à cette classification réside dans le cas où aucun des deux processus est prépondérant vis-à-vis de l'autre (c'est le cas du protocole IEEE 802.3 DCR (c.f. III.2.4.2)).

À notre avis, la prépondérance relative de ces deux processus ne doit pas être le seul paramètre choisi pour effectuer une classification de protocoles de MAC temps-réel.

Nous proposons une autre classification des protocoles MAC temps-réel basée non sur la prépondérance relative de ces deux processus, mais sur la *mise en oeuvre* de l'ordonnancement du transfert des messages.

Nous élargissons d'abord la définition de ces deux processus, en considérant séparément les notions de flux de messages et de station:

#### Définition

- *Le processus d'arbitrage d'accès détermine quand le flux de messages ou la station a le droit d'utiliser la ressource de transmission.*
- *Le processus de contrôle de la durée de transmission détermine combien de temps le flux de messages ou la station a le droit d'utiliser la ressource de transmission*

Si on observe les systèmes existants, on constate que les protocoles MAC temps-réel mettent en oeuvre des mécanismes qui travaillent, soit sur l'accès des flux de messages, soit sur l'accès des stations (plus précisément, qui garantissent un temps d'accès borné aux stations).

En conséquence, nous proposons la définition de deux grandes classes de protocoles MAC temps-réel (figure 3.1), selon que l'ordonnancement est mis en oeuvre sur les flux de messages ou sur les stations:

- La classe 1 qui est relative aux protocoles réalisant un *ordonnancement basé sur une assignation de priorité aux flux de messages* (priorité traduisant les contraintes temporelles);

les protocoles MAC de la classe 1 travaillent à partir des contraintes temporelles des flux de messages, qu'ils interprètent et qu'ils utilisent pour réaliser l'ordonnancement (ordonnancement global) et les transferts.

- La classe 2 qui est relative aux protocoles réalisant un *ordonnancement basé sur la notion d'une garantie d'un temps d'accès borné aux entités MAC (donc aux stations)*; dans ce cas, le protocole MAC offre simplement un service d'accès, en temps borné et en exclusion mutuelle, à la ressource de transmission; ce service doit être utilisé par le(s) niveau(x) supérieur(s) au niveau MAC, qui doit (doivent) mettre en oeuvre l'ordonnancement des flux de messages pendant le temps d'accès (ordonnancement local).

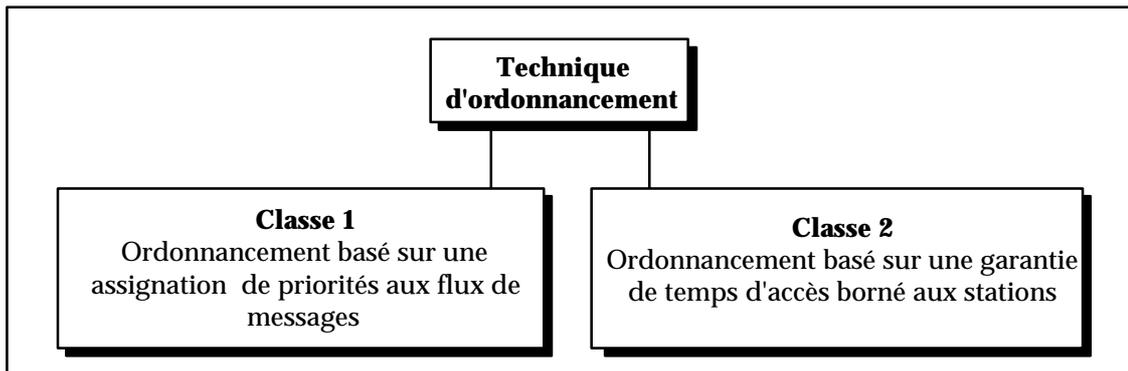


Figure 3.1: Deux classes de protocoles

La classe 1 est, à notre avis, vraiment représentative de ce que les protocoles MAC dits "temps-réel" doivent être. De toute manière, quelle que soit la classe de protocole que l'on considère comme point de départ, c'est-à-dire l'ordonnancement des flux de messages ou l'ordonnancement des stations, on doit considérer cet ordonnancement comme faisant intervenir les deux processus suivants:

- *L'arbitrage d'accès* (accès de flux de messages pour la classe 1; accès de stations pour la classe 2);
- *Le contrôle de la durée de transmission* (pour un flux de messages dans la classe 1; pour une station dans la classe 2).

## 2.2. L'arbitrage d'accès et le contrôle de la durée de transmission

### 2.2.1. Classe 1

- L'arbitrage d'accès consiste en deux mécanismes:
  - Un mécanisme *d'assignation de priorités* à chaque flux de messages (assignation statique) ou à chaque message individuel (assignation dynamique).
  - Un mécanisme global *d'arbitrage de priorités* qui détermine, en-ligne ou hors-ligne, le flux de message ou le message qui accède à la ressource de transmission.
- Le contrôle de la durée de transmission est, en général, très simple: on transfère un message par accès arbitré.

L'opération générale de l'arbitrage d'accès dans les protocoles de "classe 1" est équivalente à celle de l'opération d'un ordonnanceur de tâches dans un système monoprocesseur. L'équivalence se traduit par l'existence d'une seule ressource à partager: *processeur/réseau*, parmi plusieurs utilisateurs: *tâches/flux de messages*. Compte tenu de la similarité de ces deux systèmes et du grand nombre de travaux faits sur l'ordonnancement monoprocesseur temps-

réel, on a assisté ces dernières années, à un intérêt croissant pour l'adaptation des algorithmes classiques d'ordonnancement de tâches à l'ordonnancement temps-réel des messages dans les réseaux de communication.

Dans le cas de l'arbitrage d'accès en-ligne, on a simultanément la mise en oeuvre de l'arbitrage d'accès et la mise en oeuvre effective du transfert. De plus, la mise en oeuvre du transfert est transparente aux évolutions des caractéristiques des flux de messages à ordonnancer (c'est-à-dire des modes de fonctionnement).

Dans le cas de l'arbitrage d'accès hors-ligne, on doit distinguer deux activités au sein d'un mode de fonctionnement:

- Une première activité qui, à partir de la connaissance des priorités, évalue la séquence des flux de messages à transférer;
- Une deuxième activité qui consiste en la mise en oeuvre effective du transfert de la séquence des flux de messages.

La stratégie hors-ligne n'est pas transparente aux changements de mode, et il faut donc prévoir, en-ligne, un mécanisme de changement de mode de fonctionnement.

En ce qui concerne le trafic périodique, l'ordonnancement hors-ligne est toujours plus compétitif que celui effectué en-ligne [B\_all, 92]. Nous pouvons citer les avantages suivants par rapport à un ordonnancement en-ligne: taux d'utilisation élevé; prédiction de la séquence temporelle du trafic; garantie de non surcharge du réseau; coût de fonctionnement réduit; propagation de fautes temporelles limitée.

En ce qui concerne le trafic aperiodique, comme l'ordonnanceur hors-ligne n'a qu'une connaissance très limitée sur le futur, sa performance sera toujours inférieure à celle d'un ordonnanceur en-ligne [B\_all, 92]. Donc, ce type de trafic ne peut être pris en compte qu'en utilisant des moyens de réservation de bande passante peu efficaces. De plus, la mise en oeuvre de moyens de réservation de bande passante, requiert des mécanismes d'arbitrage supplémentaires, ce qui réduit encore leur efficacité.

### **2.2.2. Classe 2**

- L'arbitrage d'accès a pour but de permettre l'accès de manière équitable aux différentes stations.
- Le contrôle de la durée de transmission peut être très élémentaire (un message transmis par station), mais peut être également très sophistiqué: en effet, toute station doit pouvoir disposer d'un temps de transmission suffisant pour transmettre les flux de messages temps-réel venant du niveau supérieur, et donc il est important d'utiliser des techniques d'allocation de bande passante qui permettent de garantir le respect des contraintes temporelles associées aux flux de messages des différentes stations. Cette allocation de bande est effectuée par l'intermédiaire de deux mécanismes:
  - Un mécanisme de gestion globale qui gère le compromis de la répartition d'une ressource potentiellement insuffisante, c'est-à-dire le temps d'utilisation du réseau, parmi un ensemble de stations avec des besoins temps-réel différents;
  - Un mécanisme de gestion locale qui évalue d'abord les besoins particuliers de chaque station pour en informer la gestion globale, et ensuite gère la répartition de l'allocation de bande parmi les flux de messages appartenants à la station, de façon à garantir leurs contraintes temporelles.

### 2.3. Mise en oeuvre de l'arbitrage d'accès et du contrôle de la durée de transmission: les techniques d'accès

La présentation des techniques d'accès est la manière traditionnelle de parler des protocoles MAC (sans faire référence aux aspects d'ordonnancement qu'ils réalisent).

Nous indiquons sur la figure 3.2, les principales techniques d'accès:

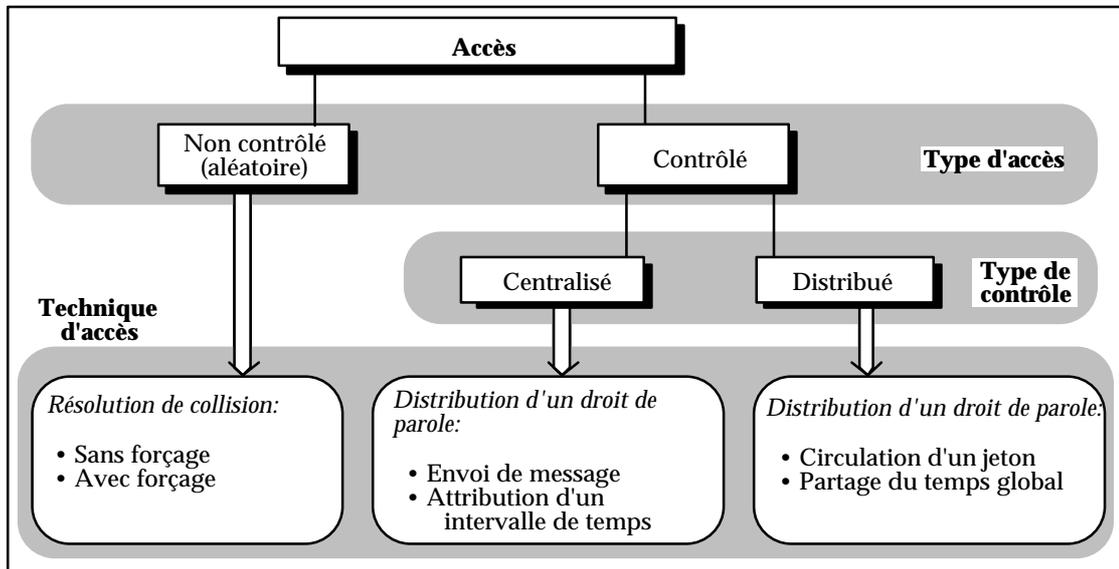


Figure 3.2: Les techniques d'accès

- L'accès non-contrôlé (aléatoire) est une technique dite de compétition qui génère, par définition, des collisions.

Deux variantes, à des finalités temps-réel, de cette technique ont été définies:

- la résolution de collision sans forçage.
- la résolution de collision avec forçage.
- L'accès à contrôle centralisé est basé sur l'existence d'une station de contrôle qui distribue un droit de parole aux différentes stations. On distingue deux variantes:
  - soit la station de contrôle envoie à chaque station un message qui lui donne le droit d'utiliser le réseau;
  - soit la station de contrôle joue le rôle d'un horloge qui définit des intervalles de temps que les stations peuvent utiliser.
- L'accès à contrôle distribué est basé sur une coopération entre toutes les stations afin de définir laquelle a le droit de parole, c'est-à-dire, le droit d'utiliser le réseau. On distingue également deux variantes:
  - la circulation d'un jeton (technique de jeton circulant) que les stations se transmettent; le jeton est un mécanisme de coopération explicite;
  - la technique de partage du temps global, qui est basée sur l'hypothèse que chaque station a la connaissance du temps global et des intervalles de temps où elle peut utiliser le réseau; c'est un mécanisme de coopération implicite.

### 3. Protocoles de la "classe 1"

#### 3.1. Principales normes

Les principales normes de "classe 1" existantes sont représentées sur la figure 3.3.

