

***Introduction
aux
bus et réseaux
temps réel***

Bertrand Dupouy

Bus et réseaux temps réel

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- Ordonnancement de tâches communicant par messages

Plan

- Le temps dans les bus et réseaux
 - 1- **Simplification du modèle ISO**
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnement

- Exemple de bus temps réel : le bus CAN
- Ordonnement de tâches communicant par messages

Le réseau Qu'est ce que c'est ?

- Le réseau est un composant essentiel : il est le support de la communication entre tâches ,
 - c'est donc une ressource **critique** ...
 - ... qui devra respecter des **contraintes temporelles**
 - Pour nous c'est un **système de communication**, c'est à dire l'infrastructure physique **plus** :
 - des couches logicielles,
 - une interface pour accéder à ces couches (*CNI, communication network interface*)
 - Rappelons le modèle en 7 couches de l'ISO :

Bus et réseaux temps réel

Rappel : Les couches 1 à 7

5 à 7 : les services applicatifs	Application	Services offerts à l'utilisateur
	Présentation	Représentation abstraite des données
	Session	Conventions et contrôles de dialogue
1 à 4 : Le système de communication chargé des transferts d'information	Transport	Transfert fiabilisé des données et contrôle de flux
	Réseau	Adressage et routage
	Liaison	Accès au médium, arbitrage des conflits, détection des erreurs
	MAC	Partage du médium
	Physique	Transmission physique sur le réseau

Bus et réseaux temps réel

Un réseau Pour le temps réel

- Les exigences du temps réel (-> **ajout** de contraintes) :
 - le délai de communication point à point doit être borné,
 - les transferts peuvent être :
 - périodiques (*monitoring, feedback*), généralement à période courte et sous contrainte de gigue faible,
 - apériodiques(alarmes), généralement rapides (ms),
- Les **particularités du temps réel** (-> **retrait** de contraintes) :
 - les messages sont courts (fragmentation et réassemblage inutiles),
 - pb. trame ISO et message court :

Encapsulation

Message

- la diffusion se fait dans un seul domaine (pas besoin de routage),
- les applications sont bien définies (couche présentation inutile).

Du modèle ISO Vers un modèle plus simple

- Les 7 couches auxquelles sont associés services et protocoles engendrent beaucoup d'overhead et d'indéterminisme,
- Un modèle simplifié semble mieux adapté :
 - Les applications accèdent directement à la couche 2,
 - Dans l'industrie, on parle de bus de terrain (*fieldbuses*),

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel**
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- Ordonnancement de tâches communicant par messages

Bus et réseaux temps réel

Le modèle Des bus de terrain

- Modèle simplifié, les applications accèdent directement à la couche 2 :

Niveau applicatif	« Couche Application »
Le système de communication	Couche 2 : Liaison (Adressage, gestion des erreurs et MAC)
	Couche 1 : Physique (Topologie, support physique, débit, nombre de nœuds,...)

- Estimation du temps passé dans chaque couche :

Couches hautes (piles réseau + bibliothèque du type sockets) :

délai de traversée

Couche MAC

délai d'attente : calculable

Support physique :

délai de propagation : calculable

Bus et réseaux temps réel

Temps de transfert D'un message (1)

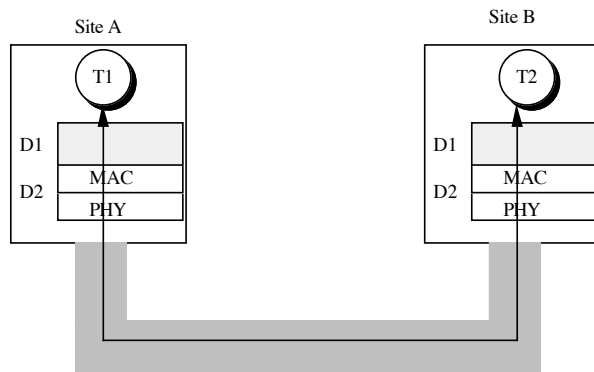
- Temps de transfert (délai de bout en bout) d'un message entre deux tâches T_i et T_j , sachant que T_i est émettrice sur le site A et T_j destinataire sur le site B :

$$\Delta_j^i = D_{1A} + D_{2A} + D_T + D_P + D_{2B} + D_{1B}$$

- Avec :

- D_{2A} : délai dû à la traversée des couches MAC et PHY sur le site A
- D_{1A} : délai dû à la traversée des couches supérieures sur le site A
- D_{2B} : délai dû à la traversée des couches MAC et PHY sur le site B
- D_{1B} : délai dû à la traversée des couches supérieures sur le site B
- D_T : délai de transmission : taille des paquets/bande passante (bits /bits par s)
- D_P : délai de propagation (longueur/vitesse) sur le medium

Temps de transfert D'un message (2)



- Temps de transfert d'un message émis par la tâche T1 vers la tâche T2 :

$$\Delta_{2}^{1} = D_{1A} + D_{2A} + D_{T} + D_{P} + D_{2B} + D_{1B}$$

Utilisation de bus ou réseaux temps réel

- Exemples :

- automobiles : CAN, LIN (<http://www.lin-subbus.org>), TT_CAN (<http://www.can-cia.org/can/ttcan>), TTP,
 - avionique, aéronautique : (Arinc 653, Aeronautical Radio Inc., <http://www.arinc.com>)
 - production industrielle, robots : Profibus (<http://www.profibus.com>), World FIP
 - trains : World FIP (www.worldfip.org)
 - systèmes multimédia (Firewire).
 - FDDI (www.fddi.org)
 - Couches MAC à protocole déterministe (CSMA)
- Problèmes à résoudre :
 - ordonnancement des tâches (global ou local), placement : statique, dynamique (avec ou sans migration)
 - choix du réseau, ou des réseaux,
 - ordonnancement des messages.

Les bus De terrain (fieldbuses)

- Citons quelques-uns des bus de terrain les plus utilisés :
 - Controller Area Network , CAN Bosch (automobile), Allemagne, Version 2 en 1991 (www.can-cia.de)
 - Réseau synchrone, cadence donnée par le bus
 - Local Interconnect Network , LIN Audi, Volvo, ... (automobile), 1999 (www.lin-subbus.org)
 - Factory Instrumentation Protocol, WorldFIP : France 1980 (www.worldFIP.org) processus industriels,
 - Site central, maître de l'arbitrage, producteur/consommateur
 - Time Triggered Protocol, TTP/C : 1990, projet MARS, Univ. Vienne, Autriche (www.ttech.com) orienté sûreté
 - FlexRay

Les couches 1 et 2

- Comme on l'a vu, dans le cas des bus de terrain, les applications accèdent directement à la couche 2 (couche liaison, *Data Link Layer*),
- Nous allons faire quelques rappels sur le fonctionnement de ces couches 1 (couche physique, *physical layer, PHY*) et 2 (DLL)
- Remarque : les modèles applicatifs restent les mêmes :
 - client-serveur, publish-subscribe, etc...

La couche physique

- Les propriétés importantes de la couche 1:
 - Topologie,
 - Support physique : cuivre, fibre optique, sans fil (radio, IR),
 - délai de propagation : Longueur/Vitesse
 - Taux de transmission,
 - Longueur maximum, nombre de nœuds, alimentation,
 - Résistance aux attaques, aux perturbations physiques,

La couche physique

Topologie	Avantages	Inconvénients
Arbre	Point à point. Plusieurs communications en parallèle	Routage. Chemins de longueurs différentes
Anneau	Point à point. Câblage simple	Temps de parcours.
Bus	Simple. Communication directe : pas de routage.	Contrôle d'accès au medium.

