

# Ordonnancement temps réel dans les réseaux

Ye-Qiong SONG

*Université de Lorraine – LORIA*

# 30 ans d'ordonnancement dans les réseaux

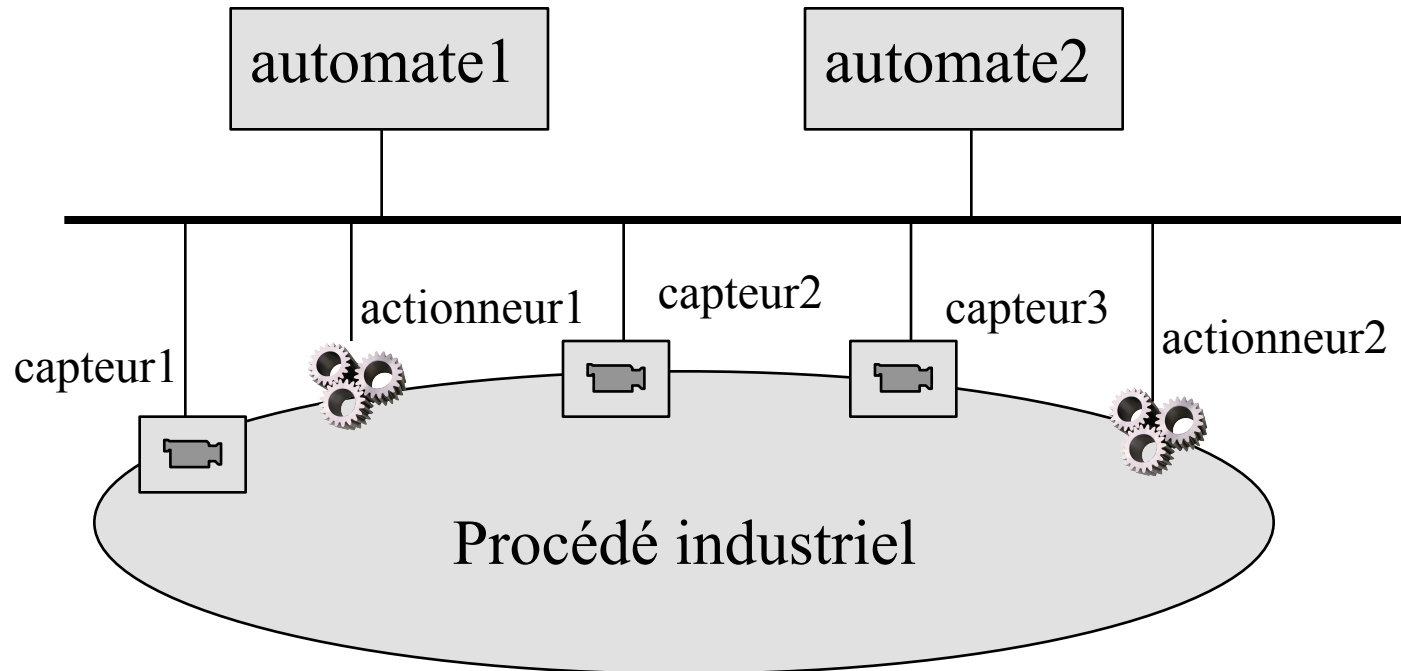
## – Plan de la présentation

---

- ◆ 72-80: ordonnancement monoprocesseur, peu de travaux d'ordonnancement sur les réseaux
- ◆ 80-90: bus de terrain (réseaux de capteurs, actionneurs et automates, filaires) et le besoin d'ordonnancer le trafic temps réel
- ◆ 90-00: CAN, bus à priorité fixe et analyse du pire temps de réponse
- ◆ 00-10: Ethernet commuté industriel (Profinet, AFDX, ...)
- ◆ 10- : Réseaux sans fil, NCS (Networked Control System) et CPS (Cyber-Physical System)
- ◆ *Réflexions sur les approches d'évaluation de temps de réponse*
- ◆ *Ordonnancement sous contrainte  $(m,k)$ -firm*

# Bus de terrain et ordonnancement

---

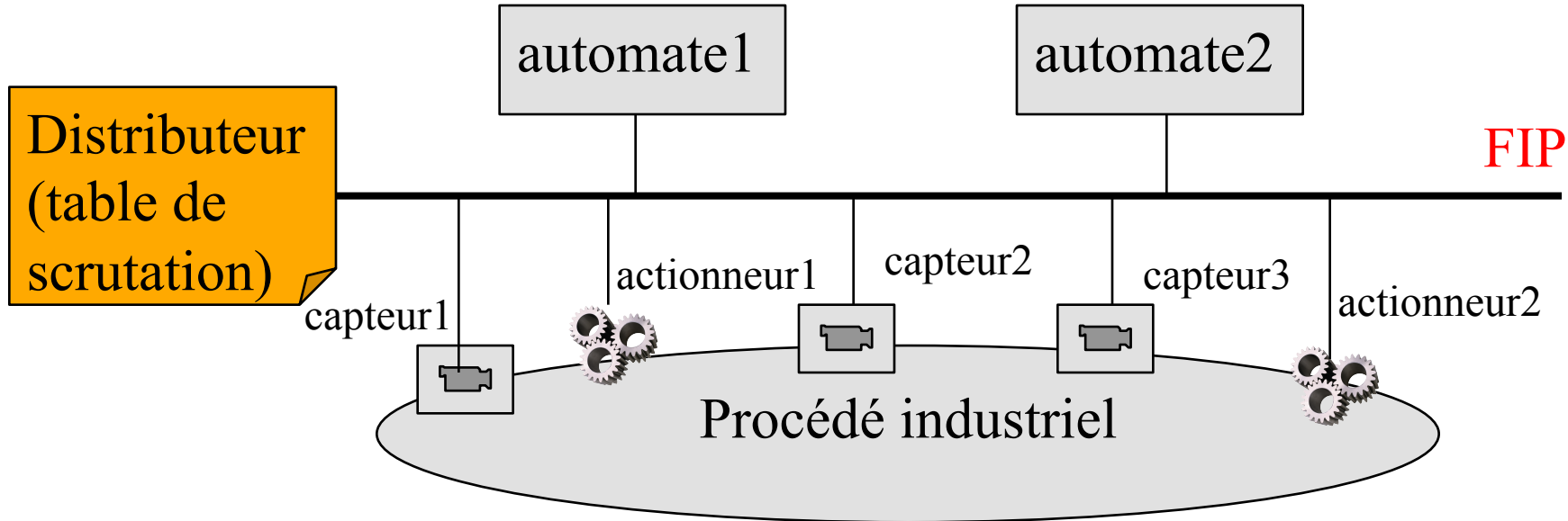


Boucles de contrôle-commande partagent le même bus de communication

Problème: **ordonnancer les demandes d'accès au bus**

Similaire aux protocoles MAC, mais avec la notion d'échéance/période

# Bus de terrain et ordonnancement



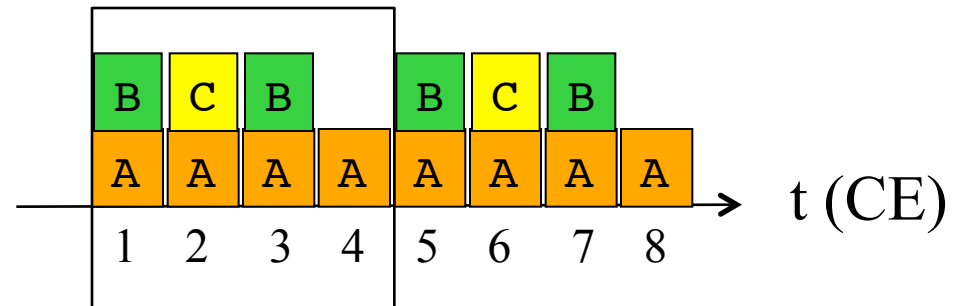
Solution de FIP: table de scrutation, modèle producteur-consommateur  
 [Thomess05]