Ces normes sont le plus généralement spécifiées en décrivant la mise en oeuvre de la technique d'accès, mais en passant sous silence l'aspect ordonnancement et, par voie de conséquence, le lien que l'on doit faire entre la technique de l'ordonnancement et la technique d'accès.

Nous présentons maintenant les éléments essentiels de l'arbitrage d'accès en mettant précisément en exergue l'aspect ordonnancement (avec les principaux travaux) et le lien avec la technique d'accès.

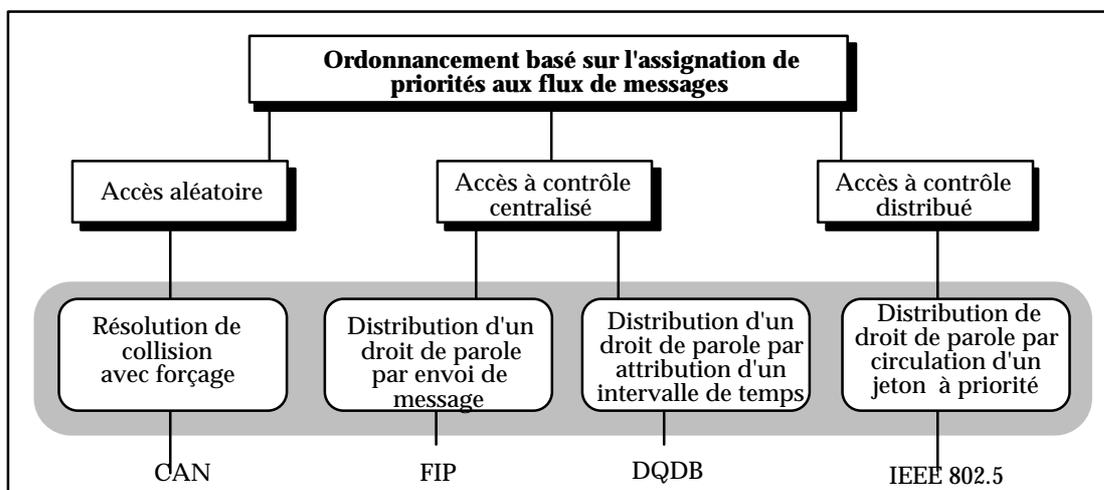


Figure 3.3: Les protocoles "classe 1" existants

#### 3.2. Le protocole CAN

##### 1. Généralités

Le protocole CAN [SAE, 92] a été proposé par Bosch dans le cadre des applications automobiles embarquées. Une caractéristique très intéressante de la norme CAN est que l'on spécifie un système de flux de messages indépendants ( $2^{11}$  ou  $2^{29}$  flux différents, selon la version de la norme), chacun possédant une priorité unique dans le système. Ceci offre un cadre pour pouvoir définir des protocoles qui implantent le contrôle de l'accès à la ressource de transmission sur la base de l'interprétation de ce schéma de priorités.

##### 2. Assignation de priorités

Un schéma d'assignation de priorités à des flux de messages périodiques, basé sur une adaptation de l'algorithme DM, a été proposé dans [TWH, 94]. Ce schéma de priorité, au lieu d'être borné sur les échéances, est basé sur des fenêtres de transmission qui prennent en compte la gigue maximale qui peut affecter l'arrivée dans les files d'attente de la sous-couche MAC des messages des flux périodiques.

Sur la figure 3.4, nous représentons la fenêtre de transmission des messages d'un flux  $M_i$  de période  $P_i$ , compte tenu que la gigue maximale est  $J_i$  et l'échéance associée aux messages est  $d_i$ . La fenêtre de transmission est  $(d_i - J_i)$ .

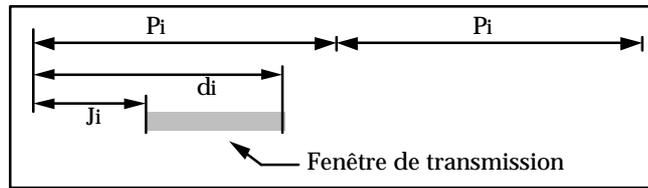


Figure 3.4: Fenêtre de transmission des messages

Dans leurs travaux, les auteurs proposent une assignation de priorités statiques inversement proportionnelle aux durées des fenêtres de transmission et déterminent la borne supérieure  $t_i$  du délai d'accès d'un message du flux  $M_i$ . La borne  $t_i$  est [TWH, 94]:

$$t_i = \max_{i+1 \leq j \leq n} C_j + \sum_{1 \leq j \leq i-1} \left\lceil \frac{t_j + J_j + t_{bit}}{P_j} \right\rceil \cdot C_j \quad (3.1)$$

où le premier terme traduit l'aspect non-préemptif de la transmission et le deuxième terme traduit l'interférence des messages des flux  $M_j$  de plus haute priorité (compte tenu de la gigue  $J_j$  de ces flux, et de la durée d'un bit  $t_{bit}$  qui est l'unité de mesure de temps). Une condition suffisante d'ordonnançabilité est:

$$\forall i, t_i \leq d_i - J_i \quad (3.2)$$

### 3. Arbitrage de priorités

La technique de résolution de collisions avec forçage, qui gère l'arbitrage d'accès, est basée, d'une part, sur des structures des en-têtes des trames qui reflètent le schéma de priorité des messages et, d'autre part, sur une durée du bit sur le bus qui est nécessairement supérieure au double du temps de propagation maximum. Compte tenu de cette propriété de la durée du bit, une trame sur le bus est alors le ET-logique des trames émises. En conséquence, toute station qui ne reconnaît pas son en-tête détecte une collision et se retire; la station qui ne détecte pas de collision (c'est celle dont l'en-tête compte la plus longue séquence de zéros) continue sa transmission (il y a forçage).

### 4. Remarques

Un inconvénient majeur de cette technique d'arbitrage d'accès est que chaque station ne peut émettre, au plus, qu'un bit par intervalle de temps (intervalle égal à quatre fois le délai maximum de propagation sur le canal). Donc, la longueur maximale du réseau est fortement dépendante du taux de transfert choisi. Dans le cas de ce protocole, les longueurs maximales définies sont assez réduites: 50m à 1 Mbit/s ou 100m à 500 Kbit/s.

En outre, pour garantir une bonne *efficacité temps-réel* du protocole, un compromis doit être établi entre la longueur maximale des messages et la longueur de l'en-tête nécessaire pour l'arbitrage des priorités:

- L'augmentation de la durée des messages augmente la durée des inversions de priorité dans le bus, ce qui peut empêcher l'ordonnançabilité de l'ensemble des flux de messages.

- La diminution de la durée des messages augmente proportionnellement le coût de l'arbitrage des priorités, ce qui peut réduire considérablement la largeur de bande utile pour la transmission de messages.

Dans le cas du protocole CAN, on a choisi de réduire la longueur des messages, ce qui amène à une efficacité discutable du protocole (en-tête de 66 bits pour des messages dont le champ utile des données ne dépasse pas les 64 bits). Par contre, la durée maximale des inversions de priorité dans le bus est assez réduite: 130 $\mu$ s à 1 Mbit/s.

### 3.3. Le protocole FIP

#### 1. Généralités

Le protocole FIP [NFa, 90] considère deux services de transmission: la transmission de variables et la transmission de messages. Les variables représentent les données échangées pour le contrôle d'un procédé industriel (on a des variables périodiques et apériodiques). Les messages permettent de gérer la configuration du système de contrôle et sont plutôt de nature apériodique. L'ordonnement de ces échanges de variables et de messages est défini et mis en oeuvre dans un contrôleur central appelé *Arbitre de Bus*.

L'arbitre de bus réalise les fonctions suivantes:

- 1- La scrutation périodique de variables périodiques (c'est-à-dire, la mise en oeuvre des échanges de variables périodiques);
- 2- La scrutation périodique de variables apériodiques et de messages (c'est-à-dire, l'envoi des requêtes périodiques à des stations, pour demander s'il y a des variables apériodiques ou des messages en attente); cette fonction peut être vue comme une fonction *serveur* au sens des techniques d'ordonnement de travaux apériodiques et peut être appelée *fonction de scrutation directe de variables apériodiques et de messages*.
- 3- La scrutation déclenchée de variables apériodiques et de messages (c'est-à-dire, la prise en compte des demandes venant des stations pour faire circuler des variables apériodiques et des messages); cette fonction peut être encore appelée *fonction de scrutation indirecte de variables apériodiques et de messages*.

Les principaux travaux sur l'arbitrage d'accès dans FIP définissent des techniques d'ordonnement qui privilégient le trafic périodique (fonctions 1 et 2): un schéma d'assignation de priorités et d'ordonnement associé aux variables et requêtes périodiques est défini. Il faut cependant remarquer que la notion de priorités pour l'échange de données a été introduite dans ces travaux, car elle n'est pas incluse dans la norme.

#### 2. Assignation de priorités

Deux principaux schémas d'assignation de priorités ont été proposés: priorités associées aux messages (priorités dynamiques) [Laine, 91] et priorités associées aux flux (priorités statiques) [RN, 93].

- Dans [Laine, 91] l'assignation de priorités est effectuée en considérant chaque message individuellement (c'est-à-dire, si le flux  $F$  génère  $m$  messages pendant un intervalle de temps donné, il y aura  $m$  messages différents à considérer par l'algorithme d'ordonnement). Un des avantages de cette assignation c'est la prise en compte des contraintes de précedence parmi les messages. Cependant, comme le nombre de messages pendant l'intervalle de temps peut être significatif (par rapport au nombre de flux de messages), ce schéma d'assignation de priorités peut devenir très complexe.

- Dans [RN, 93] les priorités sont assignées aux flux de messages (et non plus aux messages individuellement) et ensuite un ordonnancement basé sur des extensions aux algorithmes RM et ED [LL, 73] détermine la séquence des échanges.

### 3. Arbitrage de priorités

Nous indiquons maintenant le principe de la technique de la *table de scrutation* qui supporte l'ordonnancement dans le réseau FIP. La table de scrutation contient la liste de tous les identifiants des variables et des requêtes du trafic périodique et est organisée comme un échéancier d'échanges de ces identifiants. Cet échéancier est basé sur les notions de cycle élémentaire ou micro-cycle (plus petite fenêtre temporelle pendant laquelle chaque flux de trafic périodique peut transmettre au plus une entité d'information) et de macro-cycle (séquence de micro-cycles telle que les contraintes temporelles des flux du trafic périodique sont respectées sûrement). En considérant que l'on a  $p$  flux dans le trafic périodique avec des périodes  $P_1, P_2, \dots, P_p$ , les durées  $T_{mc}$  et  $T_{MC}$ , respectivement du micro-cycle et du macro-cycle, sont définies de la manière suivante:

$$T_{mc} = p \operatorname{gcd}\{P_i\}, \quad 1 \leq i \leq p \quad (3.3)$$

$$T_{MC} = \operatorname{ppcm}\{P_i\}, \quad 1 \leq i \leq p \quad (3.4)$$

Comme la durée d'un micro-cycle est généralement grande devant la durée des échanges des différents flux du trafic périodique, on a donc un espace temporel qui peut être utilisé pour le trafic lié à la fonction 3. En conséquence, le micro-cycle est décomposé en trois fenêtres: la fenêtre périodique (trafic périodique relatif aux fonctions 1 et 2), la fenêtre aperiodique (fonction 3 relative aux échanges de variables aperiodiques) et la fenêtre messages (fonction 3 relative aux échanges de messages). À la fin du macro-cycle, l'arbitre émet une séquence de bourrage du bus.