Support	Avantages	Inconvénients
Cuivre	Faible coût	Sensible aux EMI (ElectroMagnetic Interferences)
Fibre optique	Peu sensible aux EMI. Large bande. Peu d'atténuation.	Coût élevé.
Sans fil	Mobilité, flexibilité.	Très sensible aux EMI.

La couche liaison

- Les fonctions de la couche 2 (*Data Link Layer*) :
 - **adressage** :
 - explicite (on envoie un identifiant),
 - implicite (par le contenu),
 - **Logical Link Control, LLC**, gère les transferts d'information :
 - avec ou sans acquittement,
 - en mode connecté ou non,
 - **Medium Access Control, MAC** : détermine l'ordre et les délais des accès au support de communication, le fonctionnement de ce niveau est **critique pour les applications temps réel**
 - avec contrôle d'accès : une règle détermine l'accès au medium,
 - sans contrôle d'accès : on émet dès qu'on est prêt à émettre. Il faut alors gérer les collisions.

La couche liaison

- Exemple de réseaux à couche MAC déterministe :
 - PROFIBUS (Process Field Bus www.profibus.com), basé sur un protocole du type *Timed Token Protocol*,
 - FIP: un site maître distribue les accès au bus,
 - CAN, développé pour l'industrie automobile, : le site de plus haute priorité accède au bus, CSMA/BA

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. **avec contrôle d'accès (exemple : jeton temporisé)**
 - b. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- Ordonnancement de tâches communicant par messages

La couche liaison : MAC avec contrôle d'accès

- Différents types de contrôle d'accès :
 - Maître-esclave (WorldFIP, LINm Ethernet Powerlink). **Centralisé**, les accès sont gérés par un site maître, il y a une trame de contrôle par trame de données,
 - Jeton (FDDI, PROFIBUS). Le site qui détient le jeton peut accéder au réseau, problèmes pour le TR :
 - Difficile de savoir quand se font les accès au réseau,
 - La circulation détermine un ordre,
 - Borner le temps de détention du jeton,
 - TDMA (*time division multiple access*), les accès se font dans des créneaux alloués de façon cyclique, il faut synchroniser les horloges (exemple : FlexRay, PROFINET, TTP/C, TT-CAN)

La couche liaison : avec contrôle d'accès : jeton temporisé (timed token protocol)

- Sa place par rapport au protocole jeton classique :
 - Similitude: un site ne peut émettre que s'il détient le jeton,
 - Différences:
 - deux types de trafic : **synchrone** (à chaque passage du jeton on garantit l'accès à une partie de la bande passante), **asynchrone** (allocation à la demande si le jeton arrive plus tôt que prévu),
 - pas de site moniteur, chaque site participe à la surveillance de l'anneau.
- Principe de fonctionnement:
 - Lorsqu'il reçoit le jeton, un site va disposer d'une fenêtre d'accès au médium de durée D. Il va l'utiliser pendant C pour émettre un message (C : cf. politiques d'ordonnancement) ,
 - Si $C < D$, alors il pourra émettre des messages dits asynchrones,

Contrôle d'accès : jeton temporisé (timed token protocol)

- Rappel sur les paramètres physiques :
 - Déb : débit offert ou encore capacité
 - L : longueur,
 - v : vitesse de propagation du signal,
 - TTrame : taille moyenne d'une trame,
 - D1: (délai de propagation/temps d'émission) = $(L/v) / (\text{Déb} \cdot \text{TTrame})$, nbre de trames dans le canal avant arrivée du premier bit,
 - D2 : (délai de propagation*bande passante) = $(L/v) \cdot \text{Déb}$, nbre d'octets dans le canal avant arrivée du premier bit,

Exemple, avec TTrame = 1000 octets et Déb = 100 Mbps :

	L	D1	D2
FDDI	10 km	4	4000
FDDI	100 km	40	40000

Contrôle d'accès : jeton temporisé (timed token protocol)

- Types de données transmises :
 - synchrones (périodiques, à contraintes de débit),
 - asynchrones (sporadiques, à débit variable),
- Sur chaque site, quatre variables d'état permettent de gérer l'anneau:
 - TTRT (Target Token Rotation Time), durée maximale d'attente du jeton sur un site,
 - LC (Late Counter), indique si le jeton est arrivé en avance. Si oui, on pourra émettre des données asynchrones,
 - TRT (Token Rotation Timer), pour contrôler les dates de réception du jeton,
 - THT (Token Holding Timer), durée d'émission des messages asynchrones, si elle est permise.

Contrôle d'accès : jeton temporisé , les variables d'état (1)

- TTRT (Target Token Rotation Time), durée maximale d'attente du jeton par un site. Cette valeur est définie à l'initialisation du réseau :
 - chaque site propose une valeur, la plus **petite** est retenue : c'est celle proposée par le site qui a les plus fortes contraintes temporelles.
 - La valeur choisie pour TTRT résulte d'un compromis entre :
 - Une contrainte temporelle : le respect des échéances (D), assuré si $D \geq 2 * TTRT$,
 - Une contrainte de débit, ou bande passante : le temps d'accès au bus doit être suffisant pour émettre tous les bits du message (!!) ,
- LC(i) (Late Counter), autorise (LC = 0) ou interdit (LC = 1) l'émission de données asynchrones par le site i :
 - LC est (re)mis à 0 si le jeton arrive à temps (avant expiration de TTRT),
 - LC est mis à 1 si le jeton arrive en retard (après expiration de TTRT),
 - Si le jeton arrive en retard pour la seconde fois, le site réinitialise l'anneau.

Contrôle d'accès : jeton temporisé , les variables d'état (2)

- TRT(i) (Token Rotation Timer), délai de temporisation :
 - Gestion : (ré)initialisé à TTRT dans l'un des cas suivants :
 - au début,
 - le jeton arrive et LC(i) = 0 (il n'était pas en retard la dernière fois),
 - TRT a été décrémenté jusqu'à atteindre la valeur 0,
 - Après initialisation, il est régulièrement décrémenté.
 - Rôle : il fixe la durée d'émission pour les données du type asynchrone :
 - Si le jeton est revenu **avant** expiration de TRT(i) (il est en avance) et LC(i) =0 (il n'était pas en retard la dernière fois), alors :
 - Envoi de données synch. pendant TS(i) = S_i*TTRT **puis**
 - Envoi de données asynch. pendant THT(i),
 - Sinon :
 - Envoi des seules données **synch.** pendant TS(i) = S_i*TTRT

Contrôle d'accès : jeton temporisé , les variables d'état (1)

- Chaque site i dispose d'un créneau TS(i) (bande passante « synchrone »), qui est une fraction du TTRT, pendant lequel il peut émettre des données de la classe dite synchrone, TS(i) est tel qu'il respecte les deux contraintes :
 - Contrainte protocole :
$$\sum_{(i=1)}^N TS(i) \leq TTRT - k, (k \text{ est l'overhead réseau})$$
$$TS(i) = (C_i/T_i) * TTRT, \text{ si } T_i = D_i (\text{échéance} = \text{période})$$
 - Contrainte temporelle : chaque TS(i) doit être tel que les messages aient été complètement émis avant occurrence de l'échéance :

Le temps mis à disposition de i sur le réseau (TX_i) entre t et t+T_i doit être supérieur à C_i :

$$TX_i \geq C_i$$

TX_i dépend **du nombre de sites** et du **nombre de réceptions du jeton** par i pendant une période T_i.