$M_i (C_i, T_i);$

cycle élémentaire  $EC = \text{PGCD}(T_i);$

macro cycle  $MC = \text{PPCM}(T_i)$

Exemple:  $A(1, 2); B(1, 4); C(1, 8)$



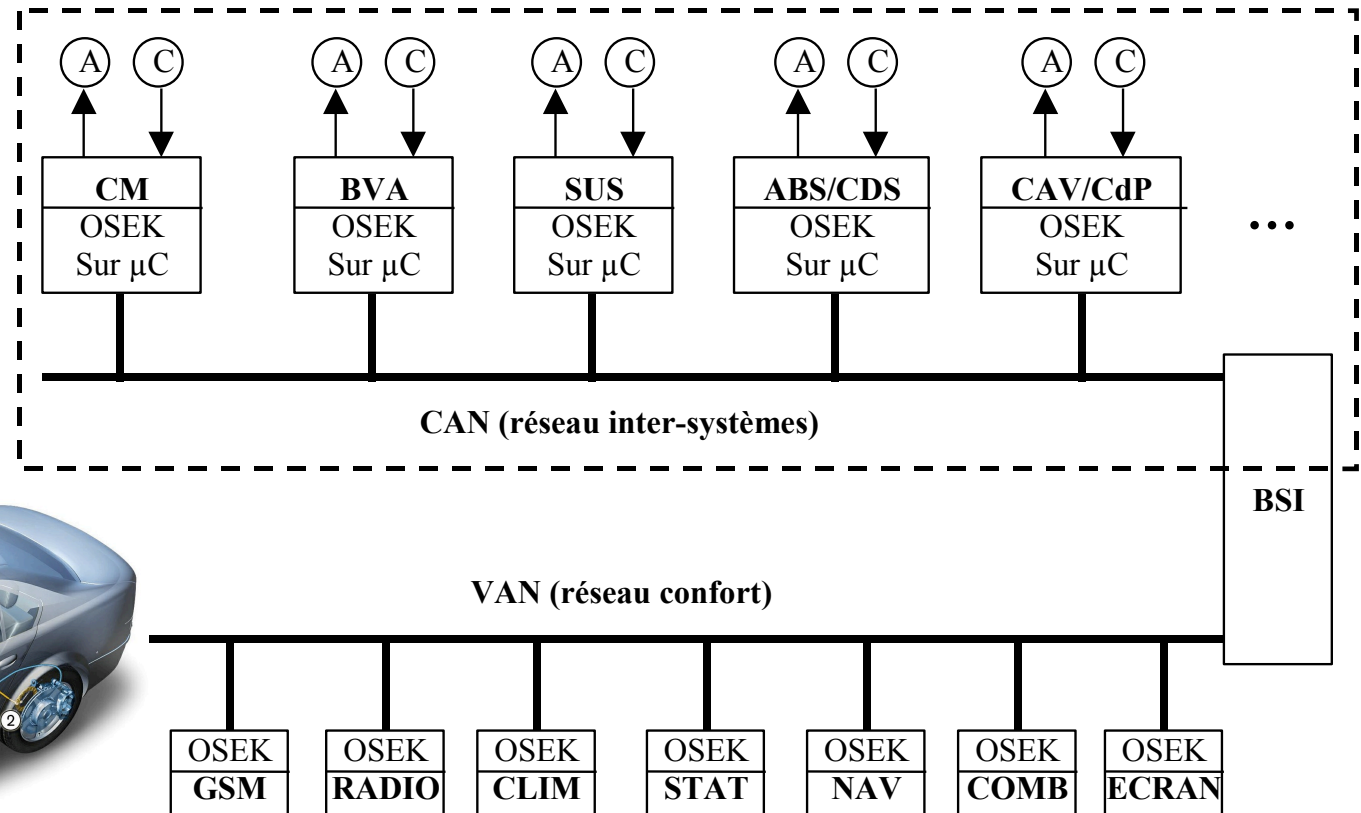
# Discussion 1: bus de terrain

---

- ◆ Réseau dédié dont les applications sont connues
- ◆ Correspond au besoin du niveau bas (terrain) des systèmes automatisés de production, une fois configurées (trafics ordonnancés), n'évoluent pas ou peu
- ◆ Exemple d'applications de FIP: TGV, LHC
- ◆ Problèmes:
  - Ordonnancement statique et hors-ligne, ne supporte pas des applications dynamiques (ajout ou retrait d'équipements, par ex)
  - Manque de la robustesse face aux aléas

# CAN (Controller Area Network)

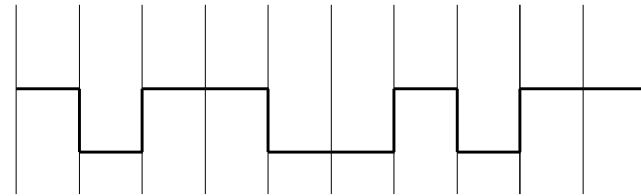
- ◆ CAN, conçu en 1983 par Bosch pour automobile, standard ISO depuis 1994



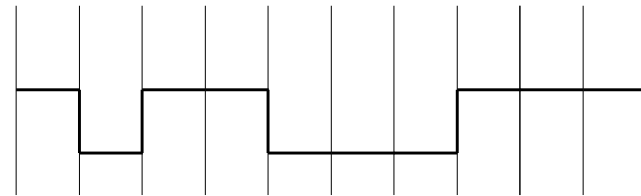
# CAN, bus à priorité fixe

- ◆ Méthode d'accès
  - Bit dominant 0 et bit récessif 1 (Codage NRZ)
  - CSMA avec arbitrage bit par bit sur le champ ID (par ET logique sur la ligne)
- ◆ Débit limité par le MAC

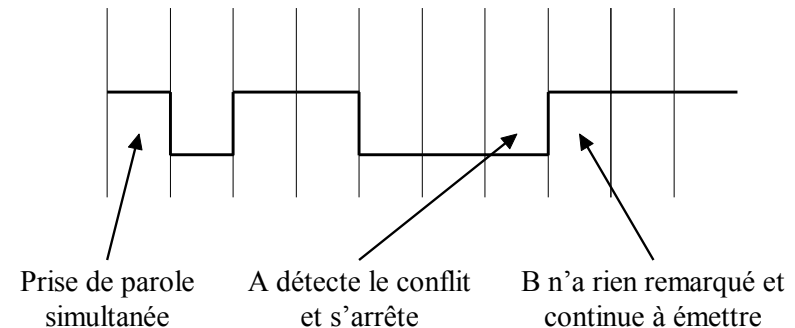
Données de la machine A



Données de la machine B

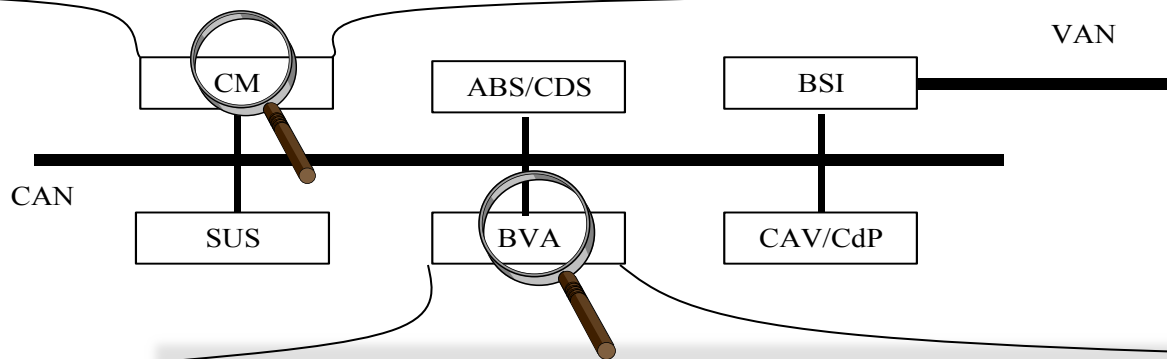


Données sur le média



# CAN: exemple d'applications

Nom de tâche	Message entrant	Durée (ms)	Période (ms)	Message généré	Charge
T_CM1		2	10	M1	0,20
T_CM2		2	20	M3	0,10
T_CM3		2	100	M10	0,02
T_CM4	M4	2	15		0,13
T_CM5	M2	2	14		0,14
T_CM6	M8	2	50		0,04
T_CM7	M6	2	40		0,05
					0,69



Nom de tâche	Message entrant	Durée (ms)	Période (ms)	Message généré	Charge
T_BVA1		2	15	M4	0,13
T_BVA2		2	50	M11	0,04
T_BVA3	M8	2	50		0,04
T_BVA4	M2	2	14		0,14
					0,36



# CAN: pire temps de réponse

---

- ◆ Temps de réponse des messages du pire cas [Tindell94] :

$$R_m = C_m + I_m + J_m$$

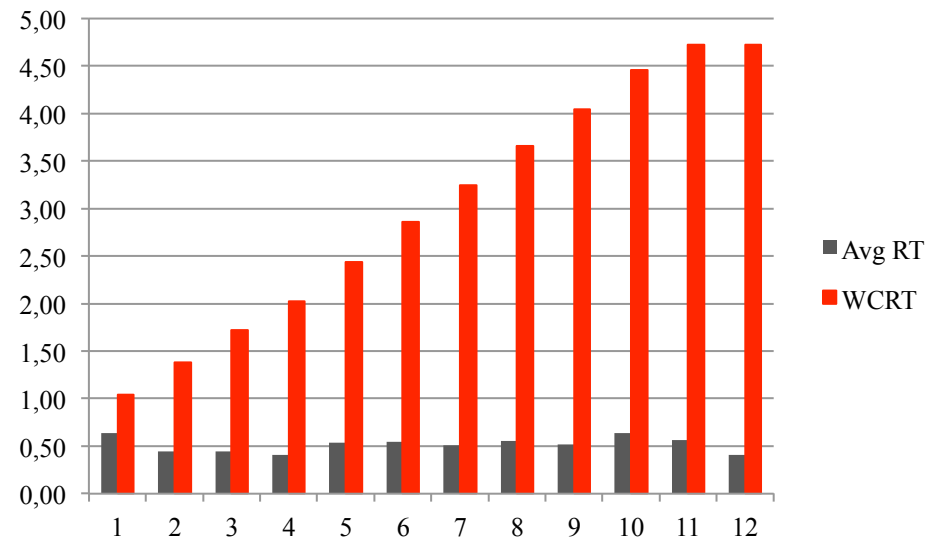
$$I_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left[ \frac{I_m^n + J_j + \tau_{bit}}{T_j} \right] C_j$$

# CAN: Application à la messagerie PSA

DLC (byte)	Periode (ms)	Priority	Avg RT	WCRT
8	10	1	0,63	1,04
3	14	2	0,44	1,38
3	20	3	0,45	1,72
2	15	4	0,41	2,02
5	20	5	0,54	2,44
5	40	6	0,54	2,86
4	15	7	0,51	3,24
5	50	8	0,55	3,66
4	20	9	0,52	4,04
7	100	10	0,64	4,46
5	50	11	0,56	4,72
1	100	12	0,40	4,72

AvgTR si M/G/1 avec priorité  
et sans préemption

WCRT: pire temps de réponse



Messagerie PSA Peugeot-Citroën  
pour les 1<sup>ères</sup> voitures avec multiplexage CAN

[Simonot-Lion95]

# Discussion 2: CAN et surdimensionnement

---

- ◆ Réseau dédié, donc applications connues
- ◆ Largement répandu dans l'automobile
- ◆ Certaine flexibilité (ajout ou suppression de messages de faible priorité a peu d'impact)
- ◆ Débit très limité à cause de l'arbitrage bit par bit
- ◆ Surdimensionnement du besoin en bande passante à cause de la considération du pire cas
  - Solution de déphasage (cf. travaux de Navet)
- ◆ Robustesse face aux erreurs de transmission
  - Garantie probabiliste du pire temps de réponse

# CAN: prise en compte des erreurs de tx

---

[Navet00]