Un exemple de table de scrutation est donné sur la figure 3.5:

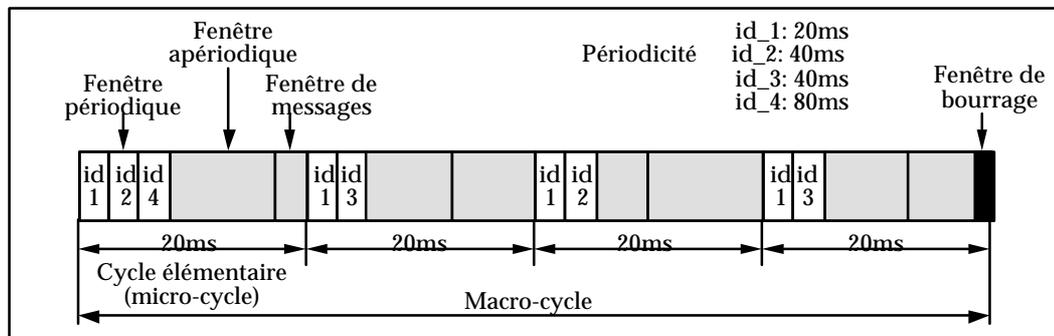


Figure 3.5: Table de scrutation dans FIP

Dans cet exemple, on considère un trafic périodique relatif à quatre variables et/ou requêtes de périodes  $20ms$ ,  $40ms$ ,  $40ms$  et  $80ms$  représentées, respectivement, par les identifiants  $id_1$ ,  $id_2$ ,  $id_3$  et  $id_4$ . La table de scrutation de l'arbitre a un macro-cycle de durée  $T_{MC} = 80ms$  découpé en quatre micro-cycles de durée  $T_{mc} = 20ms$ , chacun des micro-cycles comportant, pendant la fenêtre périodique, l'émission d'un ou plusieurs identifiants de trafic lié aux fonctions 1 et 2.

### 4. Mise en oeuvre de l'arbitrage d'accès dans l'arbitre

Les mécanismes de l'arbitrage d'accès sont représentés sur la figure 3.6.

L'ordonnancement durant un micro-cycle, en considérant uniquement le transfert de variables (on ne considère pas la messagerie) consiste en:

- 1- La mise en oeuvre de la fenêtre périodique qui consiste, successivement, en la consultation de la séquence élémentaire des identifiants de variables périodiques et de requêtes périodiques, la mise en circulation de ces identifiants à travers, respectivement, les trames *id\_dat* et *id\_rq*, et enfin, la prise en compte des réponses à ces trames.

Les réponses aux trames *id\_dat* peuvent être de deux types:

- a) soit ce sont des trames *rp\_dat* qui ne véhiculent que les valeurs des variables périodiques;
- b) soit ce sont des trames *rp\_dat\_rq* qui, outre les valeurs des variables périodiques, véhiculent également des requêtes (ces requêtes sont stockées dans les files de "demandes aperiodes" qui peuvent avoir le caractère urgent ou normal); c'est ce mécanisme qui initie ce que nous avons appelé la *scrutation indirecte de variables aperiodes* et qui sera réalisé dans la fenêtre aperiode.

Les réponses aux trames *id\_rq* sont des trames *rp\_rq* qui véhiculent des identifiants de variables aperiodes (ces identifiants sont stockés dans le buffer de reprise et vont être mis en circulation à travers des trames *id\_dat* dans la fenêtre périodique); cet aspect représente ce que nous avons appelé la *scrutation directe de variables aperiodes*.

- 2- La mise en oeuvre de la fenêtre aperiode, c'est-à-dire:

- a) la prise en compte des requêtes des files de "demandes aperiodes" et la mise en circulation de trames *id\_rq* véhiculant les identifiants de ces requêtes; les trames réponses (*rp\_rq*) véhiculent des identifiants de variables aperiodes qui sont stockés dans la file "aperiodes en cours".
- b) la prise en compte des identifiants de la file "aperiode en cours" et la mise en circulation de trames *id\_dat* contenant ces identifiants.

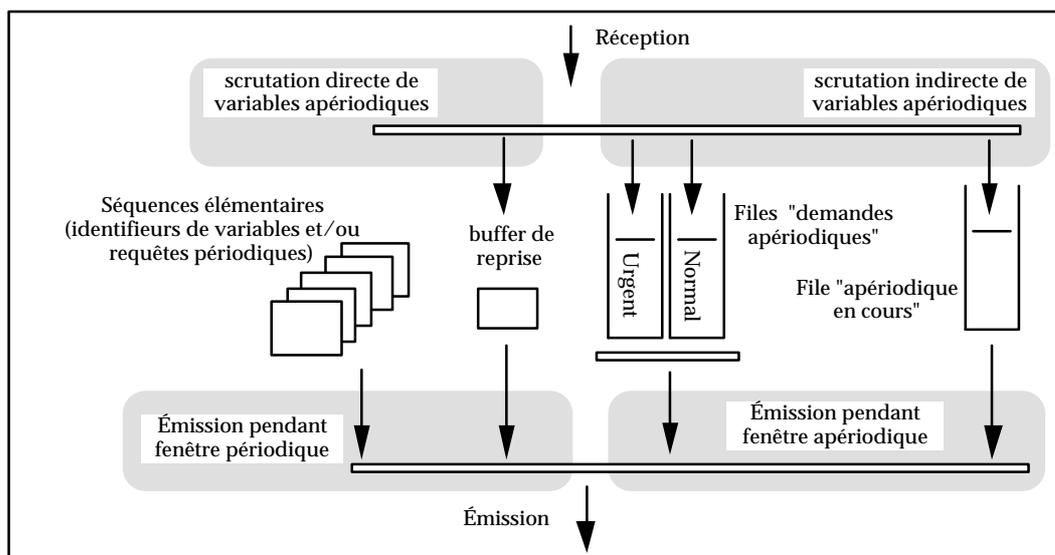


Figure 3.6: Les mécanismes de l'arbitrage d'accès dans FIP

Remarque 1: Le déroulement de la fenêtre aperiode d'un micro-cycle comporte les phases suivantes:

- tout d'abord, on traite toute la "file aperiodique en cours", c'est-à-dire, on fait circuler les identifiants de variables aperiodiques que l'on n'avait pas pu faire circuler lors de la fenetre aperiodique du micro-cycle précédent;
- ensuite, lorsque la file "aperiodique en cours" devient vide, on a l'exécution d'une demande stockée dans la file "demandes aperiodiques" urgent, s'il y en a, ou normal dans le cas contraire; rappelons qu'à la suite de ces exécutions, on aura de nouveaux identifiants qui arriveront dans la file "aperiodique en cours".

**Remarque 2:** L'ordonnancement du trafic périodique et du trafic aperiodique suivant la technique de scrutation directe est garantie au préalable. Par contre, l'ordonnancement des variables aperiodiques urgentes, suivant la technique de la scrutation indirecte, doit être évalué. En effet, le transfert de ce trafic dans la fenetre aperiodique est faiblement lié (du point de vue temporel) avec les besoins des transferts. Nous étudions ce problème dans le chapitre IV.

## 5. Scénarios d'échange de variables

Nous schématisons les principaux exemples de scénarios d'échange de variables:

- 1: La scrutation périodique de variables périodiques, qui est la fonction de base (figure 3.7);

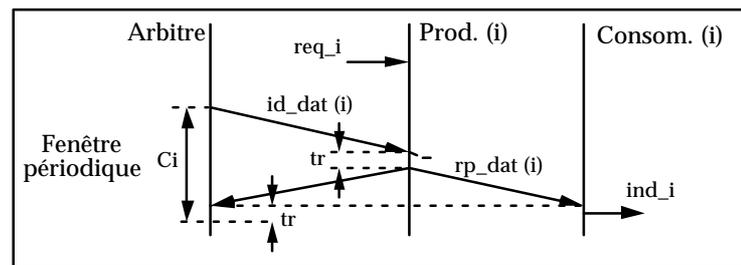


Figure 3.7: Scrutation périodique de variables périodiques

L'arbitre envoie périodiquement un PDU  $id\_dat$  adressé au producteur de la variable du flux périodique  $Mp_i$  (variable identifiée par  $i$  et que nous appelons variable  $i$ ); comme réponse (après la durée  $t_r$  fixée par le temps de retournement de la station), le producteur diffuse le PDU  $rp\_dat$  (qui contient la valeur de la variable  $i$ ), qui sera consommé par tous les consommateurs de la variable  $i$ .

La durée maximale d'utilisation du réseau  $C_i$  comprend, en particulier, la durée  $2 \cdot t_r$ , c'est-à-dire, outre le temps de retournement du producteur, on a également le temps de retournement de l'arbitre afin que ce dernier puisse démarrer une nouvelle scrutation.

- 2: La scrutation directe de variables aperiodiques (figure 3.8);

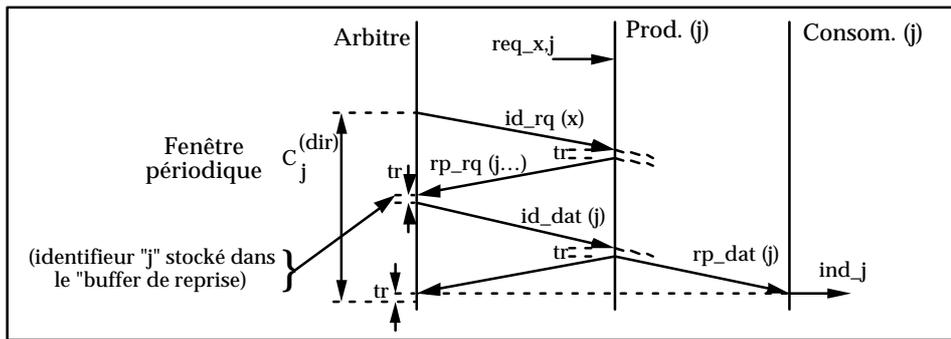


Figure 3.8: Scrutation directe de variables apériodiques

L'arbitre envoie périodiquement un PDU  $id\_rq$  adressé à un identifieur de requête  $x$  de la station à scruter; comme réponse, la station lui renvoie (dans un PDU  $rp\_rq$ ) la liste des flux apériodiques ( $j...$ ) qui ont des variables en attente; cette liste est stockée dans le "buffer de reprise", et immédiatement après, l'arbitre déclenche le transfert de toutes les variables en attente (pendant la fenêtre périodique en cours), en envoyant des PDUs  $id\_dat$  adressés à chacune de ces variables ( $j...$ ).

La durée maximale d'utilisation du réseau pour le transfert d'une variable apériodique  $j$  est notée  $C_j^{(dir)}$ .

3: La scrutation indirecte de variables apériodiques (figure 3.9);

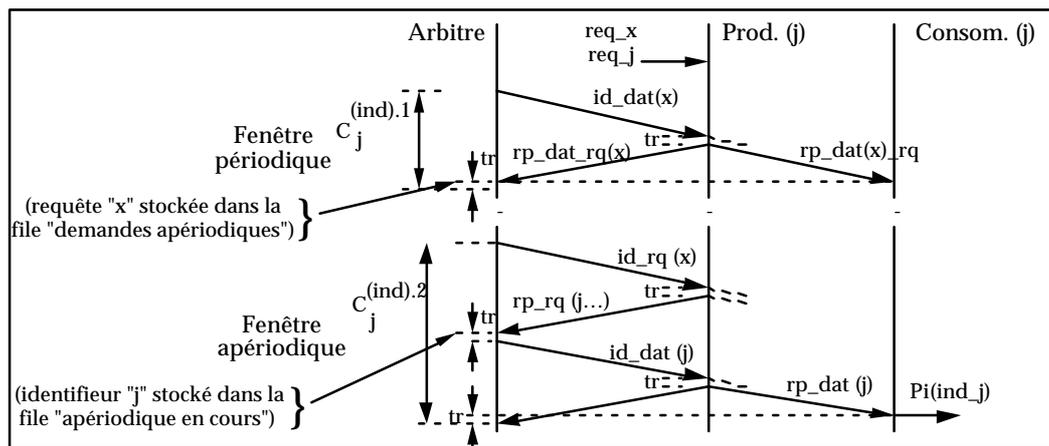


Figure 3.9: Scrutation indirecte de variables apériodiques

Pendant la scrutation périodique d'une variable  $x$ , la station scrutée peut aussi informer l'arbitre sur l'existence de trafic apériodique en attente (par l'intermédiaire d'un fanion  $rq$  attaché au PDU  $rp\_dat$  qui véhicule la valeur de la variable  $x$ ); cette information sera stockée par l'arbitre dans la "file de demandes apériodiques" sous la forme de l'identifieur  $x$ .