Résultat : cette contrainte temporelle est respectée si T_i ≥ 2*TTRT,

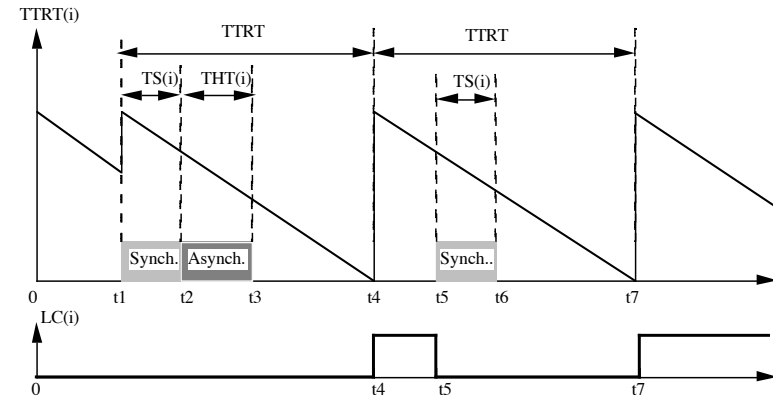
**Contrôle d'accès :
jeton temporisé , les variables d'état (2)**

- Si le jeton arrive en avance, le site peut émettre des données dites « asynchrones » pendant THT(i) (Token Holding Timer), le délai de garde.
- Utilisation de THT(i), on remarque qu'il n'y a pas de garantie de bande passante comme pour les données synchrones qui disposent d'un créneau de TS(i) unités de temps d'accès au support de communication.
 - Si le jeton arrive avant expiration de TRT(i), c'est à dire si il est en **avance** et si LC(i) = 0, c'est à dire s'il n'était pas en retard au tour précédent, alors :

THT(i) = TRT(i) et émission de données asynchrones pendant THT(i)

Jeton temporisé , exemple

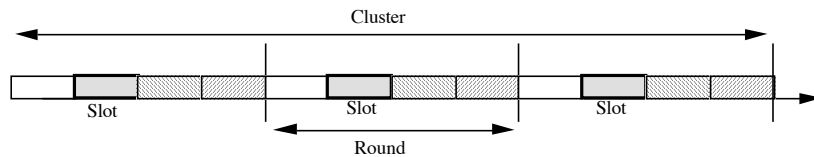
- Scénario montrant les principaux cas :



Date	Événements sur le site i
0	TRT(i)=TTRT, LC(i)=0, le jeton démarre.
T1	Le jeton arrive en avance. THT(i)=TRT(i). TRT(i)=TTRT. LC(i)=0 Début d'émission des données synch. (pendant TS(i) = t3-t2).
T2	Fin d'émission de données synch. Commencer à décrémenter THT(i)
T3	Fin d'émission de données asynch. Jeton relâché.
T4	TRT(i) expire (jeton aurait du arrive). TRT(i) =TTRT, LC(i)=1.
T5	Jeton arrive ! mais en retard puisque LC(i)=1. LC(i)=0. Début d'émission données synch. (pendant TS(i) = t6-t5).
T6	TS(i) expire. Fin d'émission de données synch. Jeton relâché.
T7	TRT(i) expire à nouveau. TRT(i) =TTRT. LC(i)=1.

La couche liaison avec contrôle d'accès : TDMA

- Fonctionnement de TDMA :
 - chaque site dispose d'un créneau temporel pour émettre, pendant lequel il est propriétaire du bus,
 - les accès sont gérés sur la base d'une horloge globale : chaque site sait à quel instant il doit émettre ou recevoir,
 - problèmes : pas d'arbitrage par priorité, pas dynamique



- Exemple d'utilisation, le field bus TTP :
 - ordonnancement statique *off line* par construction d'une table,
 - les sites sont identifiés par la date d'émission : il n'y a pas de champ adresse dans les trames,
 - chaque site dispose de « slots » **exclusifs** sur le bus, l'ordonnancement peut donc être testé indépendamment sur chaque site,

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)**
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- Ordonnancement de tâches communicant par messages

La couche 2 sans contrôle d'accès : Gestion des collisions

- Gestion déterministe des collisions :
 - **Prévention** : CSMA/CA , utilisé par le bus CAN en version CSMA/BA (BA pour *bit wise arbitration*)
 - **Détection** et traitement : CSMA/DCR (*Deterministic Collision Resolution*), variante de CSMA/CD, *Carrier Sense Multiple Access/Collision Detection* (utilisé par Ethernet),
 - En cas de collision, les essais de reprise du bus se font suivant des priorités

- Pour des raisons de coût (domotique) et/ou de débit (multimédia), on essaie de plus en plus d'utiliser Ethernet. Nombreux travaux de recherche.

La couche 2 : Ethernet: CSMA/CD

- Partagé/segmenté : hubs / switches

La couche 2 : *Rappel : CSMA/CD*

- L'algorithme de détection des collisions n'est pas défini, Ethernet utilise BEB (Binary Exponential Back-off) qui n'est pas déterministe
- L'algorithme BEB :
 - Les collisions sont détectées dans des fenêtres (*collision windows, slots*),
 - Chaque site gère un compteur de collisions C, initialisé à 0.
Algorithme :
 - $C = C + 1$ à chaque collision,
 - Si $C = 16$, le message est marqué « *undeliverable* » et éliminé,
 - Puis on attend b unités de temps avant le prochain incrément de C. b prend une valeur **aléatoire** allant de 0 à $2^{\min(C,10)} - 1$, l'attente est donc de 1024 slots maximum,
- Inconvénients de l'algorithme BEB :
 - Le WCET est très long : l'unité de temps est égale à 512 *bits time*, soit :
 - 51.2 μs à 10 Mbits,
 - 5.12 μs à 100 Mbits

La couche 2 : *Ethernet déterministe : DCR*

- Déterministe, recherche des collisions dans un arbre binaire, c'est à dire que les messages sont hiérarchisés suivant leur priorité, ce mécanisme permet le calcul du délai max. d'attente de retransmission pour un message
- En mode normal, on utilise 802.3, mais dès la détection d'une collision, on passe en DCR, et on revient en mode 802.3 quand toute les trames ont été réémises
- Inconvénient : le WCET pour le temps de transfert est beaucoup plus mauvais que le cas moyen. Donc une analyse en terme de WCET conduit à l'utilisation d'une faible bande passante

La couche 2 : Ethernet déterministe : DCR

- BEB est remplacé par un algorithme déterministe de gestion des collisions appelé DCR (*Deterministic Collision Resolution*) :
 - chaque site est repéré par un numéro (son adresse ou autre caractéristique) et connaît le nombre de sites du réseau,
 - DCR effectue une suite de partitionnements dichotomiques sur les sites qui ont le droit de réémettre,
 - on appelle « époque » l'intervalle de temps qui s'écoule entre la collision initiale et la fin de son traitement,
 - on peut donner une borne supérieure à l'époque (au temps de latence),
 - Avantages/inconvénients : assure qu'après une collision toutes les trames concernées sont ré-émises dans un délai donné. Problème : pendant ce temps, les autres stations n'ont pas accès au médium.