- ◆ Erreurs de transmission augmentent le TdR

$$I_m^{n+1} = E_m(I_m^n + C_m) + B_m + \sum_{\forall j \in hp(m)} \left[ \frac{I_m^n + J_j + \tau_{bit}}{P_j} \right] C_j$$

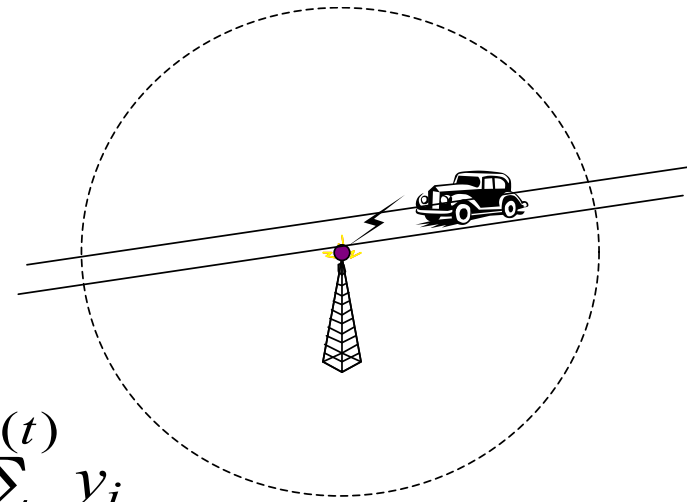
$$E_m(t) = N_{err}(t) \left( 23\tau_{bit} + \max_{\forall j \in hp(m) \cup \{m\}} C_j \right)$$

- ◆ WCDFP pour mesurer la robustesse:

$$P[N_{err}(R_{i\max}) > K_i] \quad (K_i \text{ nb max de retransmissions avant de conduire à la violation de l'échéance})$$

# CAN: modèle d'occurrence des erreurs

Erreurs individuelles et en rafale



Nombre d'erreurs durant  $[0, t]$ : 
$$N_{err}(t) = \sum_{i=0}^{N(t)} y_i$$

Avec:

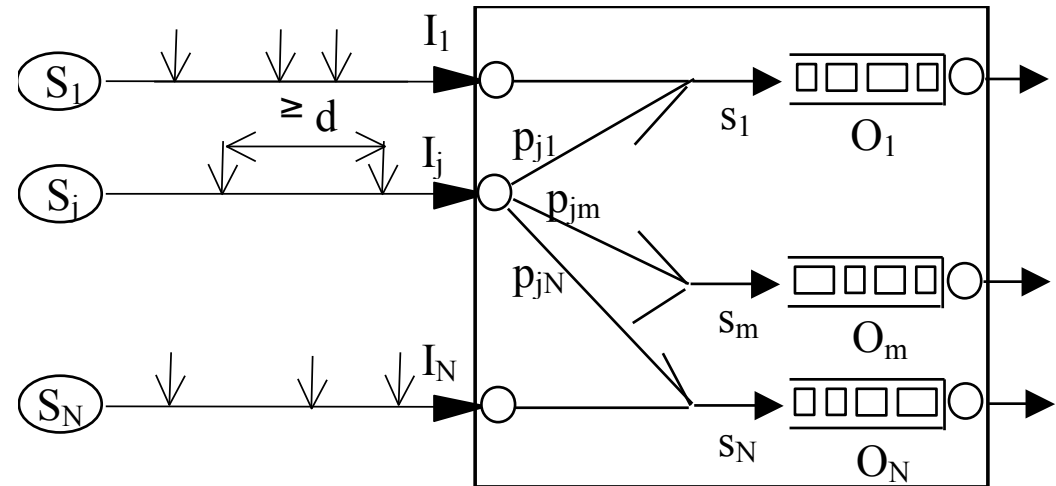
–  $N(t)$  suit une loi de Poisson composé

$$y_i = \begin{cases} u, & \text{avec une probabilité } \alpha \\ 1, & \text{avec une probabilité } 1 - \alpha \end{cases}$$

–  $u$  est une v.a. qui peut suivre une distribution quelconque

# Ethernet commuté pour communication indus.

Real-Time tasks	Non-Real-Time tasks
Control loops Alarms	HTTP services FTP services
Real-time API	HTTP, FTP, SNMP...
UDP	TCP
Internet Protocol(IP)	
Priority scheduling	
Ethernet MAC (and LLC)	
Ethernet PHY	



# Ethernet Commuté: évaluation de délai

## ◆ Hypothèse:

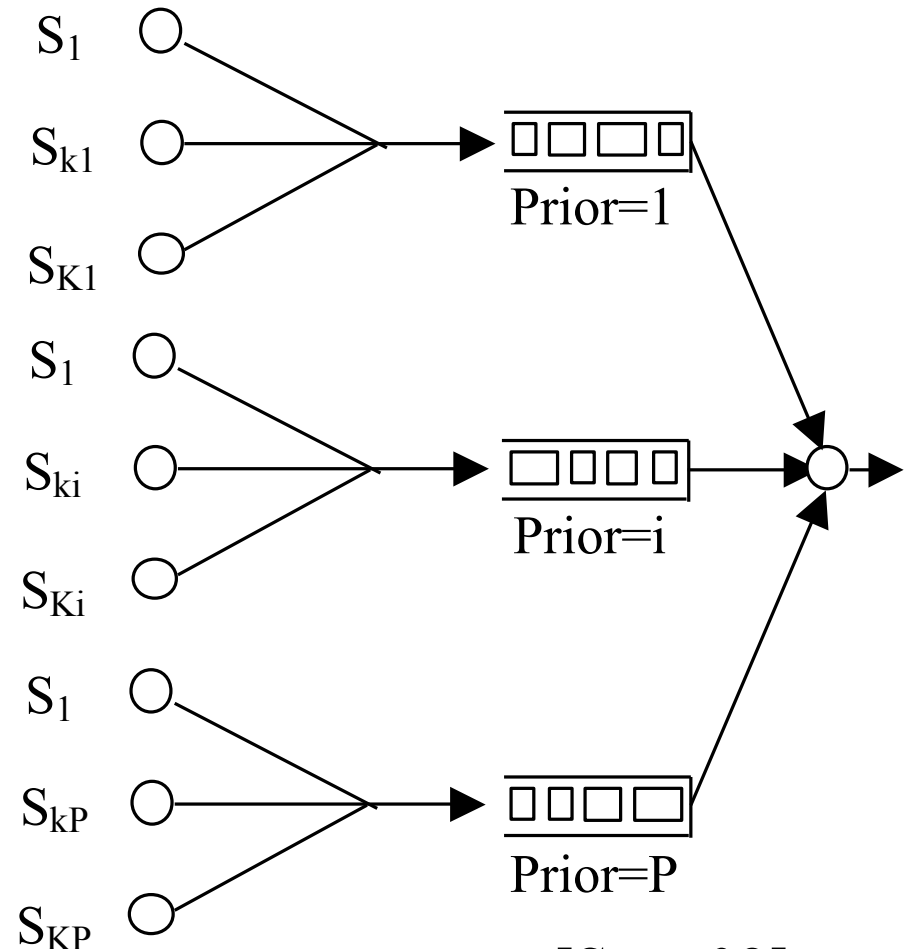
- M sources périodiques classées en P priorités ( $P < 9$ )
- Destinées au même port de sortie (i.e., nous connaissons l'application !)

## ◆ Notation:

$$\{C_i(k_i), T_i(k_i), D_i(k_i)\}$$

$$k_i = 1, 2, \dots, K_i$$

$$i = 1, 2, \dots, P$$



[Song02]

# Ethernet commuté: pire temps de réponse

- ◆ Pour  $K_i > 1$  et quelque soit l'échéance, un paquet de priorité  $i$  peut être bloqué par un autre de la même priorité
  - Paquets issues d'autres sources mais classifiés en la même priorité (effet de  $K_i > 1$ )
  - Paquets en attente, issues de la même source, et pas encore transmis au moment de l'arrivée du paquet considéré (effet de l'échéance supérieure à la période)
- ◆ **Comment définir le pire temps de réponse?**



# Ethernet commuté: pire temps de réponse

- ◆ « Busy period » de priorité  $i$  a une durée finie ssi:

$$\sum_{j=1}^i \rho_j = B_i + \sum_{j=1}^i \sum_{k_j=1}^{K_j} \frac{C_j(k_j)}{T_j(k_j)} \leq 1$$

- ◆ « Busy period » démarre avec l'hypothèse que toutes les autres sources de priorité supérieure ou égale envoient leurs paquets, et qu'un paquet de priorité inférieure vient de commencer sa tx
- ◆ Calculer le délai  $R_{i,k}(k_i)$  pour tous les  $k$  ( $k = 1, 2, \dots$ ) paquets transmis durant la « busy period »
- ◆ « Busy period » se termine quand on rencontre  $R_{i,k}(k_i) < T_i(k_i)$
- ◆ Le max de  $R_{i,k}(k_i)$  correspond au pire temps de réponse

# Ethernet commuté: pire temps de réponse

$$I_i^0(k_i) = B_i + \sum_{j=1}^i \sum_{k_j=1}^{K_j} C_j(k_j)$$

$$I_i^{n+1}(k_i) = B_i + kC_i(k_i) + \sum_{j=1}^i \sum_{\substack{k_j=1 \\ k_j \neq k_i}}^{K_j} \left[ \frac{I_i^n(k_i)}{T_j(k_j)} \right] C_j(k_j)$$