Ensuite, pendant une fenêtre apériodique, l'arbitre envoie la trame  $id\_rq(x)$  qui suscite une trame réponse  $rp\_rq(j...)$  spécifiant la liste des identifieurs ( $j...$ ) de variables apériodiques qui ont des données prêtes (ces identifieurs sont stockés dans la file "demandes apériodiques en cours"). Enfin, l'arbitre fait la scrutation de ces variables ( $j...$ ).

La durée maximale d'utilisation du réseau pour le transfert d'une variable apériodique  $j$  est notée  $C_j^{ind}$  avec  $C_j^{ind} = C_j^{(ind).1} + C_j^{(ind).2}$ .

### 3.4. Le réseau DQDB

#### 1. Généralités

Le réseau DQDB a été adopté par l'organisme de normalisation IEEE comme une norme (IEEE 802.6) pour les réseaux métropolitains [IEEE, 90]. L'architecture du réseau est composée d'une paire de bus unidirectionnels, d'une série de noeuds connectés aux deux bus et de deux générateurs de slots (intervalles de temps) à l'extrémité des bus (figure 3.10). Les générateurs de slots du bus A et du bus B sont appelés respectivement HOB(A) et HOB(B). Les HOB(A) et HOB(B) génèrent des slots de taille fixe.

Deux mécanismes pour contrôler l'accès au bus sont définis: "Queue Arbitrated" (QA) et "Pre-Arbitrated" (PA). Le premier est réservé au trafic asynchrone (données). Le deuxième est réservé au trafic isochrone (voix, vidéo). Les générateurs des slots des bus A et B génèrent donc deux types de slots (slots QA et slots PA).

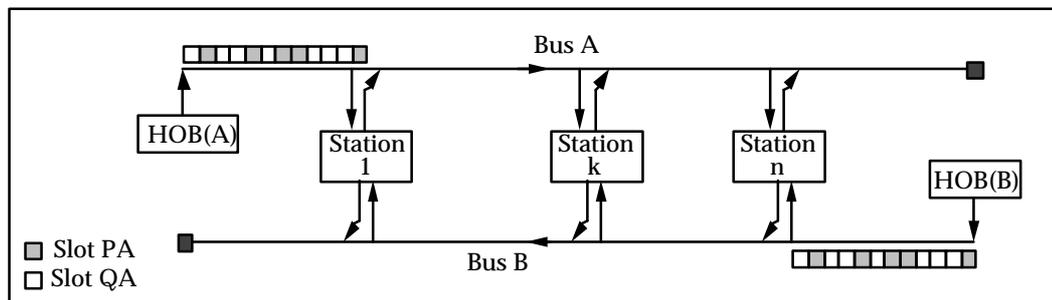


Figure 3.10: Le réseau DQDB

Le mécanisme QA (dont l'idée est de créer et de maintenir une file d'attente distribuée basée sur une stratégie FIFO) n'est pas compatible avec les exigences du trafic temps-réel [SSS, 92].

Le mécanisme PA, par contre, est prévu pour offrir des mécanismes temps-réel. Son principe est le suivant:

Chaque générateur de slots génère sur son bus des slots PA en adéquation, en termes de distribution temporelle, avec les besoins de chaque flux de messages isochrones à échanger sur ce bus (les slots PA portent dans l'en-tête un identifieur de flux de messages; cet identifieur est appelé *numéro de circuit virtuel*); Un slot PA (avec un identifieur  $x$ ) est utilisé en écriture par la station productrice du flux  $x$  et en lecture par les stations consommatrices du flux  $x$ .

La norme DQDB ne précise rien quant à l'ordonnancement du trafic PA. Peu de travaux ont été faits à ce sujet, si ce n'est les travaux de [SSMT, 93] et nos propres travaux [CVJ, 94] où nous avons proposé la mise en oeuvre dans chaque HOB d'un algorithme d'ordonnancement RM hors-ligne et un algorithme de changement de mode en-ligne (afin de prendre en compte l'évolution des besoins des échanges des flux isochrones). L'algorithme de changement de mode est une contribution originale que nous présentons dans le chapitre IV.

Nous résumons maintenant la proposition de la mise en oeuvre d'un algorithme d'ordonnancement RM, et plus précisément la méthode d'assignation de priorités.

#### 2. Assignation de priorités

Nous proposons un schéma d'assignation de priorités aux flux de messages temps-réel selon l'algorithme RM avec un double objectif:

- a) chaque flux  $Mp_i$  (avec période  $P_i$  et durée d'utilisation maximale du réseau  $C_i$ ) de trafic isochrone doit recevoir par période  $P_i$  le nombre de slots PA nécessaire pour satisfaire ses échéances (en supposant que l'on a  $n$  flux, la borne supérieure de l'utilisation maximale du réseau est  $n \cdot (2^{l^n} - 1)$  et a la valeur 69% lorsque  $n \rightarrow \infty$ ).
- b) si la borne supérieure de l'utilisation maximale du réseau n'est pas atteinte, on doit essayer de répartir la distribution de slots PA sur la totalité du cycle (afin d'éviter une concentration de ces slots PA au début du cycle, ce qui pénaliserait les accès au bus pour le trafic QA); pour ceci, on définit un flux fictif de période  $P_{QA}$  qui recevra donc des slots QA, tel que l'ensemble des  $n$  flux isochrones  $Mp_i$  et de ce flux fictif atteigne la borne de l'utilisation du réseau.

Nous résumons maintenant l'évaluation du test d'ordonnabilité de l'ensemble des  $n$  flux isochrones  $Mp_i$  et le calcul de la période  $P_{QA}$ :

- test d'ordonnabilité de l'ensemble des  $n$  flux isochrones  $Mp_i$  (l'utilisation  $U_i$  demandée par un flux  $Mp_i$  est égale à  $C_i/P_i$ ):

Appelons  $P_{PA_i}$  la période de génération des slots PA (durée unitaire) associés au flux  $Mp_i$ : l'utilisation  $U_{PA_i}$  du bus pour ce flux est égale à  $1/P_{PA_i}$ . On doit donc garantir que:  $U_{PA_i} \geq U_i$ , soit  $P_{PA_i} = \lfloor P_i/C_i \rfloor$ .

Pour l'ensemble de flux isochrones  $Mp_i$ , le test d'ordonnabilité doit être:

$$\sum_{i=1}^n \frac{1}{P_{PA_i}} < n \cdot (2^{l^n} - 1) \quad (3.5)$$

Notons bien que l'on doit avoir une relation de type "strictement inférieur" afin de pouvoir considérer l'objectif b).

Notons encore que si la durée des slots est l'unité de temps, l'ordonnement effectué à l'aide d'un algorithme préemptif est valable, tant que la durée des périodes est un multiple de la durée des slots (condition nécessaire pour établir une équivalence entre ordonnancement discret et ordonnancement continu [BRH, 90]); la durée d'un slot fixe la granularité temporelle du système.

- calcul de la période  $P_{QA}$ :

Si le test d'ordonnabilité pour l'ensemble des  $n$  flux isochrones  $Mp_i$  est respecté, on peut évaluer la période  $P_{QA}$  de la manière suivante:

$$\sum_{i=1}^n \frac{1}{P_{PA_i}} + \frac{1}{P_{QA}} \leq (n+1)(2^{l^{(n+1)}} - 1), \text{ d'où } P_{QA} = \left\lceil \frac{1}{(n+1) \cdot (2^{l^{(n+1)}} - 1) - \sum_{i=1}^n 1/P_i} \right\rceil \quad (3.6)$$

### 3. Arbitrage de priorités

Cet arbitrage est effectué pour chaque bus dans la station HOB (génération de slots PA avec des identifiants).

## 3.5. Le protocole IEEE 802.5

### 1. Généralités

Le protocole MAC IEEE 802.5 [IEEEb, 85] définit, sur une structure en anneau, un arbitrage d'accès global basé sur un mécanisme de passage de jeton. Toute station  $x$  qui capte le jeton,

émet une trame de données à destination d'une station  $y$  et c'est la station  $x$  qui retire la trame de données de l'anneau avant de régénérer le jeton.

La norme IEEE 802.5 définit huit niveaux de priorités pour des messages, ce qui permet donc de prévoir la mise en oeuvre d'un ordonnancement basé sur l'assignation de priorités aux flux, car le mécanisme de passage de jeton garantit que le message le plus prioritaire en attente dans tous les flux sera celui qui aura le droit d'accès accordé. Cependant, compte tenu du nombre réduit de niveaux de priorité, on doit, dans le cas le plus général, associer à chaque niveau de priorité plusieurs flux de messages (avec des priorités algorithmiques différentes), ce qui génère obligatoirement des inversions de priorité indésirables.

Quelques travaux ont été faits sur la mise en oeuvre effective d'un ordonnancement et, plus particulièrement, sur des modalités d'assignation de priorités [SML, 88] [Ple, 92].

## 2. Assignation de priorités [SML, 88]

Les auteurs proposent l'utilisation de l'algorithme RM pour effectuer l'assignation de priorités. Dans leurs travaux, ils présentent également le calcul du test d'ordonnabilité ainsi que le calcul de la durée maximale d'utilisation du réseau ( $C_i$ ) pour transférer un message périodique (flux  $Mp_i$ ).

Appelons  $T_{DATA}$  la durée d'une trame de données,  $T_{TOKEN}$  la durée du jeton et  $W_T$  la durée d'un tour de l'anneau. La durée d'utilisation du réseau  $C_i$  a les expressions suivantes:

- Si la durée d'un tour de l'anneau est inférieure ou égale à la durée de la trame de données [SML, 88],  $C_i$  est:

$$C_i = T_{DATA} + T_{TOKEN} + W_T \quad (3.7)$$

- Si la durée d'un tour de l'anneau est strictement supérieure à la durée de la trame de données [Ple, 92],  $C_i$  est:

$$C_i = T_{TOKEN} + 2 \cdot W_T + T_{FDS} \quad (3.8)$$

où  $T_{FDS}$  représente la durée de transmission des champs de contrôle de la trame de données.

Le test d'ordonnabilité est une application directe de l'algorithme RM [LL, 73].

## 3. Arbitrage de priorités

Cet arbitrage est décrit dans la procédure suivante. Considérons d'abord les variables:

$P(\text{jeton})$ : priorité actuelle du jeton;

$P(\text{dernier\_jeton})$ : priorité du dernier jeton capturé par la station;

$P(\text{réservation})$ : priorité de réservation du jeton;

$P(\text{message})$ : priorité la plus élevée parmi les priorités des messages en attente dans la station.

$S(\text{jeton})$ : statut du jeton (libre / occupé).

### Algorithme d'arbitrage de priorités du protocole IEEE 802.5

Dans chaque station, à l'arrivée du jeton, effectuer:

Tant qu'il y a des messages en attente:

Si  $P(\text{jeton}) \leq P(\text{message})$  et  $S(\text{jeton}) = \text{"libre"}:$

$P(\text{dernier\_jeton}) \leftarrow P(\text{jeton})$

$S(\text{jeton}) \leftarrow \text{"occupé"};$

```

Tant que  $P(\text{message}) \geq P(\text{réservation})$  :
     $P(\text{jeton}) \leftarrow P(\text{message})$  ;
    Envoi message ;
    Re-évaluer  $P(\text{message})$  ;
 $P(\text{jeton}) \leftarrow \max\{P(\text{dernier\_jeton}), P(\text{réservation})\}$  ;
 $S(\text{jeton}) \blacklozenge$  "libre" ;

Sinon  $P(\text{jeton}) > P(\text{message})$  ou  $S(\text{jeton}) = \text{"occupé"}$  :
    Si  $P(\text{message}) > P(\text{réservation})$  :
         $P(\text{réservation}) \blacklozenge P(\text{message})$  ;

```

#### 4. Critique

Les auteurs [SML, 88] appliquent le test d'ordonnançabilité de l'algorithme RM préemptif [LL, 73] à la transmission des messages, alors que, par définition, la transmission ne peut être préemptée au sens des tâches (un message est transmis ou n'est pas transmis) et donc on a forcément des inversions de priorité. Cependant, sachant que la condition de préemption au sein des tâches ne peut pas être respectée, ils proposent une fragmentation des messages de manière à limiter la durée des inversions de priorité (le choix du degré de fragmentation résulte d'un compromis entre l'efficacité du codage du transfert des fragments (taille des informations de contrôle par rapport à la taille des données) et la durée de l'inversion de priorité).