PERFORMANCES

- soit :
 - S le nombre de sites,
 - U la durée de transmission du message de longueur maximale,
 - TC la tranche Canal,Durée max. d'une époque = $S (TC + U) - TC$ (en mode général)

CSMA/DCR : Exemple (1)

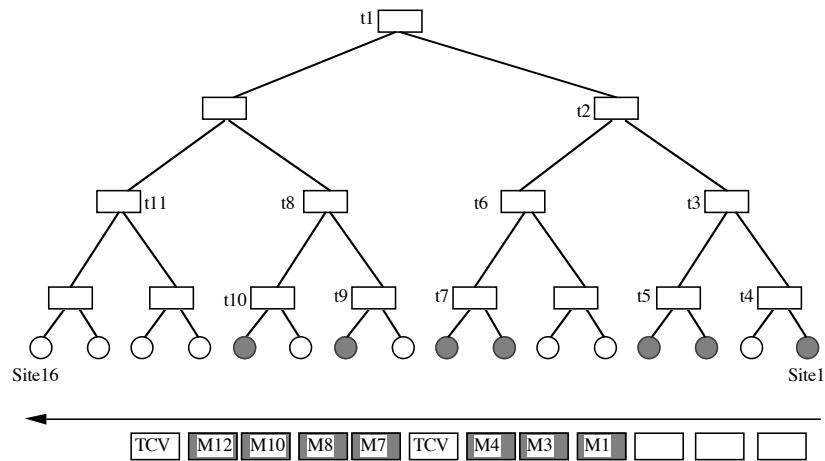
- Exemple : collision entre 7 sites S1, S2, S3, S7, S8, S10 et S12.

Après collision, un site ne réémet que si il fait partie du groupe prioritaire (prédéterminé). Le traitement se fait sur tous les sites car on ne connaît pas l'origine de la collision.

- t1 : collision initiale entre S1, S2, S3, S7, S8, S10 et S12. Seuls les sites du premier groupe émettent à nouveau (S1, S2, S3, S7 et S8).
- t2 : seconde collision entre S1, S2, S3, S7 et S8.
- t3 : les sites S1, S2 et S3 émettent. Collision.
- t4 : S1 est seul dans le nouveau groupe. Il émet.
- t5 : S3 et S4 émettent l'un après l'autre.
- t6 : on passe au groupe composé des sites S5 à S8. S5 et S6 n'ont rien à émettre -> tranche canal vide.
- t7 : S7 et S8 émettent l'un après l'autre.
- etc...

CSMA/DCR : Exemple (2)

- Schéma d'émission des trames :



Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- Ordonnancement de tâches communicant par messages

Les messages

• Définitions :

- un message contient des données et des informations de contrôle,
- une transaction est la suite d'actions nécessaire à l'émission d'un message, elle est dite atomique si elle ne peut pas être découpée,
 - o si le message est découpé en paquet : la durée de la transaction est la différence entre la date d'émission du premier et la date de réception du dernier
- Efficacité d'un protocole de transfert :

Temps de transfert des données / Durée de la transaction

• Messages temps réel:

- messages entre entités temps réel,
- doivent respecter des contraintes temporelles et garantir la cohérence entre les entités temps réel,
- messages de type événement (*event*) ou état (*time*),

• Facteurs d'indéterminisme :

- « bufferisation » des messages, gigue sur l'émission, durée de transmission

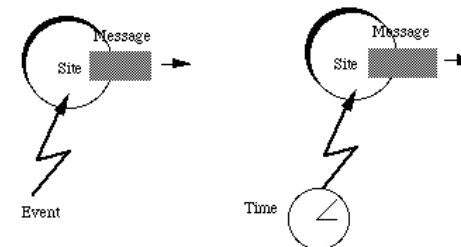
Event vs time triggered

• Event-triggered network (asynchrone) :

- les transactions sont déclenchées par des événements ,
- les destinataires connaissent l'état du système par l'intermédiaire des événements,
- bonne gestion de la bande passante en régime aperiodique, utilisée ssi nécessaire autrement,
- possibilité de surcharge en cas de rafales d'événements,

• Time-triggered network, (synchrone) :

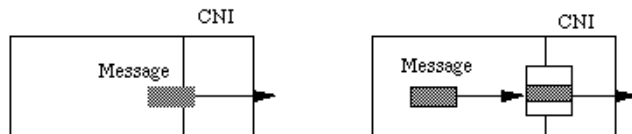
- les transactions sont déclenchées à des instants prédéfinis, bien adapté au trafic périodique
- les destinataires reçoivent des rafraîchissements périodiques sur l'état du système, perte détectées par les récepteurs,
- la charge du medium est bien déterminée, bonne gestion de la bande passante en cas de grosse charge
- contrôle de la gigue,



Gestion de La transmission

- **Qui** déclenche la transaction :

- **l'application** : les messages sont mis en file d'attente sur l'interface réseau dite CNI, Communication Network Interface, donc le comportement dépend de l'application...
- **le réseau** : l'application passe les messages dans des tampons (buffers), la CNI gère la transmission, le comportement est déterministe,



Ordonnement Des messages

- Stratégies d'ordonnement :

- déterministe, un message sera transmis avec respect de ses contraintes temporelles (contrôle industriel),
- probabiliste, les échéances sont garanties avec une probabilité connue à l'avance (multimédia),
- best effort, aucune garantie...

- Remarque sur la préemptivité :

- les messages n'acceptent généralement **pas de préemption**,
- pour utiliser des algorithmes préemptifs :
 - temps de transmission du message < durée de base d'allocation du medium
 - découper les messages

Ordonnement Statique/dynamique

- Statique :
 - basé sur une table : hors ligne, avec dispatching en ligne (TTP/C)
 - à base de priorités statiques, ordonnançabilité hors ligne, ordonancement en ligne (CAN)
- Dynamique :
 - best effort
 - planifié

Ordonancement	Avantages	Inconvénients
Statique	Respect des contraintes, sûr	Pas flexible
Dynamique	Flexible	Si best effort : respect des contraintes ?

Gestion Des messages

- Analogie tâche/message :
 - temps d'exécution -> durée de transmission
 - quelle tâche doit avoir accès au processeur -> quel message doit avoir accès au support de communication ?
 - ordonancement en ligne ou hors ligne
 - ordonancement statique ou dynamique
- On retrouve le même type de contraintes temporelles que sur les tâches :
 - Contraintes strictes sur les messages critiques (alarme),
 - Contraintes faibles sur les messages non critiques (fin d'E/S disque),
 - Les techniques de gestion du temps sont **basées sur celles de l'ordonancement des tâches** (EDF, RM , ..)
- Un message qui arrive après son échéance est considéré comme perdu (un relevé de mesure qui arrive trop tard n'est pas considéré comme correct même si il est exact),

Ordonnancement Des messages

- Différences tâche/message :
 - La transmission est généralement **non préemptible**,
 - Contraintes imposées par la couche MAC :
 - **Période** minimale (exemple : TDMA),
 - **Gigue** (exemple : irrégularités dans les dates d'arrivée d'un jeton),

- Type de protocoles mis en œuvre :
 - Pour les messages à échéances strictes :
 - protocoles avec garantie,
 - Pour les messages à contraintes temporelles faibles:
 - protocoles *best effort* (minimiser les retards et le nombre des messages en retard),

Contraintes temporelles Et commutation de paquets

- Rappels :
 - l'application (abonné) ne connaît pas les protocoles utilisés,
 - l'application demande l'établissement d'une connexion gérée par un contrat de *QoS*, la connexion est gérée par les nœuds du réseau,
 - les applications ne se consultent pas pour partager l'accès au support, le réseau gère la communication (intelligence dans le réseau).