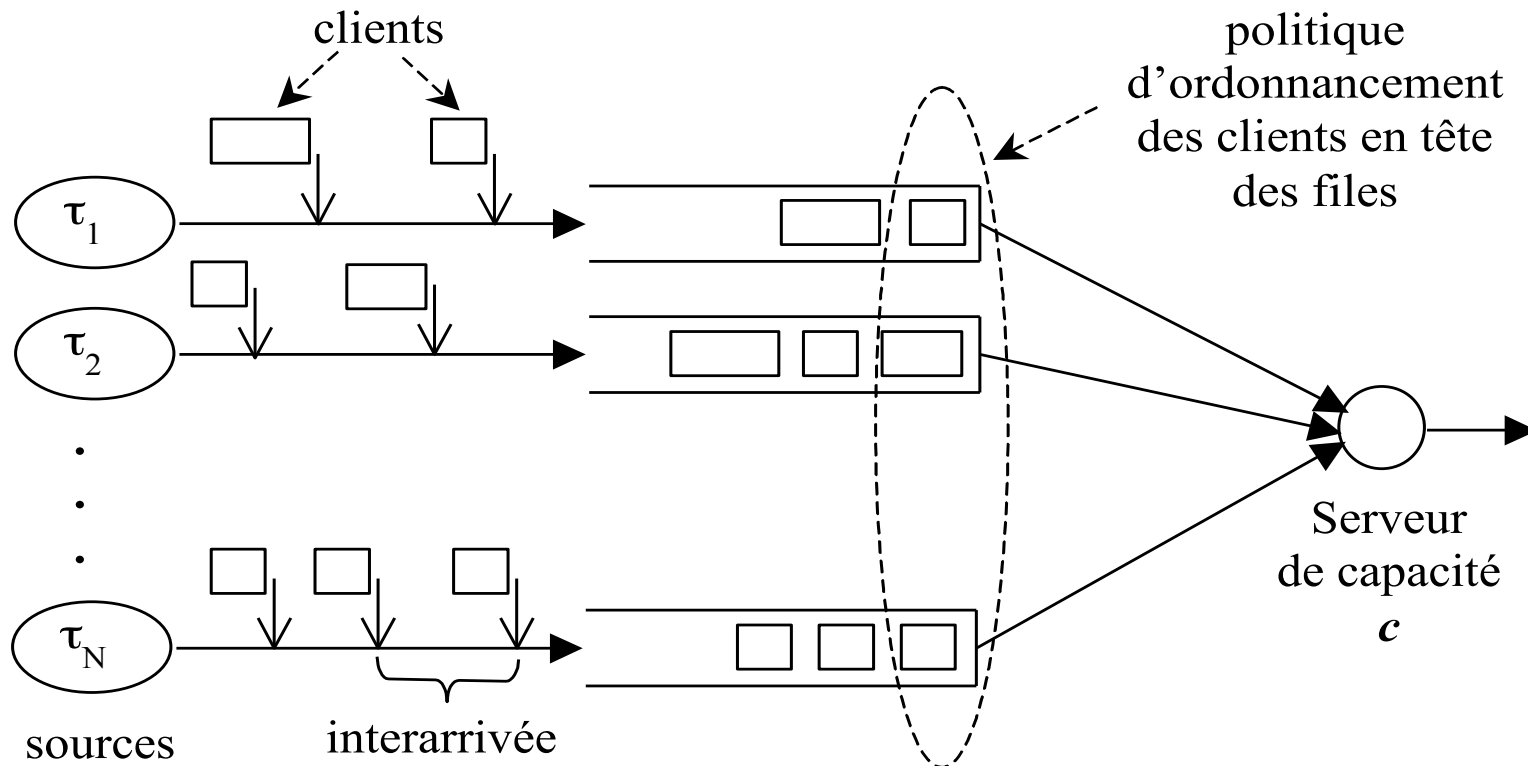
$$R_{i,k}(k_i) = I_i(k_i) - (k-1)T_i(k_i)$$

On arrête quand:  $R_{i,k}(k_i) \leq T_i(k_i)$

$$R_i(k_i) = \max_k \{R_{i,k}(k_i)\}$$

# Réflexions sur les approches d'évaluation de temps de réponse

Problème de partage d'une ressource commune, sans préemption, en respectant l'échéance → besoin d'évaluer le temps de réponse



# Flux d'arrivées et Techniques d'évaluation

---

## Hypothèses sur flux d'arrivée

- Communauté « ordonnancement », application connue
  - Tâches périodique ou sporadique:  $(C_i, T_i)$
  - Tâches périodique avec giges :  $(C_i, T_i, J_i)$
- Communauté « réseaux », application partiellement connue
  - Flux de paquets  $(\sigma_i, \rho_i)$ -borné
  - Flux de paquets aléatoire et durée de tx aléatoire:  $(\lambda, \mu)$   
avec  $\lambda = \text{moy}(1/T_i)$ ,  $\mu = \text{moy}(1/C_i)$

Techniques: -Probabiliste: *files d'attente*  
-Trajectoires majorantes :  
-*pire temps de réponse*  
-*network calculus*

# Pire temps de réponse sous l'ordonnancement non préemptif à priorité fixe

---

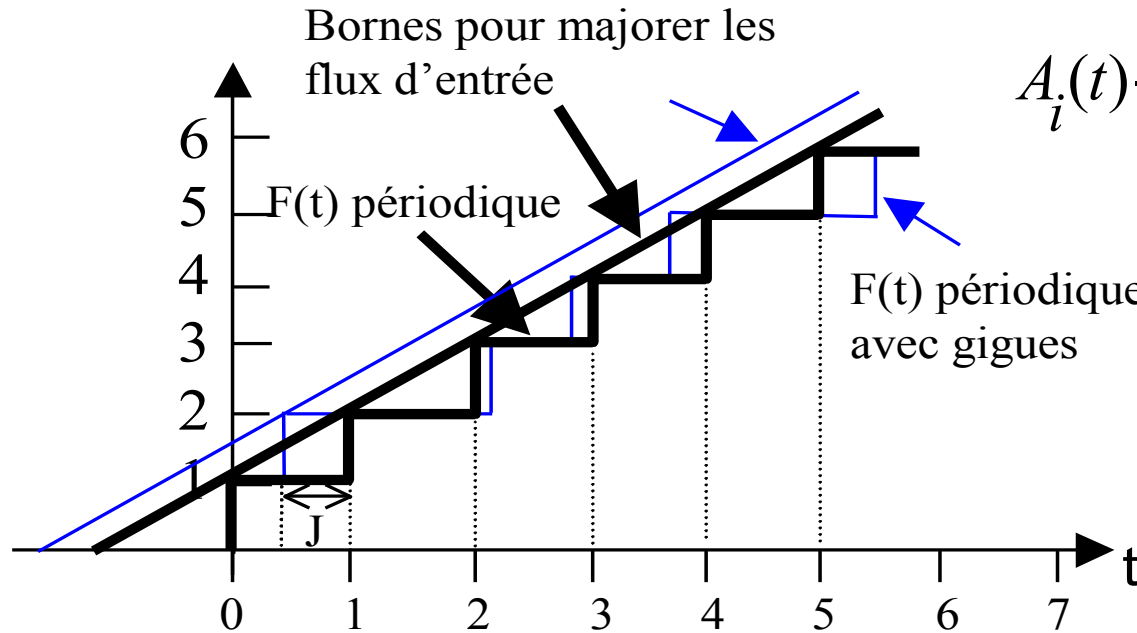
$$R_i = C_i + I_i + J_i \quad R_i \leq T_i$$

$$I_i^{n+1} = \max_{i+1 \leq j \leq N} (C_j) + \sum_{j=1}^{i-1} \left( \left\lceil \frac{I_i^n + J_j}{T_j} \right\rceil + 1 \right) C_j$$

Calcul récurrent :  $I_i^0 = 0$   $I_i^{n+1} = I_i^n$

Condition de convergence :  $\sum_{j=1}^i \frac{C_j}{T_j} \leq 1$

# Technique de « network calculus »



$$A_i(t) - A_i(s) \leq \sigma_i + \rho_i(t-s) \quad \forall 0 \leq s \leq t$$

$$D_{i_{\max}} = \frac{\sum_{j=1}^i \sigma_j + \max_{i+1 \leq j \leq N} (W_j)}{c - \sum_{j=1}^{i-1} \rho_j}$$

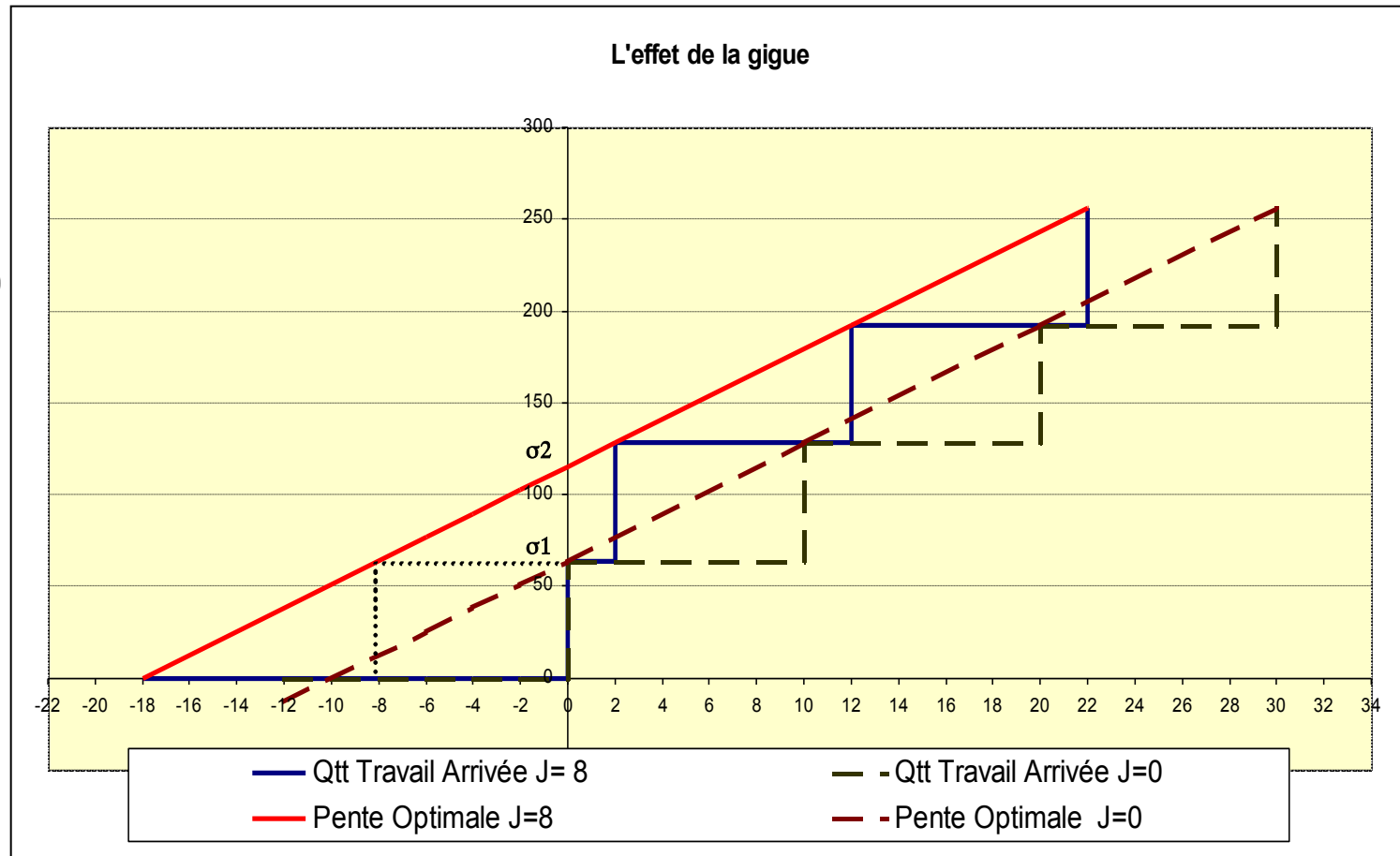
flux  $(\sum_{j=1}^{i-1} \sigma_j, \sum_{j=1}^{i-1} \rho_j)$ -borné