Il n'empêche que, malgré la fragmentation, l'application d'un test d'un algorithme préemptif au transfert de messages peut conduire à des réponses fausses: le test dit que l'ensemble de flux de message est ordonnançable et pratiquement (c'est-à-dire, en considérant la non-préemption) on ne peut pas l'ordonnancer. Donc, quel que soit le résultat du test, on ne peut jamais être sûr de l'ordonnançabilité de l'ensemble.

Considérons l'exemple suivant, où nous avons un ensemble de flux de messages temps-réel  $M = \{M_1, M_2, M_3, M_4\}$ , qui a les caractéristiques suivantes:  $P_i = \{100, 100, 100, 500\}$  et  $C_i = \{20, 20, 20, 60\}$ . Le test d'ordonnançabilité de l'algorithme RM dit que l'ensemble des flux est ordonnançable ( $U = 0.72 < n(2^{1/n} - 1) = 0.7568$ ).

Cependant, nous pouvons constater que, pour un déphasage particulier entre les flux de messages (le message de plus faible priorité  $M_4$  démarre son transfert et juste après des messages de plus haute priorité émettent des requêtes pour être transmis, mais devront atteindre la fin de la transmission du message  $M_4$ ), l'ensemble  $M$  n'est pas ordonnançable (l'échéance de  $M_3$  n'est pas respectée); ceci est visualisé sur la figure 3.11 (la partie a) représente le cas "sans fragmentation"; la partie b) représente le cas "avec fragmentation", la longueur maximale de fragmentation étant fixée à 50 et donc seul le message de  $M_4$  est fragmenté).

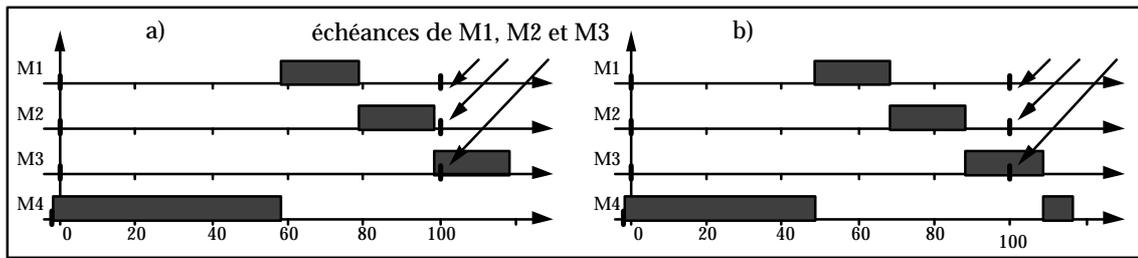


Figure 3.11: Exemple de non ordonnançabilité en utilisant l'algorithme RM

Un test d'ordonnançabilité acceptable est celui de l'algorithme RM non-préemptif que nous avons proposé au chapitre II et qui donne le résultat suivant dans le cas de l'exemple traité:

- pas de fragmentation:  $U = 0.72 + \frac{60}{100} = 1.32 > 0.7568$
- avec fragmentation:  $U = 0.72 + \frac{50}{100} = 1.22 > 0.7568$

## 4. Protocoles de la "classe 2"

### 4.1. Principales normes

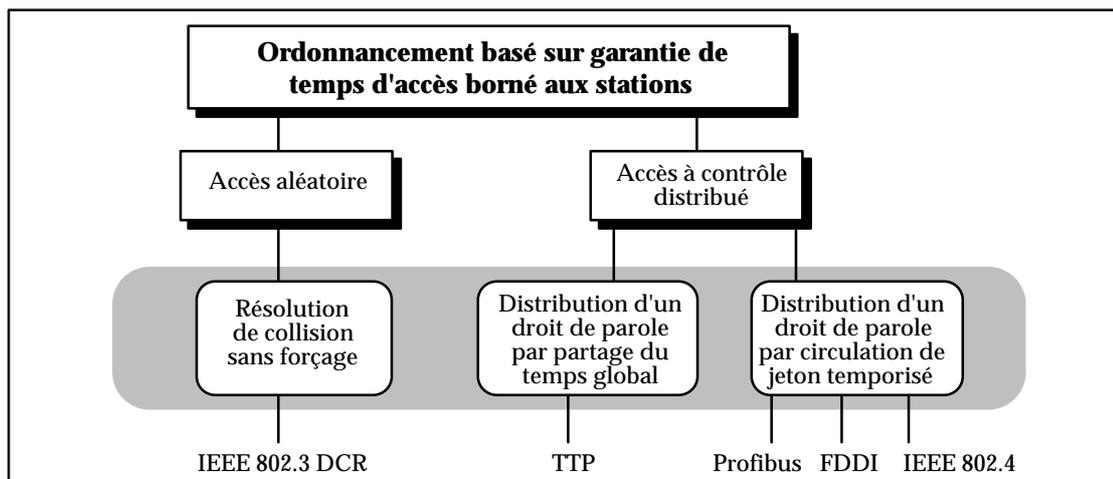


Figure 3.12: Les protocoles "classe 2" existants

Les principales normes de "classe 2" existantes sont représentées sur la figure 3.12.

Nous présentons maintenant les éléments essentiels de l'arbitrage d'accès et du contrôle de transmission dans le cas de chaque protocole.

### 4.2. Le protocole IEEE 802.3 DCR

#### 1. Principe

Ce protocole, qui s'intègre dans le cadre de la norme IEEE 802.3 [IEEEc, 85], y apporte la valeur ajoutée de la garantie d'un délai d'accès borné en modifiant l'algorithme de résolution des collisions.

- L'arbitrage d'accès est élémentaire tant qu'il n'y a pas de collisions; à l'instant où il y a une collision, le protocole utilise le principe de résolution d'une *époque temporelle* au

moyen d'un arbre binaire, qui associe des priorités soit aux stations [Rol, 89], soit aux flux de messages des stations [ARS, 90]. C'est ce mécanisme qui permet d'obtenir un délai d'accès borné. Nous considérons ici uniquement le mode général à entrées bloquées [Rol, 91].

- Le contrôle de transmission est élémentaire (on transmet un message par époque).

## 2. Comparaison entre les techniques d'association de priorités aux stations et aux flux de messages

Considérons que l'on a  $m$  stations et  $n$  flux de messages (généralement  $n \gg m$ ).

Si on associe des priorités aux stations [Rol, 89] [ARS, 90], la taille de l'arbre est minimale ( $2^{\lceil \log_2 m \rceil}$ ) mais l'ordonnancement ne tient pas compte des priorités des flux de messages, donc de leurs caractéristiques temporelles, ce qui n'est pas totalement satisfaisant dans un contexte temps-réel.

Si on associe des priorités aux flux de messages [ARS, 90], la taille de l'arbre est maximale ( $2^{\lceil \log_2 n \rceil}$ ), ce qui donne une longue époque temporelle et donc peut induire des inversions de priorité de longue durée (un flux de messages plus prioritaire que ceux qui sont impliqués dans la résolution de l'époque temporelle doit attendre la fin de cette époque pour essayer d'accéder à la ressource de transmission).

Compte tenu de ces constatations, on peut définir [ARS, 90] un schéma intermédiaire de priorités associées aux messages ( $q$  niveaux de priorités avec  $m < q < n$ ), ce qui peut donc donner une même priorité associée à des flux de messages de différentes stations). Dans ce cas, la résolution de l'époque peut se voir comme une opération à deux niveaux: tout d'abord on résout l'époque sur la base de la priorité des messages; si une collision subsiste, lorsqu'on a atteint les feuilles de l'arbre binaire, on résout la contention sur la base des adresses des stations.

De toute manière, comme la durée des inversions de priorité est proportionnelle à la taille de l'arbre, on doit donc, pour garantir une bonne efficacité temps-réel, trouver un compromis entre la taille de l'arbre et la prise en compte des caractéristiques temporelles des messages.

## 3. Arbitrage d'accès

Cet arbitrage est décrit dans la procédure suivante:

**Algorithme d'arbitrage d'accès à époque temporelle**

Dans chaque station, tant qu'il y a des messages en attente:

Si le canal de communication est libre:

L'accès est accordé immédiatement;

Si le canal de communication est occupé:

L'algorithme d'arbitrage d'accès est re-exécuté à l'instant où le canal de communication redevient libre;

Tant que le message est en cours d'émission:

Si une collision dans le canal de communication est détectée:

Le transfert du message est suspendu;

La station envoie une séquence de brouillage;

Une époque temporelle est déclarée ouverte;

Un algorithme de résolution de collision garantit que le message intervenant dans cette collision sera transmis pendant un intervalle de temps borné;

L'époque temporelle est déclarée fermée;

#### 4. Remarque

En considérant la modalité d'assignation de priorités aux flux de messages, on doit se poser la question du classement de ce protocole dans la classe 2 et non dans la classe 1. La raison en est que, du fait du concept d'époque temporelle en mode à entrées bloquées, on n'a pas un ordonnancement global des messages (seuls les premiers messages de chaque station sont ordonnancés suivant le schéma de priorité; on peut avoir, par exemple, la situation où le deuxième message d'une station est plus prioritaire que les premiers messages des autres stations et qui ne pourra donc être ordonnancé que dans la fenêtre temporelle suivante). Une fenêtre temporelle ne garantit qu'un délai d'accès borné aux messages de la station.

### 4.3. Le protocole TTP

#### 1. Généralités

Le protocole TTP [K\_all, 89] qui a été défini dans le cadre du système d'exploitation MARS, implante une méthode d'accès à contrôle distribué sur un bus. Ce protocole est une implémentation très particulière du protocole TDMA ("Time Division Multiple Access") classique, où chaque station possède une vue de l'horloge globale du système.

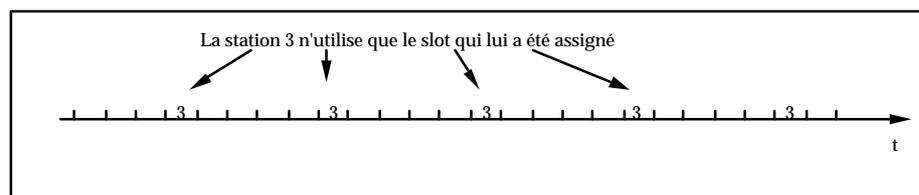


Figure 3.13: Protocole TTP

#### 2. Arbitrage d'accès

Dans ce protocole, le "temps global" est divisé (hors-ligne) en une séquence de slots, chaque slot étant pré-assigné à une seule station; cette séquence définit la table de scrutation du protocole (figure 3.13). Dans le cas le plus simple, le nombre de slots assignés dans la table de scrutation est égal au nombre de stations, et donc le temps global est équitablement partagé; cependant, quand les besoins des stations ne sont pas identiques, un nombre différent de slots sera assigné à chaque station dans la table de scrutation.

Toutes les stations, connaissant la table de scrutation et ayant une référence temporelle commune, savent quand elles peuvent utiliser le réseau (évidemment, par le principe même, un slot est perdu s'il n'y a pas de messages à transmettre).

Une particularité intéressante de ce protocole est la suivante: dans chaque station, la différence entre l'instant d'arrivée réel d'un message et l'instant d'arrivée prévu pour ce même message, donne une indication sur la différence existante entre les horloges des deux stations; cette indication est suffisante pour l'exécution d'un algorithme distribué de synchronisation d'horloges (fondamental pour le fonctionnement correct du protocole, puisque l'accès est à contrôle distribué).

#### 3. Contrôle de transmission

Une station ne peut transmettre qu'un message par slot.

## 4.4. Les protocoles basés sur le jeton temporisé: IEEE 802.4, FDDI et Profibus

L'arbitrage d'accès est tout simple: on a un passage séquentiel du jeton entre les stations. Le contrôle de transmission est le mécanisme fondamental de ces protocoles car c'est lui qui peut garantir le délai d'accès borné. Ce contrôle de transmission est basé sur le protocole du "jeton temporisé" [Gro, 82], que nous allons détailler maintenant. Nous proposerons au chapitre IV une modification de ce protocole dans l'optique de meilleures performances temps-réel.

Ensuite, nous indiquons, en prenant ce protocole comme référence, les particularités des trois normes IEEE 802.4, FDDI et Profibus qui reprennent la philosophie du protocole "jeton temporisé".