- Exemple, ATM :
 - Prise en compte des contraintes du type *QoS* :
 - CBR , constant bit rate
 - RTVBR, real time variable bit rate

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès (exemple : jeton temporisé)
 - b. sans contrôle d'accès (exemple : CSMA/DCR)

- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement

- **Exemple de bus temps réel : le bus CAN**

- Ordonnancement de tâches communicant par messages

Contraintes temporelles Et réseaux locaux

- Les extrémités (les sites) gèrent l'accès au réseau via la couche MAC (*Medium Access Control*). L'accès se fait par compétition ou consultation (jeton).
- La trame émise est reçue (sauf collision) par le site destinataire, avec ou sans passage par un pont, ou autre.
- Les mécanismes temporels sont implantés sur les sites (priorité des messages, temps d'utilisation du jeton, ...).

Le bus CAN

- Marché : 50 millions de véhicules/an,
- Bus CSMA/BA, performances suivant déploiement (sur fil de cuivre) :
 - De 1 Mbit/s sur 30 m à 10 Kbit/s sur 5Km (temps par bit : de 1 à 100 us), 8 octets max. par trame
 - adapté à son domaine, émission depuis :
 - boîte automatique , période 15 ms
 - suspension , période 20 ms
- Définition des couches physique (*off chip hardware*) et DLL (*on chip hardware*), sans précision de leur implémentation (filaire, HF, IR, ...) la couche application est à spécifier par l'utilisateur
- Arbitrage non destructif (contrairement à celui d'Ethernet) géré sur une base *bitwise* pour une disponibilité maximale du bus.

Le bus CAN

- Point fondamental :
 - synchronisation au niveau bit,
 - chaque nœud scrute la valeur du bit transmis.

- Notion de bits dominants/récessifs :
 - La priorité est présentée dans le message bit de poids fort en tête.
 - Le bit dominant s'impose (ici zéro, dans notre exemple le bus CAN est dit *wired AND*). Un site qui envoie un bit récessif sur le bus perd l'accès au bus.
 - Un site qui perd le bus devient récepteur.

Le bus CAN : Arbitrage

- Chaque site (éventuellement un micro-contrôleur...) émet une trame puis écoute :
 - si la valeur lue par ce site (le champ identificateur) est différente de celle émise, c'est qu'il a perdu le bus, il devient récepteur
 - il faut donc un aller et retour du signal avant une nouvelle émission,

- Un **problème pour l'acquittement** : chaque station qui reçoit une trame, l'acquiesce sur le ACK Slot :
 - si ACK est positionné, alors au moins un site a reçu la trame, mais ce n'est peut-être pas le destinataire : acquittement non fiable

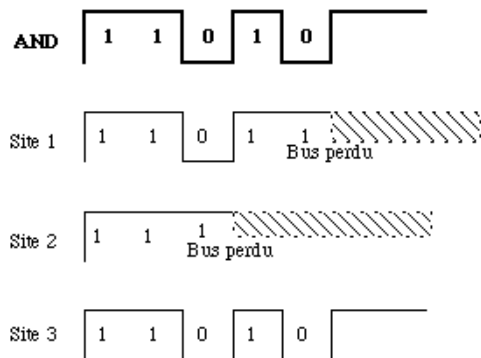
Bus et réseaux temps réel

Le bus CAN : Gestion des priorités

- Exemple sur un bus CAN du type *bitwise AND* (Motorola, Siemens) :

Site	1 (le moins prioritaire)	2	3 (le plus prioritaire)
Priorité	27 ou 11011 ₂	28 ou 11100 ₂	26 ou 11010 ₂

- Etat du bus CAN :

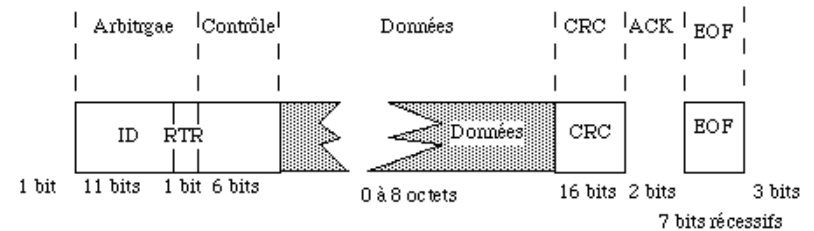


Bus et réseaux temps réel

Le bus CAN : Format des messages

- Messages de format fixe (*frames*) dont un champ détermine la priorité,

- Format des messages :



- Les principaux champs d'un message :

- Le champ ID (codé sur 11 ou 29 bits) :

- identifie le contenu du message : chaque site **vérifie** si le contenu du message le concerne, on ne transmet ni l'adresse de l'émetteur ni celle du destinataire,
- donne la priorité du message (0 le plus prioritaire en *wired AND*).

Bus et réseaux temps réel

Le bus CAN : Ordonnançabilité (1)

- **La durée de transmission** des messages varie à cause du *bit stuffing* (bits de bourrage, pour avoir des fronts montants réguliers) qui fait varier la taille de la trame :

- pour n bits, la taille de la trame **est au plus** :

$$L_{\max} = 13 + 34 + (8*n) + ((34 + (8*n) - 1) / 4) \text{ bits}$$

(temps de propagation directement proportionnel)

- En effet :

- taille de la trame au niveau MAC : $13 + 34 + (8*n)$ bits,
- si 5 bits identiques se suivent : ajout d'un bit de signe contraire, ce bit sera supprimé à la réception,
- le stuffing ne se fait pas sur CC, ACK ni fin de trame,
- dans le **pire cas**, le nombre de bits de bourrage est donc :

$$((34 + (8*n) - 1) / 4)$$

Bits de données	0	8	16	24	32	40	48	56	64
L_{\max}	55	65	75	85	95	105	115	125	135
Efficacité (%)	0	12	21	28	34	38	42	45	47

Bus et réseaux temps réel

Le bus CAN : Ordonnançabilité (2)

- **Pas de préemption sur un message déjà émis :**

On reprend les formules établies pour l'ordonnançabilité des tâches (priorité fixe), en prenant en compte le fait que les messages ne sont pas préemptibles (coefficient B_i) :

condition suffisante pour le message T_N :

$$\sum_{i=1}^N (C_i / T_i) + B_i / T_i < N (2^{1/N} - 1)$$

avec : $B_i = \max_{j=1, j \neq i} (C_j)$ où j sont les messages **moins** prioritaires que i, i qui peut être bloqué au plus une fois par un message moins prioritaire, (un message en cours d'émission)

...	Blocage pour 1	1	2	3	4
-----	----------------	---	---	---	---

- On ajoute la gigue (coefficient J_i) :

$$\sum_{i=1}^N (C_i / T_i) + (B_i + J_i) / T_i < N (2^{1/N} - 1)$$

Le bus CAN : Ordonnançabilité

- Résumé:

- T_i est la période du message
- C_i est le temps de transfert du message :
 $47 + (8*n) + ((34 + (8*n) -1) / 4) \text{ bits} * (\text{temps transfert bit})$
- B_i est le temps de transfert du message le plus long parmi tous ceux qui sont moins prioritaires que i
- Influence de la gigue, J_i , le temps de réponse est $R_i = J_i + W_i$

$$\text{Avec } W_i = C_i + B_i + \sum_{j=1}^{i-1} (W_j + J_j) / T_i \quad C_j$$

Plan

- Le temps dans les bus et réseaux
 - 1 Simplification du modèle ISO
 - 2 Couches 1 et 2 : vers le déterminisme temporel
 - a. avec contrôle d'accès
 - b. sans contrôle d'accès
 - 3 Caractéristiques des messages temps réel
 - 4 Exemples
 - a. le bus CAN
 - b. divers**
- Ordonnancement de tâches communicant par messages

Exemple FDDI : Principaux paramètres(1)

- Technique d'accès : jeton temporisé (cf. paragraphe précédent sur ce thème)
- **TTRT**, *target token rotation time*, est la durée maximale d'un tour d'anneau par le jeton. Défini à l'initialisation du réseau, il résulte d'un compromis entre :
 - bande passante (temps d'accès au bus, qui doit être suffisant pour envoyer un message),
 - respect des échéances , assuré si $D \geq 2 * TTRT$,
- FDDI est un *timed token access protocol* : chaque site dispose de deux timers et d'un compteur :
 - **TRT**, *token rotation timer*, initialisé à TTRT, décrémenté jusqu'à 0 ou jusqu'à l'arrivée du jeton,
 - **LC**, *late counter*, compte le nombre d'expirations depuis que le jeton est arrivé :
 - Si $LC = 0$ quand le jeton arrive, OK,
 - Sinon, le jeton est en retard, si $LC > 1$, on régénère un jeton
 - **THT**, *token holding timer* : initialisé à TRT quand le jeton arrive, sert pour les transferts asynchrones

Exemple FDDI : Principaux paramètres (2)

- Après réception du jeton, un site peut émettre des trames synchrones pendant H unités de temps.
- Mécanisme mis en œuvre :
 - Si $LC = 0$, THT est initialisé à TRT et TRT à TTRT, on peut transmettre pendant au plus H unités de temps des trames synchrones, puis ensuite des trames asynchrones tant que THT et TRT sont positifs. En effet THT est décrémenté dès l'envoi de trames asynchrones.
 - Sinon ($LC = 1$, en général), TRT continue à décrémenter, sans être initialisé à TTRT, on transmet les trames synchrones seules, pendant H.

Ethernet et le temps réel

- Powerlink:

Plan

- Le temps dans les bus et réseaux
 - 1- Simplification du modèle ISO
 - 2- Couches 1 et 2 : vers le déterminisme temporel
 - c. avec contrôle d'accès (exemple : jeton temporisé)
 - d. sans contrôle d'accès (exemple : CSMA/DCR)
- Les messages temps réel
 - 1- Définitions
 - 2- Ordonnancement
- Exemple de bus temps réel : le bus CAN
- **Ordonnancement de tâches communicant par messages**

Ordonnancement de tâches communicant Par messages

- Deux approches :

1- le réseau est assimilé à un site et les messages assimilés à des tâches. Notion d' « analyse holistique » (Tindell 1994) : ordonnancement / ordonnancement conjoint de tâches et de messages.

2- le réseau est vu comme une ressource partagée. On peut ramener le problème de l'ordonnancement de tâches communicant par messages à celui de l'ordonnancement de tâches dépendantes.

- Un certain nombre d'hypothèses , en particulier :

- un tâche ne (re)démarre que si elle a reçu tous les messages qu'elle attend,
- elle ne transmet des messages qu'à la fin de son temps d'activation,

- On va donc faire quelques rappels sur l'ordonnancement de tâches dépendantes.

Rappel sur l'ordonnancement de tâches dépendantes (EDF)

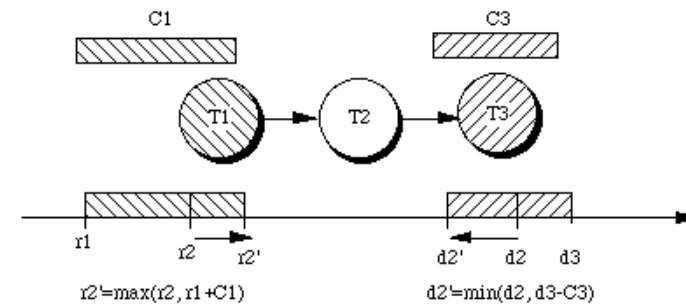
- Modification des paramètres pour prendre en compte les dépendances entre tâche EDF :

- chaque tâche doit avoir une échéance plus petite que celles des tâches qu'elle précède,
- une tâche ne peut être activée que si toutes ses précédentes ont terminé leur activation. Donc la date de réveil d'une date doit être supérieure à celle de ses précédentes à laquelle a été ajoutée la durée d'activation, c'est à dire :

$$r_i^* = \max \{ r_i, \max \{ r_j^* + C_j \} \}, \quad \text{pour } j \text{ tel que } T_j \text{ précède } T_i$$

$$D_i^* = \min \{ D_i, \min \{ D_j^* - C_j \} \}, \quad \text{pour } j \text{ tel que } T_i \text{ précède } T_j$$

- Exemple pour trois tâches T1, T2 et T3 :



Rappel sur l'ordonnancement de tâches dépendantes : Exemple (1)

- L'ordonnancement est EDF, on reprend le schéma de dépendance précédent :



- Dans lequel on caractérise ainsi les trois tâches T1, T2 et T3, le coefficient C (coût) de T2 est modulable :

Tâche	Coût	Echéance	Réveil
T1	5	15	10
T2	x	40	12
T3	10	50	20

(Les relations de dépendance sont : T1 précède T2 qui précède T3.)

- Quel peut être le coût maximum de T2 pour que l'ensemble des tâches T1, T2 et T3 soit ordonnançable ?

Rappel sur l'ordonnancement de tâches dépendantes : Exemple (2)

- Calculs des nouvelles dates de réveil et échéances :

- r^* pour T2 précédée par T1 :

- $r^*_2 = \max \{12, \max\{r^*_1 + C_1\}\} = \max \{12, 10 + 5\} = 15$

- r^* pour T3 précédée par T2 :

- $r^*_3 = \max \{20, \max\{r^*_2 + C_2\}\} = \max \{20, 15 + x\}$

- D^* pour T2 précédant T3 :

- $D^*_2 = \min \{D_2, \min\{D^*_3 - C_3\}\} = \min \{45, (50 - 10)\} = 40$

- D^* pour T1 précédant T2 :

- $D^*_1 = \min \{15, \min\{D^*_2 - C_2\}\} = \min \{15, (40 - x)\}$

- Or, il faut $x < D^*_2 - r^*_2$ (coût < échéance - réveil), donc $x < (40 - 15)$, c'est-à-dire : $x < 25$

- Il faut aussi :

$$(\text{échéance T1}) < (\text{échéance T2}) < (\text{échéance T3}) \text{ et}$$

$$(\text{réveil T3}) > (\text{réveil T2}) > (\text{réveil T1})$$

*Ordonnement
de tâches communicant
par messages*

• Si la **dépendance** entre tâches est une **communication** par message, on ajoute le délai Δ_j^i dû à la transmission du site i vers le site j :

- $r_i^* = \max \{ r_i, \max \{ r_j^* + C_j + \Delta_j^i \}, \text{ pour } j \text{ tel que } T_j \text{ précède } T_i$

• Si les tâches T_i et T_j sont sur le même site, on suppose que le temps de transfert de messages est nul, c'est à dire :

$$\Delta_j^i = 0,$$

On revient à la formule de base pour tâches dépendantes.