# Relation entre périodique pire cas et $(\sigma, \rho)$ -borné

$$\rho_i = \frac{W_i}{T_i}$$

$$\sigma_i = \frac{W_i}{T_i} (T_i + J_i)$$

$$C_i = W_i / c$$



# Temps de réponse probabiliste

---

- ◆ Hypothèse: demandes de ressource aléatoires
- ◆ Modèle: file d'attente M/G/1 avec priorité et sans préemption

- ◆ 
$$E[R_i] = E[q_i] + C_i = \frac{\lambda E[C^2]}{2(1 - a_{i-1})(1 - a_i)} + C_i$$

$$\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$$

$$a_i = \lambda_1 C_1 + \lambda_2 C_2 + \dots + \lambda_i C_i$$

$$E[C^2] = \frac{\lambda_1}{\lambda} E[C_1^2] + \frac{\lambda_2}{\lambda} E[C_2^2] + \dots + \frac{\lambda_n}{\lambda} E[C_n^2]$$

- ◆ Distribution des temps de réponse pour des cas particuliers (e.g., Poisson, Binomial)  $\rightarrow P[R_i > D_i] < \varepsilon$



# Quelle approche dans un réseau multi-saut?

Point clé: caractériser le flux de sortie:

- des résultats existent pour le flux  $(\sigma_i, \rho_i)$ -borné, mais problème de surdimensionnement [Grieu04], [Bouillard10]
- Pire trajectoire par flux sur tout le chemin, valable pour des flux statiques [Martin04], [Bauer10]
- flux de sortie d'une file M/G/1 n'est plus Markovien en général, rendant difficile l'analyse probabiliste

$$\hat{\sigma}_i = \sigma_i + \rho_i * \frac{\sum_{j=1}^{i-1} \sigma_j + \max_{i+1 \leq j \leq N} (W_j)}{c - \sum_{j=1}^{i-1} \rho_j}$$

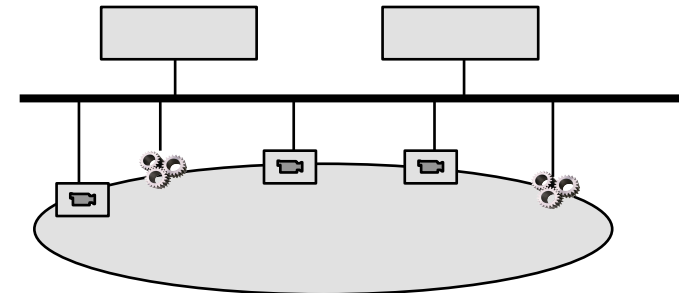


Evaluation de délais de bout en bout est un problème difficile, nécessite plus de recherches

# D'où vient l'échéance?

---

- ◆ Jusqu'à présent, on suppose un modèle  $(C_i, T_i, D_i)$  qui conduit à considérer le pire cas et à une allocation de ressource surdimensionnée. Mais est-il possible de relâcher la contrainte sur échéance afin d'éviter le surdimensionnement?
  - Applications "temps réel" dans l'Internet (voix, vidéo), le modèle (m,k)-firm peut convenir
  - Les boucles de contrôle-commande fermées sont souvent robustes, dont la fréquence d'échantillonnage est sur-estimée (4 à 10 fois)
- ◆ Possible d'exploiter un espace de solutions plus large pour des applications temps réel "souple" ou "firm"



# Modèle $(m,k)$ -firm

---

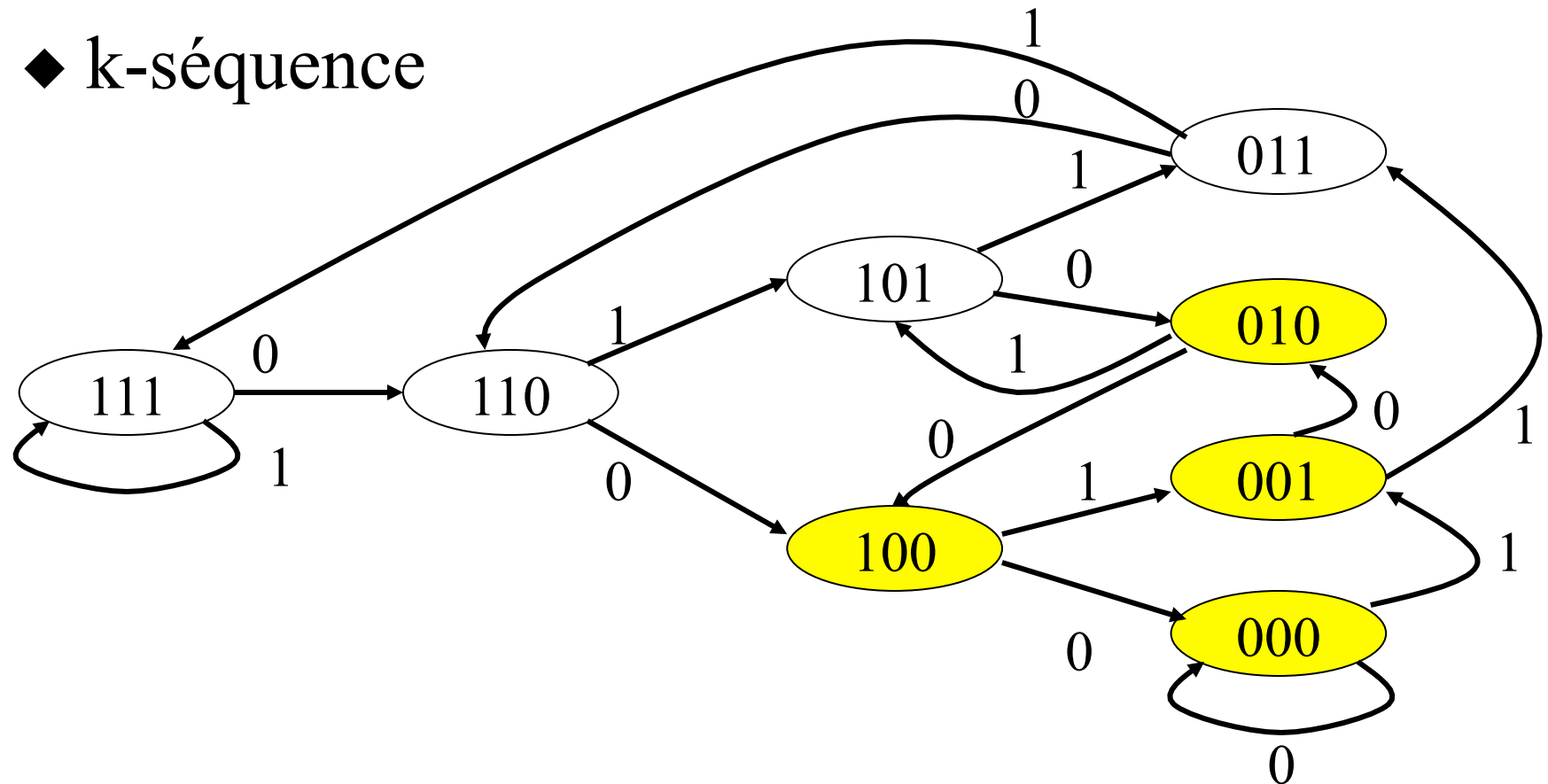
- ◆ *Temps réel dur*: non respect d'une échéance entraîne des conséquences catastrophiques
- ◆ *Temps réel souple*: non respect des échéances entraîne une diminution de performances (QoS dégradée)
  - *Temps réel « firm »*: temps réel souple mais avec le **non traitement des paquets** ne pouvant pas respecter leur échéances (paquets rejetés)
  - $(m,k)$ -firm: respect des échéances d'au moins  $m$  parmi  $k$  paquets consécutifs quelconques [Hamdaoui95]