### 4.4.1. Le protocole "jeton temporisé"

#### 1. Principe

Ce protocole est basé sur la considération de deux types de trafic: le trafic synchrone (appelé "synchronous class") qui désigne un trafic avec des contraintes temporelles et le trafic asynchrone (appelé "asynchronous class") qui désigne un trafic non contraint temporellement.

Le trafic synchrone est considéré comme le trafic de base. L'idée fondamentale de ce protocole est:

- d'une part, d'allouer à chaque station une durée pré-définie qui représente le temps maximum pendant lequel elle peut transmettre du trafic synchrone chaque fois qu'elle reçoit le jeton;
- d'autre part, de contrôler la durée de rotation du jeton ou temps de cycle (intervalle de temps entre deux arrivées consécutives du jeton à une station), afin que les contraintes temporelles du trafic synchrone puissent être satisfaites.

Ici, nous considérerons que les trafics temps-réel (périodique ou apériodique) et non-temps-réel sont véhiculés, respectivement, par le trafic synchrone et par le trafic asynchrone.

Un paramètre fondamental, négocié dans la phase d'initialisation d'un réseau, est le paramètre  $T_{TRT}$  ("Target Token Rotation Time") qui définit le temps de rotation cible du jeton (temps vers lequel le temps de rotation doit tendre en moyenne). C'est par rapport à ce paramètre que l'on définit le temps maximum pendant lequel une station peut transmettre du trafic temps-réel chaque fois qu'elle reçoit le jeton (ce temps maximum est encore appelé la bande passante allouée à la station). Plus précisément, en supposant que l'on a  $m$  stations dans un réseau, on spécifie la relation suivante:

$$\sum_{k=1}^m H_k \leq T_{TRT} - t \quad (3.10)$$

où  $H_k$  est la bande passante allouée à la station  $k$  et  $t$  est la partie de  $T_{TRT}$  non allouable aux stations (on a  $t = q + D$ ,  $q$  étant la latence totale de l'anneau logique et  $D$  représentant le surplus qui dépend du protocole (durée du PDU "jeton", "asynchronous overrun", etc)).

Cette relation, qui peut encore s'écrire

$$\sum_{k=1}^m H_k \leq T_{TRT}(1 - a), \quad a = t/T_{TRT} \quad (3.11)$$

est la relation traduisant la **contrainte du protocole**, c'est-à-dire, la somme des allocations de bande passante pour le trafic temps-réel doit être inférieure à  $T_{TRT}$ .

En ce qui concerne le trafic non-temps-réel, ce protocole prévoit qu'une station peut en transmettre si le jeton lui revient au bout d'un temps inférieur à  $T_{TRT}$ .

La mise en oeuvre de ce protocole dans une station  $k$  est basé sur l'utilisation de trois compteurs:  $TRT_k$  ("Token Rotation Time") qui évalue le temps de cycle (ce compteur est initialisé à la valeur  $T_{TRT}$  et re-initialisé à cette valeur  $T_{TRT}$  soit lorsque le jeton arrive à la station, soit lorsque le temps  $T_{TRT}$  est écoulé);  $THT_k$  ("Token Holding Time") qui définit le temps pendant lequel la station  $k$  peut transmettre du trafic non-temps-réel;  $LC_k$  ("Late Counter") qui mémorise le nombre de fois où le temps  $T_{TRT}$  s'est écoulé depuis la précédente arrivée du jeton à la station  $k$ .

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

```

Algorithme "jeton temporisé" [Gro, 82]

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

Tant que le réseau est en fonctionnement:
  si  $TRT_k = 0$ , faire:                                  /* attente du jeton*/
     $TRT_k \leftarrow T_{TRT}$ ;
     $LC_k \leftarrow LC_k + 1$ 

  À l'arrivée du jeton, faire:                          /* transfert de données */
    cas  $LC_k = 0$ :                                       /* cas jeton en avance */
       $THT_k \leftarrow TRT_k$ ;
       $TRT_k \leftarrow T_{TRT}$ 
      Démarrer compteur  $TRT_k$  (décrémental)
      transfert de messages temps-réel pendant jusqu'à  $H_k$  unités de
temps
      Démarrer compteur  $THT_k$  (décrémental)
      tant que  $THT_k > 0$  et (mgs non-temps-réel en attente)
        transfert de messages non-temps-réel
        passage du jeton à la station ( $k+1$ )
    cas  $LC_k = 1$ : /* cas jeton en retard */
       $LC_k \leftarrow 0$ 
      transfert de messages temps-réel pendant jusqu'à  $H_k$  unités de
temps
      passage du jeton à la station ( $k+1$ )

    cas  $LC_k > 1$  : /* cas erreur */
      procédure "récupération d'erreur"

```

Les points suivants doivent être remarqués:

- on a les notions d'arrivée du jeton en avance ( $LC_k = 0$ ) ou en retard ( $LC_k = 1$ , le cas  $LC_k > 1$  étant considéré comme une erreur (par exemple: perte du jeton)).
- quelle que soit la condition du jeton lorsqu'il arrive dans une station (en avance ou en retard), on transfère toujours le trafic temps-réel (s'il y en a) pendant un temps égal au maximum à  $H_k$ ).
- le trafic non-temps-réel n'est pas transmis si le jeton est en retard.
- le retard du jeton est engendré par une durée trop longue d'envoi de trafic non-temps-réel par des stations.

Le phénomène de la potentialité d'un retard dans le passage du jeton est un aspect très important que nous présentons maintenant, en considérant le cas le plus dur, c'est-à-dire, qui induit la plus longue durée pour le temps de cycle du jeton (cas pire). Prenons le cas limite pour la contrainte de protocole  $\left(\sum_{k=1}^m H_k = T_{TRT}(1-a)\right)$  et considérons le scénario où d'une part toutes les stations ont du trafic temps-réel à transmettre en permanence sur toute leur bande passante et, d'autre part, la première station utilise tout le temps dont elle dispose pour transmettre du trafic non-temps-réel (elle peut en transmettre pendant un temps  $T_{TRT}$ ).

Nous donnons sur la figure 3.14 le diagramme temporel représentant l'occupation du temps pour des échanges sur le réseau (on considère 3 stations).

On a:  $T_{CYCLE_{max}} = \sum_{k=1}^m H_k + T_{TRT} + t$  avec  $\sum_{k=1}^m H_k = T_{TRT} - t$ , soit  $T_{CYCLE_{max}} = 2 \cdot T_{TRT}$  et donc:

$$T_{CYCLE} \leq 2 \cdot T_{TRT} \quad (3.12)$$

Cette relation  $T_{CYCLE} \leq 2 \cdot T_{TRT}$  est une **propriété temporelle** du protocole "jeton temporisé".

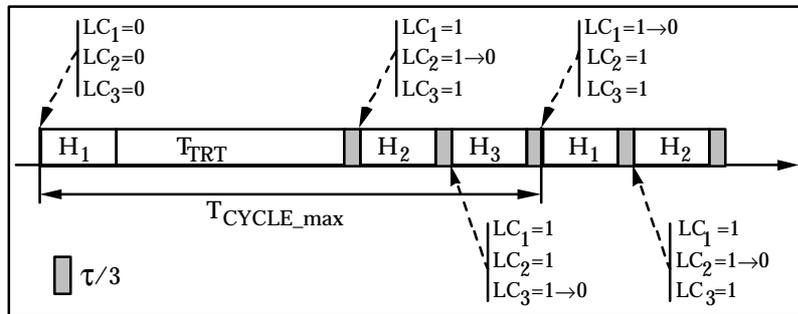


Figure 3.14 Diagramme temporel

**Remarque 1:** Dès que l'on vient d'entrer dans ce "cas le plus dur", on peut dire que tant que toutes les stations utilisent toute la bande passante qui leur est attribuée pour le trafic temps-réel ( $H_k$  pour la station  $k$ ), on ne pourra plus faire circuler du trafic non-temps-réel. Ce n'est que si les stations n'utilisent plus la totalité de leur bande passante  $H_k$  que le jeton pourra de nouveau être en avance et donc du trafic non-temps-réel pourra être retransmis de nouveau.

**Remarque 2:** Si toutes les stations respectent la contrainte de protocole, on peut voir que l'on n'aura jamais la valeur  $LC = 2$ , c'est-à-dire, une situation d'erreur.

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

Dans le paragraphe précédent, nous avons exposé les caractéristiques du protocole du "jeton temporisé" en termes d'allocation de bande passante aux stations pour le trafic temps-réel, mais cette présentation a été transparente aux caractéristiques des flux de trafic à transmettre. Il faut donc maintenant préciser quelles sont les relations que l'on doit avoir entre les caractéristiques des flux temps-réel et les caractéristiques du protocole afin que les échéances des flux temps-réel soient satisfaites.

Nous considérons, dans la suite de cette présentation, que l'on a un flux périodique par station (comme dans les travaux de [ACZD, 92]) et nous considérons donc un ensemble de  $m$  flux de messages périodiques  $\{M_i\}$ ,  $1 \leq i \leq m$ , avec  $M_i = (P_i, C_i)$ , où  $P_i$  est la période du flux  $M_i$  et  $C_i$  est la durée maximale d'utilisation du réseau nécessaire à ce flux  $M_i$  par période. La bande passante allouée à la station qui génère le flux  $M_i$  est appelée  $H_i$ . Les échéances sont sur requête.

Afin que les contraintes des échéances de tous les flux  $M_i$  soient satisfaites, il faut que:

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

La condition 1 implique que la période  $P_i$  soit supérieure au temps maximum de cycle, donc:

$$\forall i, 1 \leq i \leq m, P_i \geq 2 \cdot T_{TRT} \quad (3.13)$$

Le calcul de la condition 2 nécessite d'évaluer le nombre minimum  $v_i$  de passages du jeton dans la station pendant une période  $P_i$ . Pour faire cette évaluation, il faut, d'une part, prendre en compte l'asynchronisme entre les requêtes des flux de messages et les passages du jeton et donc se placer dans le cas pire de déphasage et, d'autre part, considérer la circulation du jeton dans le cas le plus dur de retard (figure 3.15).

Le nombre minimum  $v_i$  de passages du jeton pendant la période  $P_i$  se déduit du cas représenté sur la figure:  $v_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor$ , d'où on obtient:  $X_i = v_i \cdot H_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor \cdot H_i$ . La condition 2 implique que le temps  $X_i$  soit supérieur ou égal à  $C_i$ , soit:

$$\left\lfloor \frac{P_i}{T_{TRT}} - 1 \right\rfloor \cdot H_i \geq C_i \quad (3.14)$$

Cette condition 2 exprime ce que l'on appelle la **contrainte de l'échéance**.

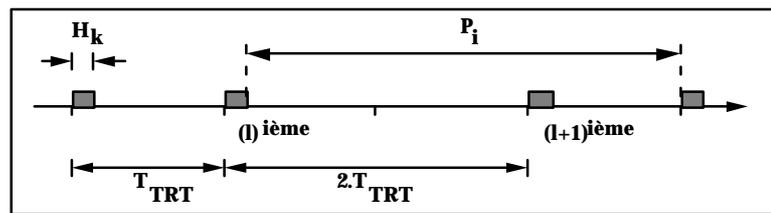


Figure 3.15: Évaluation de  $v_i$

### 3. La problématique de l'allocation de la bande passante

Un algorithme d'allocation de bande passante, doit évaluer, en fonction de  $T_{TRT}$ , l'ensemble  $\vec{H} = \{H_1, H_2, \dots, H_i, \dots, H_n\}$  des allocations de bande passante aux différentes stations, sachant que seules les caractéristiques temporelles  $(C_i, P_i)$  des flux de messages sont connues au préalable.