*Ordonnement
de tâches communicantes :
exemples*

• Nous allons illustrer les conditions d'ordonnabilité (EDF) de tâches communicant par messages à l'aide de deux exemples :

- dans le premier, étant données les contraintes temporelles sur les tâches, on va calculer le temps maximum que l'on peut accorder aux transferts de messages,
- dans le second, étant donnés les temps de transfert des messages, on va modifier les dates de réveil et calculer quelle peuvent être les meilleures échéances des tâches qui communiquent,

Bus et réseaux temps réel

Exemple 1 (1)

- Soit les 5 tâches, ordonnancées EDF :

Tâche T_i	r_i	C_i	D_i
T1	0	50	250
T2	250	100	350
T3	0	100	250
T4	0	50	500
T5	0	150	600

(unité : μs)

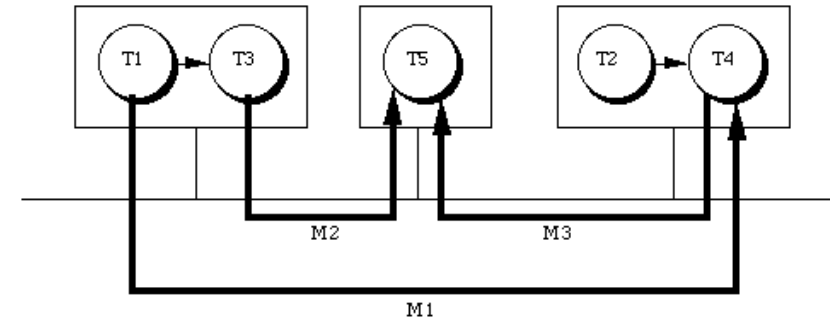
T3 dépend de T1, T4 dépend de T2

- Ces tâches sont réparties sur trois sites reliés par un bus CAN à 1 Mbits/s :
 - les délais de traversée des couches réseau sont négligeables,
 - un tâche ne démarre que si elle a reçu tous les messages qu'elle attend et ne transmet des messages qu'à la fin de son exécution,

Bus et réseaux temps réel

Exemple 1 (2)

- Répartition et sens de transfert des messages :



Remarque : les tâches sont ordonnancées sur chacun des sites.

- Caractéristiques des messages :

Message	Taille	Emission/Destination
M1	2 octets	T1 → T4 (S1 vers S3)
M2	8 octets	T3 → T5 (S1 vers S2)
M3	4 octets	T4 → T5 (S3 vers S2)

- Problème :

Quels sont les délais de communication maximums pour que les tâches restent ordonnancées ?

Exemple 1 (3) :
Calcul d'ordonnançabilité

- Pour calculer les r_i^* , on commence par les tâches sans précédentes :
 - $r_i^* = \max \{ r_i, \max \{ r_j^* + C_j + \Delta_j^i \} \}$, pour j tel que Tj précède Ti
- Pour calculer les D_i^* , on commence par les tâches qui n'en précèdent aucune,
 - $D_i^* = \min \{ D_i, \min \{ D_j^* - C_j \} \}$, pour j tel que Ti précède Tj,

Exemple 1 (4) :
Calcul des nouvelles échéances

- Calculs des nouvelles échéances. On commence par les tâches qui n'en précèdent aucune. Ici on a :
 - T5 ne précède aucune tâche,
 - T4 précède T5,
 - T3 précède T5,
 - T2 précède T4,
 - T1 précède T3 et T4,

Pour T5 : $D_5^* = \min \{ 600, \min \{ \emptyset \} \} = 600$ (ne précède personne),

Pour T4 : $D_4^* = \min \{ D_4, \min \{ D_5^* - C_5 \} \} = \min \{ 500, \min \{ 600 - 150 \} \}$

$$D_4^* = 450$$

Pour T3 : $D_3^* = \min \{ D_3, \min \{ D_5^* - C_5 \} \} = \min \{ 250, \min \{ 600 - 150 \} \}$

$$D_3^* = 250$$

Pour T2 : $D_2^* = \min \{ D_2, \min \{ D_4^* - C_4 \} \} = \min \{ 350, \min \{ 450 - 50 \} \}$

$$D_2^* = 350$$

Pour T1 : $D_1^* = \min \{ D_1, \min \{ D_3^* - C_3, D_4^* - C_4 \} \}$

$$= \min \{ 250, \min \{ 250 - 100, 450 - 50 \} \}$$

$$D_1^* = 150$$

Bus et réseaux temps réel

Exemple (1)5: Calculs des nouvelles dates de réveil

- $r_i^* = \max \{ r_i, \max \{ r_j^* + C_j + \Delta_i^j \} \}$, pour j tel que Tj précède Ti
 - T1 et T2 n'ont pas de précédente,
 - T3 est précédée par T1
 - T4 est précédée par T1 et T2
 - T5 est précédée par T3 et T4

Pour T1 : $r_1^* = \max \{ 0, \emptyset \} = 0$

Pour T2 : $r_2^* = \max \{ 250, \emptyset \} = 250$

Pour T3 : $r_3^* = \max \{ 0, \max \{ 0 + 50 + \Delta_3^1 \} \} = 50$

En effet $\Delta_3^1 = 0$ puisque T1 et T3 sont sur le même site,

Pour T4 : $r_4^* = \max \{ 0, \max \{ 0 + 50 + \Delta_4^1, 250 + 100 + \Delta_4^2 \} \}$

$= \max \{ 0, \max \{ 0 + 50 + \Delta_4^1, 250 + 100 \} \}$

En effet $\Delta_4^2 = 0$ puisque T2 et T4 sont sur le même site,

$= \max \{ 50 + \Delta_4^1, 350 \}$

Pour T5 : $r_5^* = \max \{ 0, \max \{ (50 + 100 + \Delta_5^3),$

$\max \{ 0 + 50 + \Delta_5^1, 250 + 100 \} + 50 + \Delta_5^4 \} \}$

$= \max \{ 150 + \Delta_5^3, \max \{ 100 + \Delta_5^4 + \Delta_5^1, 400 + \Delta_5^4 \} \}$

Bus et réseaux temps réel

Exemple (1)6: récapitulatif

• On obtient :

tâche	C_i	r_i^*	D_i^*
T1	50	0 (personne ne le précède)	150 (précède T3 et T4)
T2	100	250 (personne ne le précède)	350 (précède T4)
T3	100	50 (précédée par T1)	250 (précède T5)
T4	50	$\max(0, 50 + \Delta_4^1, 350)$ (précédée par T1 et T2)	450 (précède T5)
T5	150	max $(0, 150 + \Delta_5^3, \max(400, 100 + \Delta_5^4) + \Delta_5^4)$ (précédée par T3 et T4)	600

(unité : μs)

Exemple 1(7)