# (m,k)-firm et états du system

---

◆ Exemple de (2,3)-firm

◆ k-séquence

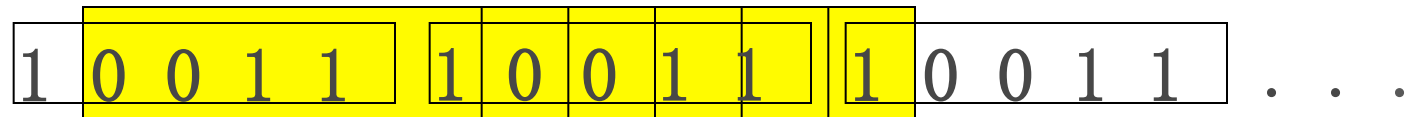


# k-séquence et expression de contraintes

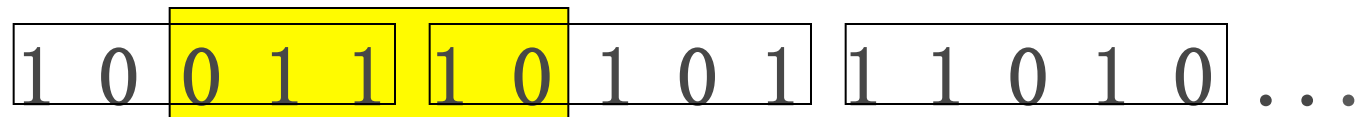
---

(3,5)-firm

- k-séquence fixe = k-pattern



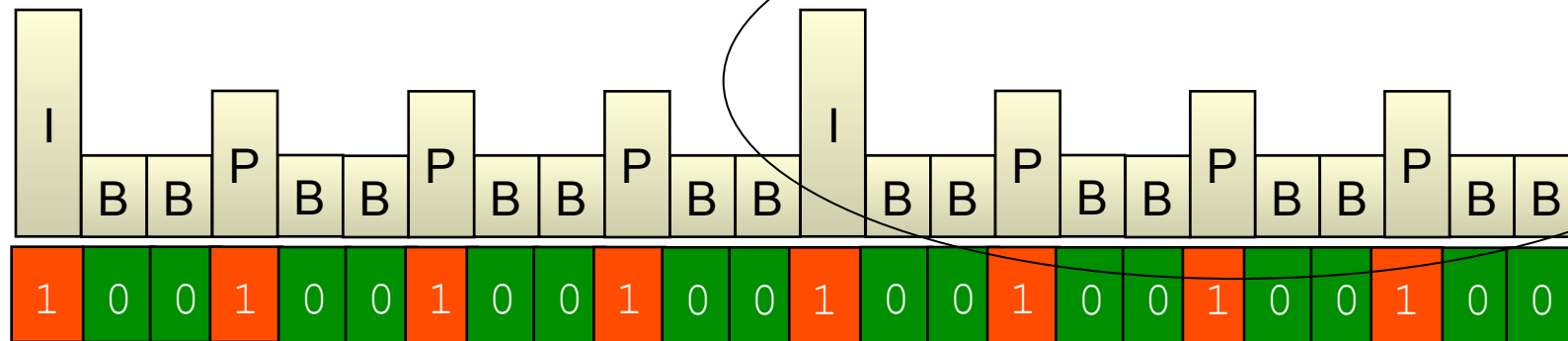
- k-séquence dynamique



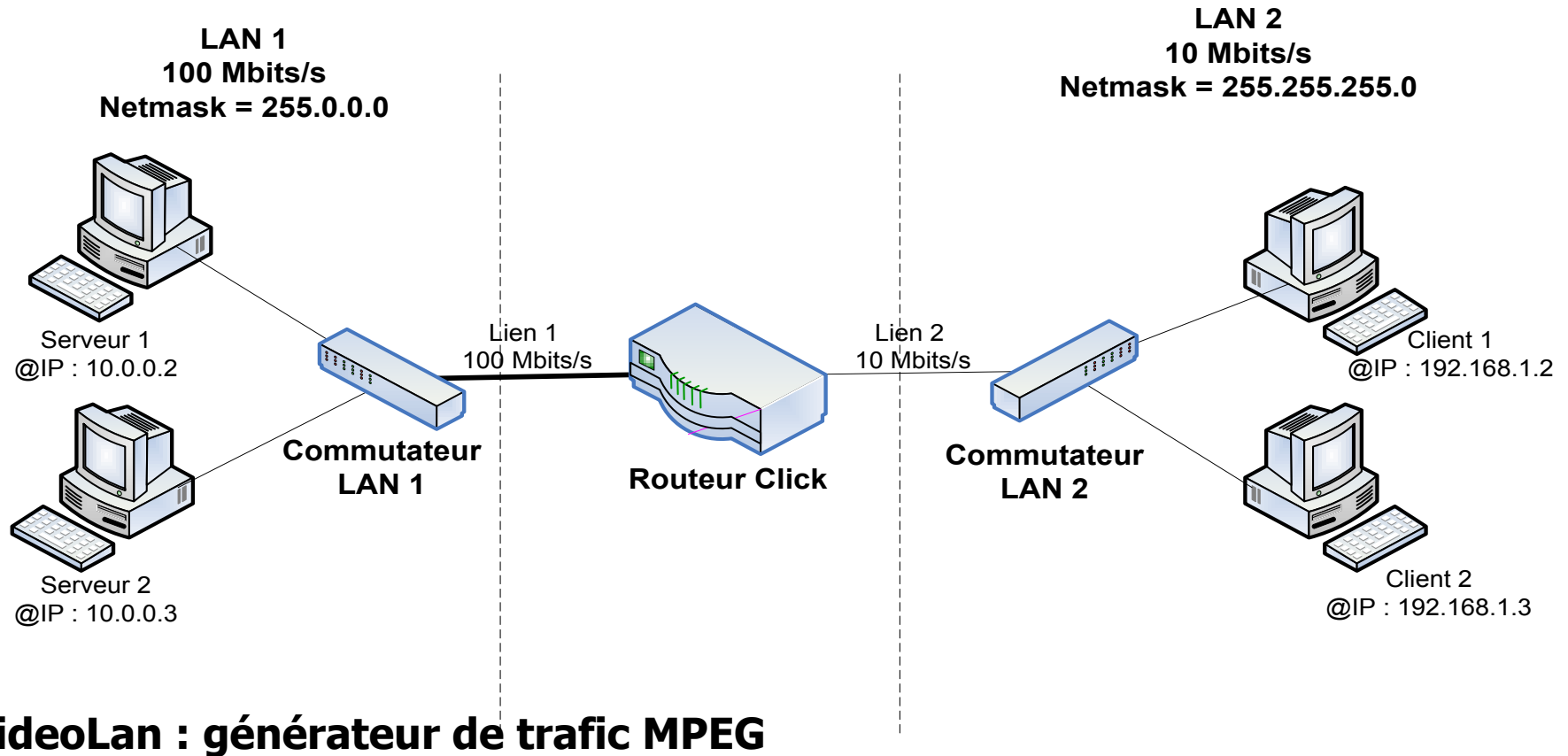
# Exemple d'une application acceptant la contrainte $(m,k)$ -firm

---

## ◆ Flux vidéo MPEG



# Maquette de tests



# Test sur maquette

---

Vidéo initiale:





# Test sur maquette

---

Test 1: rejet de tous les paquets de type I



Image fixe



# Test sur maquette

Test 2: rejet de tous les paquets de type P



# Test sur maquette

Test 3: rejet de tous les paquets de type B

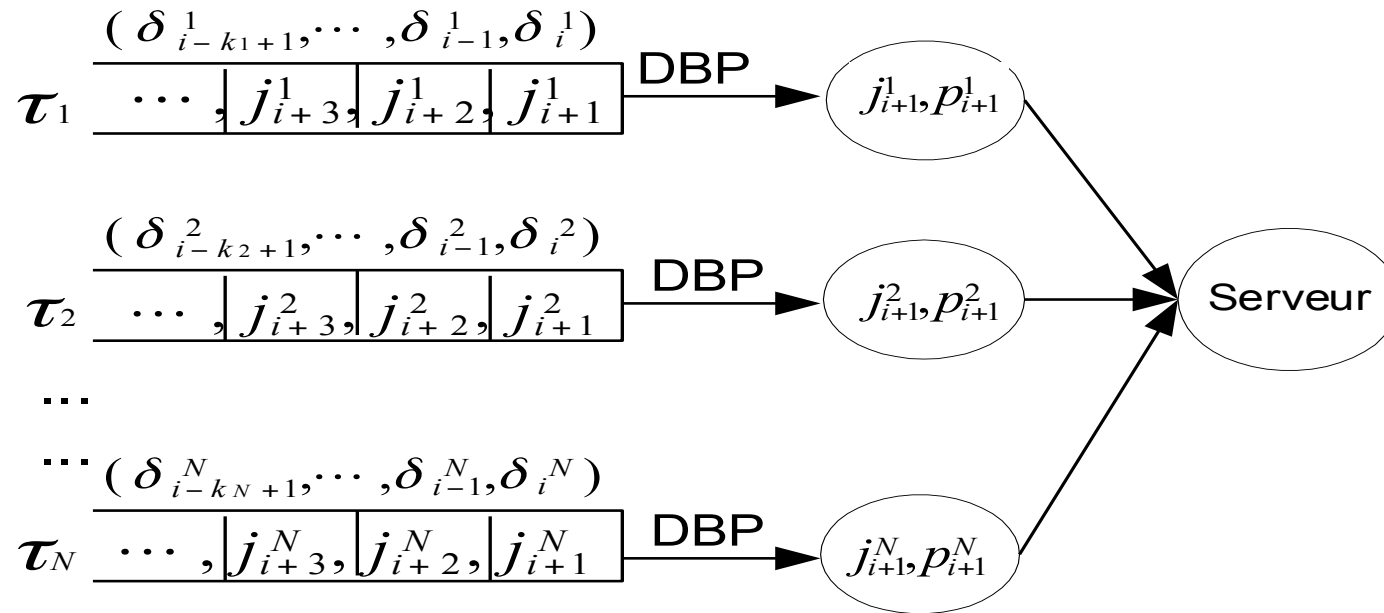


# DBP (Distance based Priorité)

---

- ◆ DBP d'une *k-séquence* est définie comme le nombre des échéances non respectées consécutives (nombre de zéros) conduisant à un état d'échec
- ◆ Exemple de (3,5)-firm
  - (11011) a DBP = 2 car 11 (01100)
  - (10111) a DBP = 3 car 101 (11000)
  - (10001) a DBP = 0
- ◆ L'affectation de priorité dans DBP est dynamique (qui dépend de k-séquence)

# DBP dans notre modèle



$j_i^x$ : ième client de source x     $p_i^x$ : priorité de ième client de source

- ◆ Une source est caractérisée par:  $\tau_i = \{c_i, p_i, D_i, m_i, k_i\}$
- ◆ DBP peut être implémenté matériellement par un registre
- ◆ En cas d'égalité de priorité DBP, EDF par défaut

# Ordonnancement NP-DBP-EDF

---

- ◆ Priorité 0: le paquet en cours de service (transmission) a la plus haute priorité (*non préemption*)
- ◆ Priorité 1: paquets avec DBP=**1** en attente de transmission (*doivent être garantie, sinon violation de la contrainte*)
- ◆ Priorité i: paquets avec DBP=**i** ( $i > 1$ ) en attente de transmission (*transmis s'ils peuvent l'être avant leur échéance, écartés sinon*)
- ◆ Paquets avec la même valeur de DBP sont ordonnancés selon EDF

# Garantie déterministe de (m,k)-firm

---

Garantie déterministe de (k,k)-firm donnée par condition suffisante de [Jeffay91]

Charge:  $\sum_{i=1}^N \left[ \frac{c_i}{p_i} \frac{m_i}{k_i} \right] \leq 1$  au lieu de  $\sum_{i=1}^N \left[ \frac{c_i}{p_i} \right] \leq 1$

Question: (m,k)-firm permet-elle de relaxer la demande de ressources par rapport à (k,k)-firm?