L'évaluation de cet ensemble  $\vec{H}$  doit être effectuée en deux étapes:

- D'abord, une valeur doit être choisie pour  $T_{TRT}$ , de manière à ce que le jeton passe dans chaque station au moins une fois pendant la période de chaque flux  $M_i$ : donc il faut que  $v_i = \lfloor (P_i/T_{TRT}) - 1 \rfloor \geq 1$ . Ceci implique le choix d'une valeur pour  $T_{TRT}$  telle que:  $\forall i, 1 \leq i \leq m, P_i \geq 2 \cdot T_{TRT}$ .
- Ensuite, l'utilisateur doit vérifier si l'ensemble des allocations de bande passante spécifié respecte les contraintes de protocole et de l'échéance (respectivement:  $(\sum_{i=1}^n H_i \leq T_{TRT}(1-a))$  et  $(\lfloor (P_i/T_{TRT}) - 1 \rfloor H_i \geq C_i)$ ), qui fournissent la garantie de respect des contraintes temporelles des flux de messages.

Nous présentons maintenant les principaux algorithmes d'allocation de bande passante.

#### 4. Les principaux algorithmes d'allocation de bande passante

Les algorithmes proposés dans [ACZD, 92], [ACZ, 93] et [ACZD, 94] ont été définis dans le cadre d'un seul flux de messages temps-réel par station, et sont basés sur les connaissances, d'une part, des caractéristiques des flux de messages et, d'autre part, du temps  $T_{TRT}$  qui est défini à la configuration. Les principaux algorithmes ainsi définis sont:

- L'algorithme d'allocation *durée complète* qui alloue à un flux  $M_i$  une bande passante  $H_i$  égale à la durée  $C_i$  d'un message:

$$H_i = C_i \quad (3.15)$$

Avec cet algorithme, à l'arrivée d'un jeton, le message en attente, s'il y en a, est complètement transmis (il n'y a pas de fragmentation de messages).

- L'algorithme d'allocation *proportionnel* qui alloue à un flux  $M_i$  une bande passante proportionnelle à ses besoins d'utilisation du réseau. Sachant que  $(T_{TRT} - t)$  représente le temps total maximum de cycle du jeton disponible pour le trafic temps-réel, on prend:

$$H_i = \frac{C_i}{P_i} \cdot (T_{TRT} - t) \quad (3.16)$$

Avec cet algorithme les messages du flux  $M_i$  sont toujours fragmentés ( $P_i$  étant toujours supérieure à  $(T_{TRT} - t)$  puisque  $P_i \geq 2 \cdot T_{TRT}$ ) et donc, à chaque arrivée du jeton, le flux  $M_i$  dispose d'un temps suffisant pour transmettre un seul fragment de chaque message temps-réel en attente. La durée de ce fragment est:

$$\frac{C_i}{P_i / (T_{TRT} - t)} \quad (3.17)$$

L'inconvénient de cet algorithme c'est qu'il essaye de diviser la durée du transfert d'un message pour un nombre non entier d'arrivées du jeton ( $P_i / (T_{TRT} - t)$  n'est pas forcément un entier!).

- L'algorithme d'allocation *proportionnel adapté* est un algorithme d'allocation équivalent à l'algorithme proportionnel, mais qui prend en compte que le jeton n'arrive à la station pour servir le flux  $M_i$  qu'un nombre  $v_i$  entier de fois pendant l'intervalle  $[t, t + P_i]$ :

$$H_i = \frac{C_i}{\lfloor (P_i / T_{TRT}) - 1 \rfloor} \quad (3.18)$$

#### 5. Mesure de performance des algorithmes d'allocation de bande passante

Les algorithmes d'allocation de bande passante sont généralement comparés au moyen de l'évaluation de la borne inférieure de l'utilisation maximale [ACZD, 92]. C'est une adaptation de l'évaluation proposée pour l'analyse de l'algorithme RM [LL, 73].

La borne minimale  $U_x$  de l'utilisation maximale associée à un algorithme  $x$  d'allocation de bande passante est ainsi définie:

- si  $\forall M, U(M) \leq U_x$ , alors  $M$  est ordonnançable par l'algorithme  $x$ .

L'intérêt d'une borne minimale  $U_x$  de l'utilisation maximale c'est que tant que on a une configuration de flux de messages temps-réel  $M$ , telle que  $U(M) \leq U_x$ , on sait que l'algorithme  $x$  garantit le respect des contraintes temporelles. Par contre, dans le cas contraire, il faut vérifier (donc calculer) si ces contraintes sont satisfaites.

Les algorithmes d'allocation présentés ont les bornes  $U_x$  suivantes [ACZD, 92] [ACZD, 94]:

- algorithmes *durée complète* et *proportionnel*:  $U_x = 0\%$ ; dans ce cas, il faut donc toujours vérifier si les contraintes temporelles de la configuration  $M$  considéré sont satisfaites;
- algorithme *proportionnel adapté*:  $U_x \rightarrow 33.3\%$  ( $a$  étant voisin de zéro).

#### 4.4.2 Les particularités des protocoles FDDI, IEEE 802.4 et Profibus

Bien que le protocole du "jeton temporisé" puisse être vu comme le dénominateur commun des mécanismes des protocoles FDDI [ANSI, 90], IEEE 802.4 [IEEEa, 85] et Profibus [DIN, 91], il existe cependant des différences entre ces protocoles dont nous indiquons maintenant (sans rentrer dans le détail) les points essentiels. Nous nous focalisons sur les points suivants: la classification des messages; le concept de  $T_{TRT}$ ; le trafic temps-réel; le trafic non temps-réel; le concept "temps maximal de cycle".

##### 1. Classification des messages

Deux classes de trafic sont toujours considérées: le trafic synchrone ou haute-priorité (c'est-à-dire, celui qui peut supporter du trafic temps-réel) et le trafic asynchrone ou basse-priorité (c'est-à-dire, du trafic non-temps-réel). Dans le trafic asynchrone, plusieurs niveaux de priorité sont considérés (8 pour FDDI; 3 pour IEEE 802.4; 4 pour Profibus).

##### 2. Le concept de $T_{TRT}$

FDDI et Profibus utilisent ce concept de manière similaire au protocole du "jeton temporisé", c'est-à-dire, une seule valeur pour  $T_{TRT}$  quelle que soit la classe de trafic considérée.

Par contre, IEEE 802.4 utilise le concept de  $T_{TRT}$  uniquement pour le trafic asynchrone, avec des valeurs différentes pour chaque niveau de priorité. Le temps de cycle du jeton est maintenant évalué indépendamment pour chaque niveau de priorité, et en fonction de la dernière fois que la station a pu transférer du trafic de cette priorité.

##### 3. Le trafic temps-réel

A chaque arrivée du jeton (dans FDDI et IEEE 802.4) on dispose d'un temps pré-défini ( $H_k$  pour FDDI; *High Priority Token Holding Time* -  $T_{HP\_THT}$  pour IEEE 802.4) pendant lequel on peut transférer du trafic temps-réel. En ce qui concerne Profibus, il n'y a aucun temporisateur associé spécifiquement au trafic temps-réel: à chaque arrivée du jeton, on peut toujours transférer au moins 1 message temps-réel (on ne peut transférer qu'un message temps-réel si la cible  $T_{TRT}$  est dépassée; si cette cible n'est pas dépassée, on peut transférer plusieurs messages (jusqu'à l'écoulement de la cible)).

##### 4. Le trafic non-temps-réel

Dans FDDI et Profibus, la condition pour faire du transfert de trafic non-temps-réel est identique à celle présentée pour le protocole "jeton temporisé", c'est-à-dire, la station peut transférer ce trafic tant que la cible  $T_{TRT}$  n'est pas dépassée. En ce qui concerne les différents niveaux de priorités dans FDDI, chaque niveau dispose d'un temps maximum d'émission pré-défini (allocation de bande), tandis que, dans Profibus, on envoie tout le trafic d'un niveau avant de considérer le niveau inférieur.

Dans IEEE 802.4, la poursuite du transfert de trafic d'un niveau de priorité  $x$  (auquel est associé un  $T_{TRT\_x}$ ) dépend du temps de rotation effectif du jeton ( $T_{REJ}$ ), c'est-à-dire, du temps écoulé depuis le précédent passage du jeton pour ce niveau de priorité: si, et tant que,

$T_{REJ} \leq T_{TRT\_x}$  on peut effectuer des transferts; dans le cas contraire, on ne peut pas. Une analyse intéressante de ce fonctionnement est faite dans [PT, 89].

## 5. Concept de temps maximal de cycle

On a les résultats suivants:

- FDDI (identique au "jeton temporisé":  $2 \cdot T_{TRT}$  [SJ, 87];
- IEEE 802.4:  $\sum_{k=1}^m H_k + \max_x \{T_{TRT\_x}\}$  [MCV, 92];
- En ce qui concerne Profibus, à notre connaissance, il n'y a pas de résultats bien établis sur le temps maximal de cycle. Nous présentons notre propre réflexion sur ce protocole au chapitre IV.

## 5. Un protocole mixte: classe 1&2

### 5.1. Généralités

La proposition de norme ISA SP-50 / IEC-65C [IEC, 94] définit un protocole centralisé, où le contrôleur central, appelé l'*ordonnanceur actif de la ligne* ("LAS: Link Active Scheduler"), gère les accès des autres stations (appelées "DLE: Data Link Entity") au bus. De nombreuses possibilités d'adressage sont définies (CEP, SAP, groupes de CEPs et de SAPs), ce qui permet d'offrir l'accès au bus à différents types de sources d'information, que ce soit un flux d'une station, un groupe de flux ou la station globalement.

Trois PDUs sont définis, afin de permettre au LAS de distribuer trois types de droit de parole (on dit encore de déléguer un jeton (notion de jeton délégué) pour permettre à une station DLE d'utiliser le bus):

- type 1: Le PDU *CD* ("Compel Data") qui permet au LAS de forcer l'envoi immédiat d'un PDU qui est dans un DLE (l'envoi de ce PDU redonne aussitôt la main au LAS).
- type 2: Le PDU *ET* ("Execute Transaction"), qui permet au LAS de demander à un DLE d'exécuter immédiatement une transaction (séquence de deux PDUs, où l'envoi du deuxième PDU redonne aussitôt la main au LAS); notons que le premier PDU de la transaction peut être un PDU *CD* (ce qui permet un degré d'indirection dans le forçage d'envoi).
- type 3: Le PDU *ES* ("Execute Sequence"), qui permet au LAS de donner à un DLE un temps de droit de parole (un paramètre *ES duration* est dans ce PDU). L'envoi, par le DLE, du dernier PDU (c'est un PDU qui a un fanion particulier ou c'est un PDU *EE* ("End of Execution")) redonne la main au LAS.

Le LAS doit évidemment posséder les informations concernant les besoins dans les DLEs, afin d'ordonner les échanges. Certains besoins (échanges périodiques) sont définis à la configuration et la séquence temporelle de ces échanges peut donc être connue du LAS, ce qui permet au LAS de les ordonner. D'autres besoins (échanges apériodiques) peuvent être connus de manière dynamique par le LAS (au moyen de PDUs qui lui redonnent la main).

Il est important de noter que les profils d'utilisation de cet ensemble de fonctionnalités ne sont pas définis dans la norme ISA SP-50 / IEC-65C. L'implantation de politiques d'ordonnement temps-réel est encore ouverte. Cependant, cette norme ISA SP-50 / IEC-65C est riche en possibilités d'ordonnements permis, comme nous l'indiquons maintenant.

Dans le chapitre V, nous proposons précisément une architecture de communication temps-réel basée sur notre interprétation de cette norme.

## 5.2. Ordonnancements permis

Les droits de parole de type 1 et de type 2 définissent ce que nous appelons la technique du jeton délégué à réponse immédiate:

- le type 1 permet, par exemple, d'ordonnancer le transfert de flux de messages périodiques depuis leurs producteurs vers leurs consommateurs. Ce service est identique au service de scrutation périodique de variables de FIP.
- le type 2 permet (dans l'hypothèse où le premier PDU de la séquence est le PDU CD) d'ordonnancer des demandes périodiques de consommateurs à des producteurs.

La technique du jeton délégué à réponse immédiate peut donc être utilisée pour mettre en oeuvre un ordonnancement basé sur une assignation de priorités aux flux de messages (ordonnancement de classe 1, tel comme nous l'avons défini).

Le droit de parole de type 3 définit ce que nous appelons la technique du jeton délégué temporisé. Cette technique, appliquée successivement et de manière ordonnée à l'ensemble des stations DLE, permet d'implanter un protocole du type "jeton temporisé", c'est-à-dire, on peut donc mettre en oeuvre un ordonnancement basé sur une garantie de droit d'accès borné aux stations DLE (ordonnancement de classe 2, tel comme nous l'avons défini).