Calcul des limites des délais de transmissions

- Quels sont les limites des valeurs des délais de communication pour que les tâches soient ordonnancables ?
 - T4 et T5 dépendent du réseau, pour qu'elles manquent leurs échéances, il faut :
 - $r_4^* > D_4^* - C_4$
 - $r_5^* > D_5^* - C_5$
- On prend les valeurs calculées dans le tableau précédent :
 - $\max(350, 50 + \Delta_4^1) > 400$, c'est à dire :
 - $\Delta_4^1 > 350$
 - $\max\{150 + \Delta_5^3, \max\{100 + \Delta_5^4 + \Delta_4^1, 400 + \Delta_5^4\}\} > 450$, c'est à dire :
 - $\Delta_5^3 > 300$
 - $\Delta_5^4 > 50$
 - $\Delta_4^1 > 300 + \Delta_5^4 > 350$
 - on en déduit les **délais de communications à ne pas dépasser** :
 - $\max \Delta_4^1 = 300$
 - $\max \Delta_5^3 = 300$
 - $\max \Delta_5^4 = 50$

Exemple 1(fin)

- On rappelle les tailles des messages à transférer :

Message	Taille	Emission/Destination
M1 (de T1 à T4)	2 octets	T1 → T4 : S1 vers S3
M2 (de T3 à T5)	8 octets	T3 → T5 : S1 vers S2
M3 (de T4 à T5)	4 octets	T4 → T5 : S3 vers S2

- On calcule alors le temps de transmission, en rappelant la formule pour un bus CAN :

temps de transmission pour n octets (TT) :

$$TT = 47 + (8*n) + ((34 + (8*n) - 1) / 4)$$

- On obtient :

Message entre tâches	Taille	Temps transmission max. entre machines
M1 (de T1 à T4)	2 octets	75 μ s
M2 (de T3 à T5)	8 octets	137 μ s
M3 (de T4 à T5)	4 octets	95 μ s

Bus et réseaux temps réel

Exemple 2 (1)

- On donne maintenant un jeu de tâches et, étant donnés leurs dates de réveil et leurs durées d'activation :
 - on va **prendre en compte les délais dus aux transmissions de messages** pour
 - modifier les dates de réveil ,
 - calculer leurs échéances,
- Les contraintes sont toujours les mêmes :
 - une tâche ne démarre que si elle a reçu tous les messages qu'elle attend et ne transmet de message qu'à la fin de son exécution,
 - La répartition des tâches est celle-ci :

Site A Site B Site C

Délais de transmission :

- entre A et B : 200
- entre A et C : 100
- entre B et C : 200

Bus et réseaux temps réel

Exemple 2(2)

Caractéristiques des tâches : Date de réveil , durée d'activation

- Les tâches sont caractérisées ainsi :

Tâche T_i	r_i	C_i
T1	10	50
T2	5	100
T3	0	200
T4	10	50
T5	10	30

- Il faut d'abord calculer la **modification des paramètres** r_i pour prendre en compte les dépendances dues aux communications par message, on va donc appliquer :

$$r_i^* = \max \{ r_i, \max \{ r_j^* + C_j + \Delta_i^j \} \}, \text{ pour } j \text{ tel que } T_j \text{ précède } T_i$$

- Remarques :

- T3 est précédée par T1, T4 est précédée par T1 et T2, T5 est précédée par T3 et T4, donc :

$$\Delta_3^1 = \Delta_4^2 = 0, \quad \Delta_4^1 = 100, \quad \Delta_5^3 = \Delta_5^4 = 200$$

Exemple 2 (3)
Calcul des nouvelles dates de réveil

• Calculs des :

$$r_i^* = \max \{ r_i, \max \{ r_j^* + C_j + \Delta_j^i \} \}, \text{ pour } j \text{ tel que } T_j \text{ précède } T_i$$

$$\text{Pour } T_1 : r_1^* = \max \{ 10, \emptyset \} = 10$$

$$\text{Pour } T_2 : r_2^* = \max \{ 5, \emptyset \} = 5$$

$$\text{Pour } T_3 : r_3^* = \max \{ 0, \max \{ 10 + 50 + \Delta_3^1 \} \} = 60$$

$$\text{Pour } T_4 : r_4^* = \max \{ 10, \max \{ 10 + 60 + \Delta_4^1, 5 + 100 + \Delta_4^2 \} \}$$

$$= \max \{ 10, \max \{ 70 + \Delta_4^1, 105 \} \}$$

$$= \max \{ 70 + \Delta_4^1, 105 \} = 170$$

$$\text{Pour } T_5 : r_5^* = \max \{ 10, \max \{ (60 + 200 + \Delta_5^3),$$

$$\max \{ 70 + \Delta_4^1, 105 \} + 50 + \Delta_5^4 \} \}$$

$$= \max \{ 260 + \Delta_5^3, \max \{ 120 + \Delta_5^4 + \Delta_4^1, 155 + \Delta_5^4 \} \}$$

$$= \max \{ 260 + 200, \max \{ 120 + 200 + 100, 155 + 200 \} \}$$

$$= 460$$

Exemple 2 (4)
Quelles sont les meilleures échéances possibles ?

• Récapitulatif des modifications à faire sur les dates de réveil :

Tâche T_i	r_i	C_i	r_i^*
T1	10	50	10
T2	5	100	5
T3	0	200	60
T4	10	50	170
T5	10	30	460

• Quelles doivent être les valeurs des échéances pour qu'elles soient respectées ?

- il faut : $D_i > r_i^* + C_i$, c'est à dire :

○ $D_1 > 10 + 50 = 60$

○ $D_2 > 105$

○ $D_3 > 260$

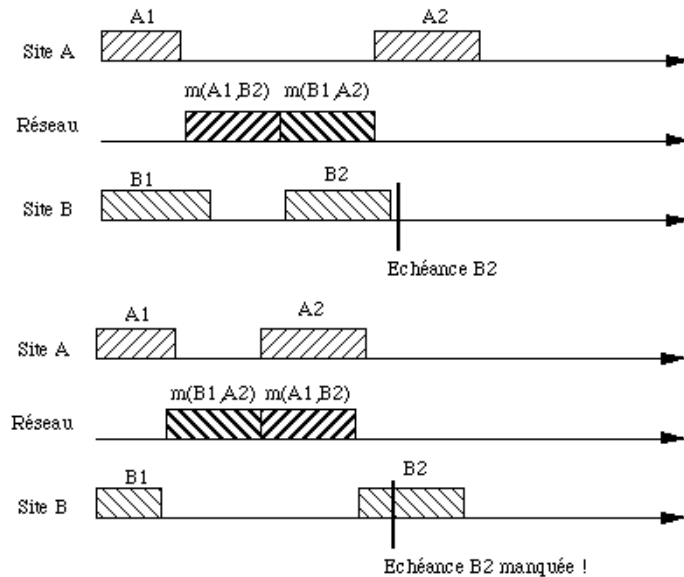
○ $D_4 > 220$

○ $D_5 > 490$

Bus et réseaux temps réel

Problèmes dûs au WCET

- Problèmes dus à la gigue sur l'activation (ou au WCET), exemple :



Dans le premier cas, la tâche B1 s'exécute en WCET, dans le second son exécution est plus courte, elle émet alors un message avant A1 et B2 manque son échéance...

Bus et réseaux temps réel

Perspectives

- - Support pour l'informatique diffuse (domotique, ...) : utilisation de Ethernet (Powerlink Ethernet, ...) sans IP ?
 - intégration du réseau dans les ressources du système, dans la conception des applications :
 - utilisation d'intergiciels temps réel (Real Time CORBA)
 - codesign matériel/logiciel
 - gestion d'une grande variation du nombre d'utilisateurs (réseaux sans fil, gestion de trafic),
 - gestion de systèmes reconfigurables : avionique, automobiles,