Analyse de l'ordonnançabilité sous contrainte (m,k)-firm

# DBP: condition suffisante (1)

---

- ◆ Pour trouver la condition suffisante d'ordonnançabilité, le point clé est de trouver la **période pire cas** et sa **charge (workload)** correspondante
- ◆ Le pire cas a lieu pendant la **DBP=1 busy period** où seuls les paquets avec DBP=1 sont transmis.



# DBP: condition suffisante (2)

---

*Supposons l'existence d'une telle période qui commence à l'instant  $t_0$  et se termine à  $t_d$ .*

*Trois cas sont identifiés:*

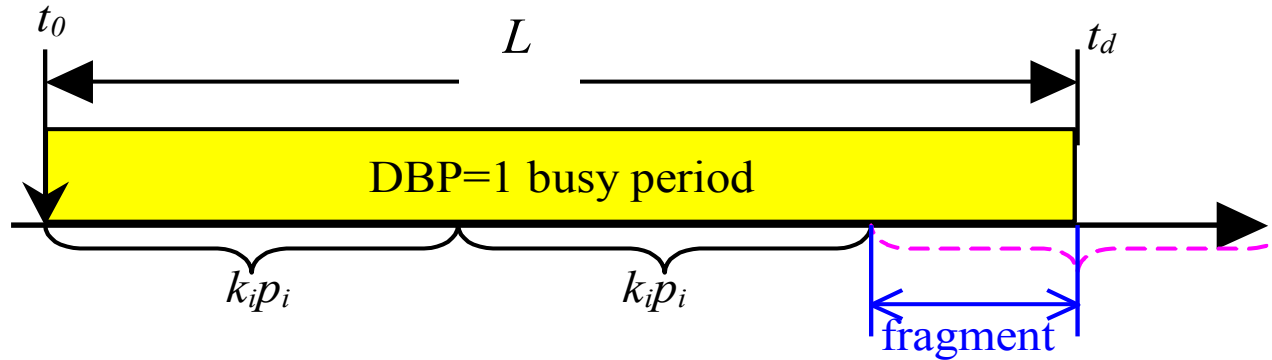
- 1) DBP=1 busy period commence quand le serveur est libre à  $t_0$ , et tous les paquets ont leur échéance avant  $t_d$ .
- 2) DBP=1 busy period commence mais le serveur est bloqué par un paquet de DBP>1 à  $t_0$ , et tous les paquets ont leur échéance avant  $t_d$ .
- 3) Il y a quelques paquets dont l'échéance est après  $t_d$ .

# Workload dans le cas 1

---

- ◆  $L = t_d - t_0$  : longueur de la DBP=1 busy period
- ◆ Paquets classés en deux sous-ensembles :
  - $U$  : paquets avec DBP=1 à  $t_0$
  - $a()-U$  : paquets avec DBP >1 à  $t_0$  mais devenus DBP=1 à  $t_x < t_d$
- ◆ Tous les paquets doivent être transmis car leur échéance est avant  $t_d$ , la **workload** est définie comme le **temps nécessaire pour transmettre tous ces paquets**

# Cas 1: Workload du sous-ensemble $U$

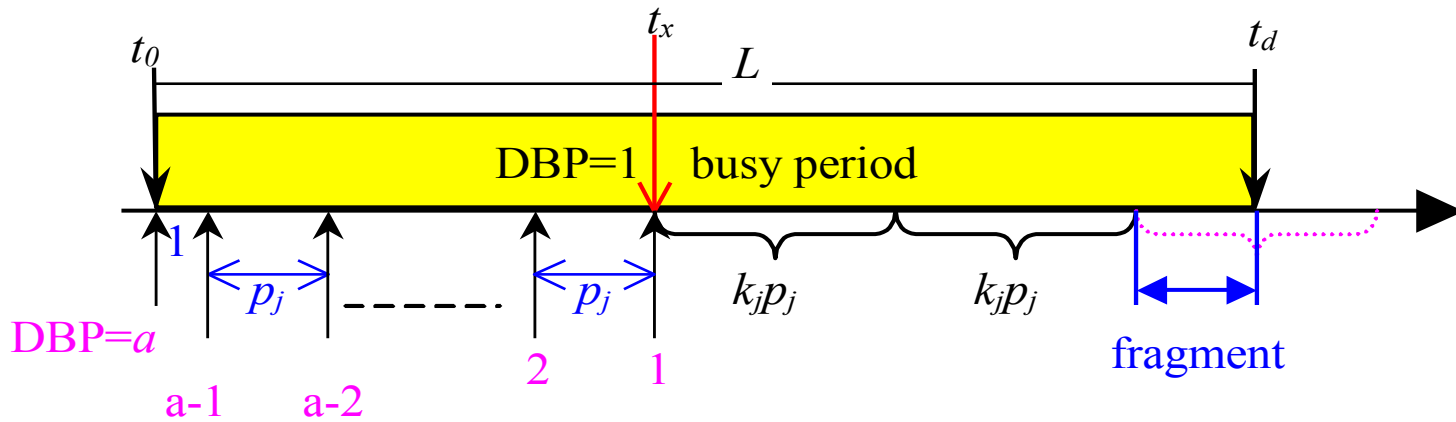


Workload durant la part entière :  $\left\lfloor \frac{L}{k_i p_i} \right\rfloor$   $W_U^1 = \sum_{i \in U} \left( \left\lfloor \frac{L}{k_i p_i} \right\rfloor m_i \right) c_i$

Workload durant la partie du fragment:

$$L - k_i p_i \left\lfloor \frac{L}{k_i p_i} \right\rfloor \quad W_U^2 = \sum_{i \in U} \text{Min} \left( \left\lfloor \frac{L - k_i p_i \left\lfloor \frac{L}{k_i p_i} \right\rfloor}{p_i} \right\rfloor, m_i \right) c_i$$

# CAS 1: Workload du sous-ensemble $a() - U$



Dans le pire cas :  $l = t_x - t_0 = l + (DBP_j(t_0) - 2)p_j$

Workload du sous-ensemble  $a() - U$  :

durant  $\left\lfloor \frac{L-l}{k_i p_i} \right\rfloor$

$$W_{a() - U}^1 = \sum_{j \in a() - U} \left( \left\lfloor \frac{L-l}{k_j p_j} \right\rfloor m_j \right) c_j$$

durant  $(L-l) - k_i p_i \left\lfloor \frac{L-l}{k_i p_i} \right\rfloor$

$$W_{a() - U}^2 = \sum_{j \in a() - U} \text{Min} \left( \left\lfloor \frac{(L-l) - k_j p_j \left\lfloor \frac{L-l}{k_j p_j} \right\rfloor}{p_j} \right\rfloor, m_j \right) c_j$$

# Cas 1: Workload totale

---

Somme de la workload des deux sous-ensembles :

$$\sum_{i \in U} \left( \left\lfloor \frac{L}{k_i p_i} \right\rfloor m_i + \text{Min} \left( \left\lfloor \frac{L - k_i p_i \left\lfloor \frac{L}{k_i p_i} \right\rfloor}{p_i} \right\rfloor, m_i \right) \right) c_i +$$

$$\sum_{j \in \alpha(0) - U} \left( \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor m_j + \text{Min} \left( \left\lfloor \frac{(L - 1 - (DBP_j(t) - 1)p_j) - k_j p_j \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor}{p_j} \right\rfloor, m_j \right) \right) c_j$$

L'analyse similaire est appliquée aux cas 2 et cas 3

# DBP: condition suffisante (3)

---

**Théorème:** [Li06]

Pour un ensemble de paquets périodiques ou sporadiques  $a()$ ,  $a() = \{a(1), a(2), \dots, a(n)\}$ ,

$$a(i) = \{c_i, p_i, m_i, k_i\}, d_i = p_i$$

Si  $a()$  satisfait les conditions C1 et C2,

NP-DBP-EDF sera capable d'ordonnancer un ensemble periodique ou sporadique concret quelconque généré à partir de  $a()$ . *i.e. les contraintes  $(m_i, k_i)$ -firm sont garanties.*