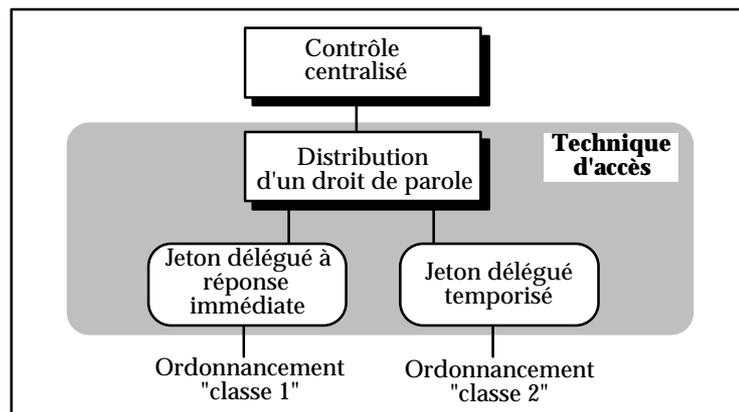


Figure 3.16: Le protocole IEC 65C

En conséquence, les mécanismes protocolaires de la norme ISA SP-50 / IEC-65C peuvent être utilisés pour mettre en oeuvre les deux classes d'ordonnancement 1 et 2 (voir figure 3.16).

## 6. Évaluation comparative des classes 1 et 2

### 6.1. Critères d'évaluation

Quatre critères sont essentiels pour caractériser des protocoles temps-réel: la robustesse (c'est-à-dire la capacité à absorber des variations de paramètres temporels sans affecter l'ordonnancement global), la borne minimale de l'utilisation maximale (c'est-à-dire la garantie de l'ordonnançabilité de la configuration des flux), la capacité de prise en compte du trafic non-temps-réel (conjointement avec du trafic temps-réel) et le temps de l'arbitrage d'accès.

## 6.1.1. Comparaison

### Robustesse

Un protocole à ordonnancement hors-ligne quelle que soit la classe (1 ou 2) est, par définition, toujours plus robuste qu'un protocole à ordonnancement en-ligne: en effet l'ordonnancement étant effectué au préalable, la séquence de transfert des messages n'est pas perturbée par la violation de spécifications temporelles dans des flux. Par exemple, si un flux augmente sa cadence de production, ce flux ne peut pas transmettre tous les messages qu'il produit (seule une séquence correspondante à l'ordonnancement pré-défini est transmise) mais les autres flux ne s'en aperçoivent pas (seul le flux fautif est pénalisé).

Dans le cas d'un protocole à ordonnancement en-ligne, l'ensemble des flux de messages temps-réel reste ordonnançable tant que le test d'ordonnançabilité est respecté. Dans le cas d'une violation par un flux des spécifications temporelles, il peut y avoir des échéances (d'autres flux) non respectées. Les différences entre la classe 1 et la classe 2 sont les suivantes:

- dans un protocole de classe 1, si l'algorithme d'assignation de priorités est dynamique la propagation d'une faute temporelle est imprévisible, c'est-à-dire, que les flux qui peuvent voir leurs échéances non respectées ne sont pas connus au préalable; par contre, si l'algorithme est statique, la propagation des fautes temporelles est prévisible car elle affecte d'abord les flux de messages qui ont les priorités les plus basses dans le système.
- dans un protocole de classe 2, la violation des spécifications temporelles n'affecte que les flux de la station concernée, car la durée pendant laquelle chaque station peut utiliser le réseau est limitée, soit à la bande passante allouée par cycle du jeton (protocole jeton temporisé), soit à un message par époque (protocole à époque temporelle).

Nota: nous ne considérons que les aspects concernant la robustesse aux fautes temporelles dans les caractéristiques des flux de messages. Pour effectuer une analyse approfondie sur la robustesse du protocole, il faudrait considérer aussi les fautes temporelles du protocole lui-même.

### Borne minimale de l'utilisation maximale

Les protocoles de classe 1 autorisent des valeurs importantes (qui peuvent avoisiner les 69% avec l'algorithme RM et les 100% avec l'algorithme ED), ce que ne permet pas l'exemple type du protocole de classe 2 qu'est le protocole du jeton temporisé (33% avec l'algorithme d'allocation de bande proportionnel adapté). Ceci résulte du fait que dans la classe 1, on a une synchronisation forte entre l'ordonnancement et les flux, alors que dans le protocole du jeton temporisé, les passages de jeton et les productions des flux sont asynchrones.

### Capacité de prise en compte du trafic non-temps-réel

La classe 1 offre des potentialités intéressantes grâce au mécanisme de serveur. En ce qui concerne la classe 2, les conclusions sont différentes selon que l'on a le protocole à jeton temporisé ou le protocole avec collision et résolution d'époque temporelle: la potentialité du trafic non-temps-réel d'une station dépend, avec le premier, du trafic non-temps-réel écoulé par les autres stations et, avec le deuxième, du trafic temps-réel de la station.

### Temps d'arbitrage d'accès

Ce temps est évidemment réduit, par définition, dans le cas des protocoles à ordonnancement hors-ligne (que l'on soit en classe 1 ou 2). En ce qui concerne les protocoles à ordonnancement en-ligne:

- dans la classe 1, une opération d'arbitrage est effectuée chaque fois qu'un message doit être transféré et donc le coût temporel des opérations d'arbitrage est élevé (en particulier dans le cas des réseaux haut-débit, ce coût peut devenir très important et donc l'ordonnancement en-ligne n'est pas intéressant de ce point de vue là).
- dans la classe 2, le protocole à jeton temporisé présente des propriétés très intéressants (avec un seul passage de jeton dans une station, plusieurs messages peuvent être transmis); par contre, le protocole avec collision et résolution d'époque temporelle peut demander des temps d'arbitrage très longs (si la taille de l'arbre est grande).

## **7. Conclusion**

Le travail présenté dans ce chapitre est, à notre avis, fondamental pour bien intégrer la problématique des protocoles MAC temps-réel, à la fois, en termes d'ordonnancement et de transfert de messages. Nous voulons mettre en exergue les points suivants:

- nous avons proposé une classification des protocoles MAC en deux grandes classes: la classe 1, où l'ordonnancement est basé sur une assignation de priorités aux flux de messages, et la classe 2, où l'ordonnancement est basé sur une garantie de temps d'accès borné aux stations;
- nous avons développé une analyse des protocoles MAC temps-réel existants dans les deux classes en termes de deux processus: l'arbitrage d'accès et le contrôle de la durée de transmission; les protocoles CAN, FIP, DQDB et IEEE 802.5 se situent dans la classe 1; les protocoles IEEE 802.3DCR, TTP, "jeton temporisé" et ses dérivés FDDI, IEEE 802.4 et Profibus se situent dans la classe 2;
- nous avons fait une présentation complète du protocole "jeton temporisé" en développant, en particulier, les schémas d'allocation de bande passante qu'il offre, ainsi que les taux d'utilisation permis;
- nous avons montré l'intérêt de la norme ISA SP-50 / IEC-65C qui permet, à la fois, la mise en oeuvre de mécanismes de la classe 1 et de la classe 2 et donc offre de grandes possibilités d'ordonnancement;
- nous avons effectué une brève réflexion comparative entre les protocoles de classe 1 et les protocoles de classe 2, par rapport aux attributs robustesse, borne minimale de l'utilisation maximale, potentialité de trafic non-temps-réel et temps d'arbitrage d'accès.

## **8. 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 Dist. 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;
- [ANSI, 90] ANSI Standard X3T9.5/88-139, Rev. 4: FDDI MAC; 1990

- [ARS, 90] K. Arvind, K. Ramamritham, J. Stankovic; Window MAC protocols for Real-Time Communication Services; COINS Technical Report 90-127, University of Massachusetts at Amherst, Jan. 1991;
- [B\_all, 92] S. Baruah, G. Koren, D. Mao, B. Mishra, A. Raghunathan, L. Rosier, D. Shasha, F. Wang; On the Comptitiveness of On-Line Real-Time Task Scheduling; Journal of Real-Time Systems, 4, pages 125-144, 1992
- [B\_all, 92] S. Baruah, L. Rosier, R. Howell; Algorithms and Complexity Concerning the Preemptive Scheduling of Periodic Real-Time Tasks on one Processor; Journal of Real-Time Systems, 42, pages 301-324, 1990
- [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
- [DIN, 91] DIN 19245 Part 1 & 2: Process Field Bus; April 1991
- [Gro, 82] R. Grow; A Timed Token Protocol for Local Area Networks; IEEE Electro'82, Token Access Protocols, 17/3, 1982
- [IEC, 94] Digital Data Communications for Measurements and Control - Field Bus for use in Industrial Control Systems; Parts 3 (105 v1.1) & 4 (106 v1.1), IEC SC65C/WG6
- [IEEEa, 85] Token Bus Access Method and Physical Layer Specifications, IEEE Std 802.4 - 1985
- [IEEEb, 85] Token Ring Access Method and Physical Layer Specification, IEEE Std 802.5 - 1985
- [IEEEc, 85] Carrier Sense Multiple Access with Collition Detection (CSMA/CD), IEEE Std 802.3 - 1985
- [IEEE, 90] Distributed Queue Dual Bus (DQDB) Subnetwork of Metropolitan Area Network, IEEE Std 802.6 - 1990
- [K\_all, 89] H. Kopetz, A. Damm, C. Koza, M. Mulazzani, W. Schwabl, C. Senft, R. Zainlinger; Distributed Fault-Tolerant Real-Time Systems: The MARS Approach; IEEE MICRO, Feb. 1989
- [Laine, 91] T. Laine; Modélisation d'Applications Réparties pour la Configuration Automatique d'un Bus de Terrain; Thèse de Doctorat, INP de Lorraine, CRIN, 29 Mai 1991
- [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;
- [MCV, 92] P. Montuschi, L. Ciminiera and A. Valenzano; Time Characteristics of IEEE 802.4 Token Bus Protocol; IEE Proceedings-E, vol. n°1, January 1992.
- [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
- [Ple, 92] P. Pleineveaux; An Improved Hard-Real-Time Scheduling for the IEEE 802.5; The Journal of Real-Time Systems, 4(2), pages 99-112, 1992;
- [PT, 89] J. Pang, F. Tobagi; Throughput Analysis of a Timer Controlled Token Passing Protocol Under Heavy Load; IEEE Tr. on Comm., 37(7), pages 694-702, July 1989.

- 
- [RN, 93] P. Raja, G. Noubir; Static and Dynamic Pooling Mechanisms for Fieldbus Networks; in *Operating Systems Review*, ACM Press, 27(3), pages 34-45, July 1993.
- [Rol, 91] P. Rolin; *Réseaux Locaux: normes et protocoles*; 2<sup>e</sup> édition, Hermès, Paris, 1989
- [SAE, 92] Controller Area Network (CAN), an In-Vehicule Serial Communication Protocol-SAE J1583; (report of the Vehicule Network for Multiplexing and Data Communication Standards Comitee); SAE Handbook, vol. II, 1992;
- [SJ, 87] K. Sevcik and M. Johnson; Cycle Time Properties of the FDDI Token Ring Protocol; *IEEE Trans. on Software Eng.*, vol SE-13, n°3, 1994, pp 376-385.
- [SML, 88] J. Strosnider, T. Marchok, J. Lehoczky; Adadvanced Real-Time Scheduling using the IEEE 802.5 Token Ring; *IEEE RTSS'88*, pages 42-52, 1988;
- [SSMT, 93] D. Saha, M. Saksena, S. Mukherjee, S. Tripathi; On Guaranteed Delivery of Time-Critical Messages in DQDB. Technical Report CS-TR-3149, University of Maryland, October 1993.
- [SSS, 92] L. Sha, S. Sathaye, J. Strosnider; Scheduling Real-Time Communication on Dual-Link Networks; *IEEE RTSS'92*, pages 188-197, 1992;
- [THW, 94] K. Tindell, H. Hansson, A. Wellings; Analysing Real-Time Communications: Controller Area Network (CAN); *IEEE RTSS'94*, S. Juan, pages 259-263, Dec. 1994