# DBP: condition suffisante (4)

---

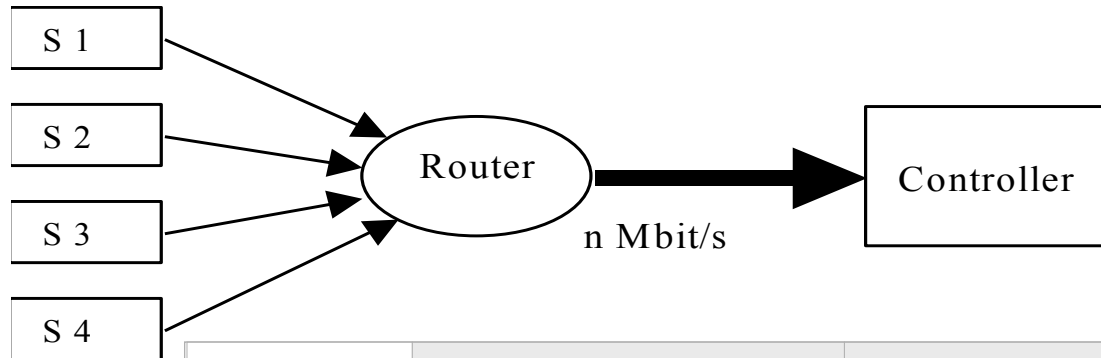
$$\begin{aligned}
 \text{C1: } & \sum_{j \in \mathcal{U}} \left( \left\lfloor \frac{L}{k_i p_i} \right\rfloor m_i + \text{Min} \left( \left\lfloor \frac{L - k_i p_i \left\lfloor \frac{L}{k_i p_i} \right\rfloor}{p_i} \right\rfloor, m_i \right) \right) c_i + \\
 & \sum_{j \in \mathcal{U}} \left( \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor m_j + \text{Min} \left( \left\lfloor \frac{(L - 1 - (DBP_j(t) - 1)p_j) - k_j p_j \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor}{p_j} \right\rfloor, m_j \right) \right) c_j \leq L
 \end{aligned}$$

$$\text{C2: } \forall i, \forall L, L > \min(p_i)$$

$$\begin{aligned}
 & \left( \left( \left\lfloor \frac{L - c_i}{k_i p_i} \right\rfloor^+ m_i + 1 \right)^+ + \text{Min} \left( \left\lfloor \frac{L - c_i - \left\lfloor \frac{L - c_i}{k_i p_i} \right\rfloor^+ k_i p_i}{p_i} \right\rfloor, m_i - 1 \right) \right)^+ c_i + \\
 & \sum_{j \in \mathcal{U} - a(i)} \left( \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor m_j + \text{Min} \left( \left\lfloor \frac{(L - 1 - (DBP_j(t) - 2)p_j) - k_j p_j \left\lfloor \frac{L - 1 - (DBP_j(t) - 2)p_j}{k_j p_j} \right\rfloor}{p_j} \right\rfloor, m_j \right) \right) c_j \leq L
 \end{aligned}$$

# Exemple numérique

---

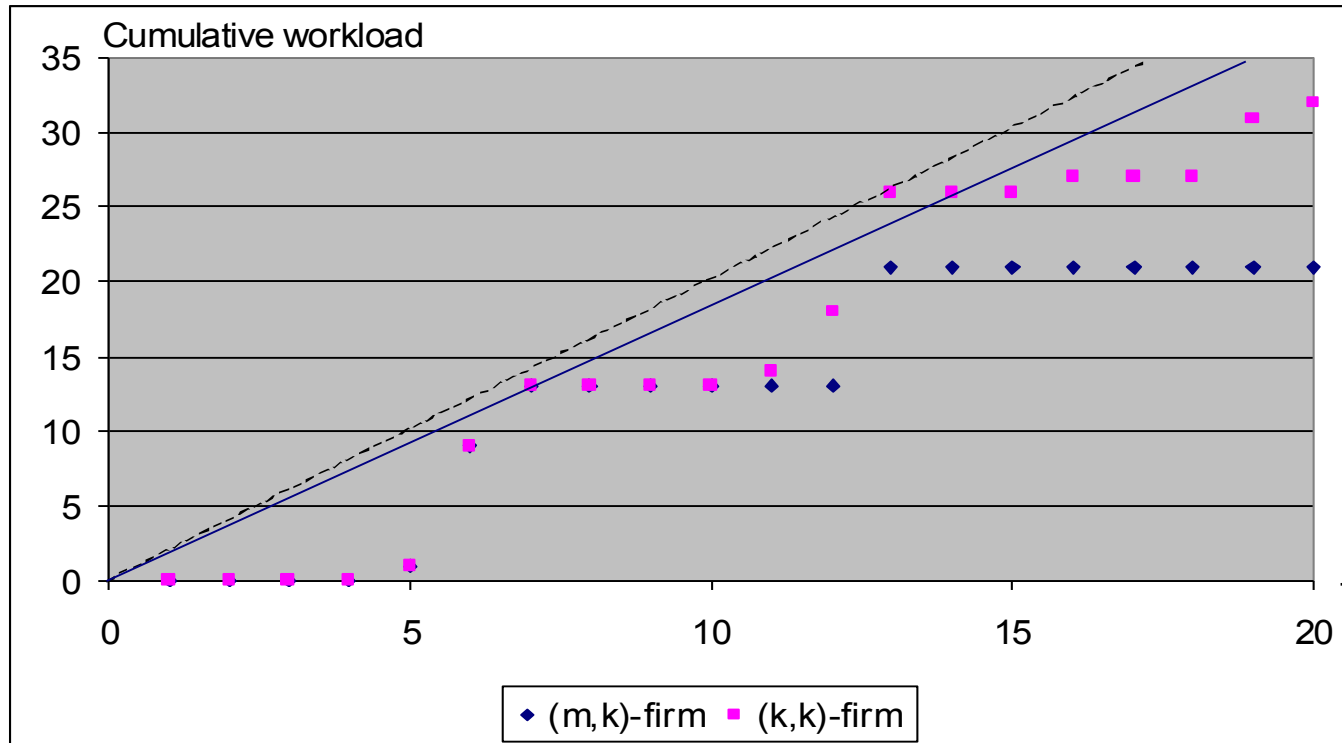


	<b>Packet size (kbit)</b>	<b>Interarrival time/ deadline (ms)</b>	<b>(m,k)-firm constraint</b>
<b>S0</b>	8	12	(2,5)
<b>S1</b>	8	20	(4,5)
<b>S2</b>	1	5	(1,4)
<b>S3</b>	4	6	(1,5)



# Exemple numérique

## ◆ $(m,k)$ -firm vs. $(k,k)$ -firm (HRT)



# Conclusion

---

- ◆ L'ordonnancement de messages/paquets dans les réseaux est nécessaire pour la QoS (e.g. Priorité, WRR)
- ◆ Messages/paquets non préemptifs
- ◆ L'approche de l'analyse du pire temps de réponse offre une borne intéressante par rapport à la borne  $(\sigma, \rho)$ , mais peut conduire au problème de surdimensionnement
- ◆ Besoin d'aller au-delà de simple échéance sur paquet et s'intéresser à l'application (e.g. boucle de contrôle, capacité de tolérance applicative)
  - $(m,k)$ -firm est un exemple de remède au problème de surdimensionnement

# Perspectives

---

- ◆ L'histoire ne s'arrête pas à l'âge de 40 ans
- ◆ L'évaluation de délais dans des réseaux multi-sauts soulève des problèmes difficiles
- ◆ Coupler l'ordonnancement en-ligne et la qualité de l'application, e.g. qualité du contrôle dans des NCS (Networked Control Systems) → vers des systèmes auto-adaptatifs [Aubrun10]
- ◆ CPS (Cyber-Physical Systems) et réseaux sans fil: ordonnancement en-ligne sous contraintes de perte de paquets et variation de débit

# Bibliographie

---

- ◆ [Thomesse05] Jean-Pierre Thomesse, Fieldbus technology in industrial automation, *Proceedings of the IEEE 93*, 6 (2005) 1073-1101.
- ◆ [Tindell94] Tindell, K., A. Burns, "Guaranteeing message latencies on controller area network (CAN)", *1st international CAN conference (iCC'94)*, Germany, sept. 1994.
- ◆ [Simonot-Lion95] Simonot-Lion F., Song Y.Q., Berthomieu B., Vernadat F., « Vérification des applications temps réel », Dans *Encyclopédie de l'informatique et des systèmes d'information*, Vuibert (ed.) (2005).
- ◆ [Navet00] N. Navet, Y.Q. Song, F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over CAN (controller area network)", *Journal of systems architecture - the EUROMICRO Journal*, 46 (2000) pp607-617.
- ◆ [Song02] Y.Q. Song, A. Koubâa, F. Simonot, "Switched Ethernet For Real-Time Industrial Communication: Modelling And Message Buffering Delay Evaluation", *WFCS'2002*, Vasteras (Sweden), pp27-35, August 28-30, 2002.
- ◆ [Koubaa04] A. Koubâa, Y.Q. Song, "Evaluation and improvement of response time bounds for real-time applications under non pre-emptive fixed priority scheduling", *International Journal of Production Research*, Vol.42, No.14, pp2899-2913, Taylor & Francis group, July 2004.

# Bibliographie

---

- ◆ [Bouillard10] Anne Bouillard, Laurent Jouhet, and Eric Thierry. Tight performance bounds in the worst-case analysis of feed-forward networks. *INFOCOM'10*, San Diego, USA, pages 1316–1324, March 2010.
- ◆ [Grieu04] Jérôme Grieu. Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avioniques. PhD thesis, INPT Toulouse, France, September 2004.
- ◆ [Martin04] Steven Martin. Maîtrise de la dimension temporelle de la qualité de service dans les réseaux. PhD thesis, Université Paris 12, 2004.
- ◆ [Bauer10] Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul. Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach. *IEEE Transactions on Industrial Informatics*, 6(4) :521–533, November 2010.
- ◆ [Hamdaoui95] M. Hamdaoui and P. Ramanathan, “A dynamic priority assignment technique for streams with  $(m, k)$ -firm deadlines”, *IEEE Transactions on Computers*, 44(4), 1443–1451, Dec. 1995.
- ◆ [Li06] Li J., Song Y.Q., Simonot-Lion F., “Providing Real-time Applications with Graceful Degradation of QoS and Fault Tolerance According to  $(m,k)$ -firm Model”, *IEEE Transactions on Industrial Informatics* 2, 2 (2006) 112-119
- ◆ [Aubrun10] Aubrun C., Simon D., and Song Y.Q. (eds), *Co-design Approaches for Dependable Networked Control Systems*, ISTE and J. Wiley, 2010, 314 pages.