Ordonnancement temps réel (tâches apériodiques)

Maria ZRIKEM

Ensa de Marrakech

Inclusion des tâches apériodiques

⇒ **les tâches à contraintes d'échéances dures ont été modélisées sous formes de tâches périodiques afin de déterminer des tests de faisabilité d'ordonnancement.**

⇒ **il faut intégrer les tâches moins critiques tout en leur permettant de bénéficier d'un traitement non trop pénalisant sans mettre en cause l'ordonnancement. Ces tâches sont assimilées à des processus apériodiques.**

⇒ **on ne peut pas les considérer comme les moins prioritaires.**

Inclusion des tâches apériodiques

- ⇒ **Se distinguent des tâches périodiques par le fait que l'on ne connaît pas l'instant d'arrivée de la requête de réveil**
- ⇒ **Contraintes temporelles strictes ou relatives**
- ⇒ **Buts à atteindre:**
  - ⇒ **si contraintes relatives : minimiser le temps de réponse**
  - ⇒ **si contraintes strictes : maximiser le nombre de tâches acceptées en respectant leurs contrainte**

Inclusion des tâches apériodiques

Solutions

1. **Traitement d'arrière plan**
2. **Une tâche périodique spéciale qui contient elle-même un ordonnanceur de tâches apériodiques (serveur). La capacité et la priorité de cette tâches pouvant être réglée au mieux en garantissant l'ordonnançabilite des autres. Souvent on lui donne la priorité la plus forte.**

Tâches apériodiques à contraintes relatives

Traitement en arrière plan BG (background processing)

⇒ **Simple à réaliser et peu couteuse.**

⇒ **Les taches apériodiques avec une priorité toujours inférieure à celles des taches périodiques.**

⇒ **Les tâches apériodiques sont ordonnancées lorsqu'il n'y a aucune tâche périodique prête à s'exécuter. Si plusieurs tâches apériodiques sont en attente, elles sont traitées selon leurs ordre de réveil (FIFO).**

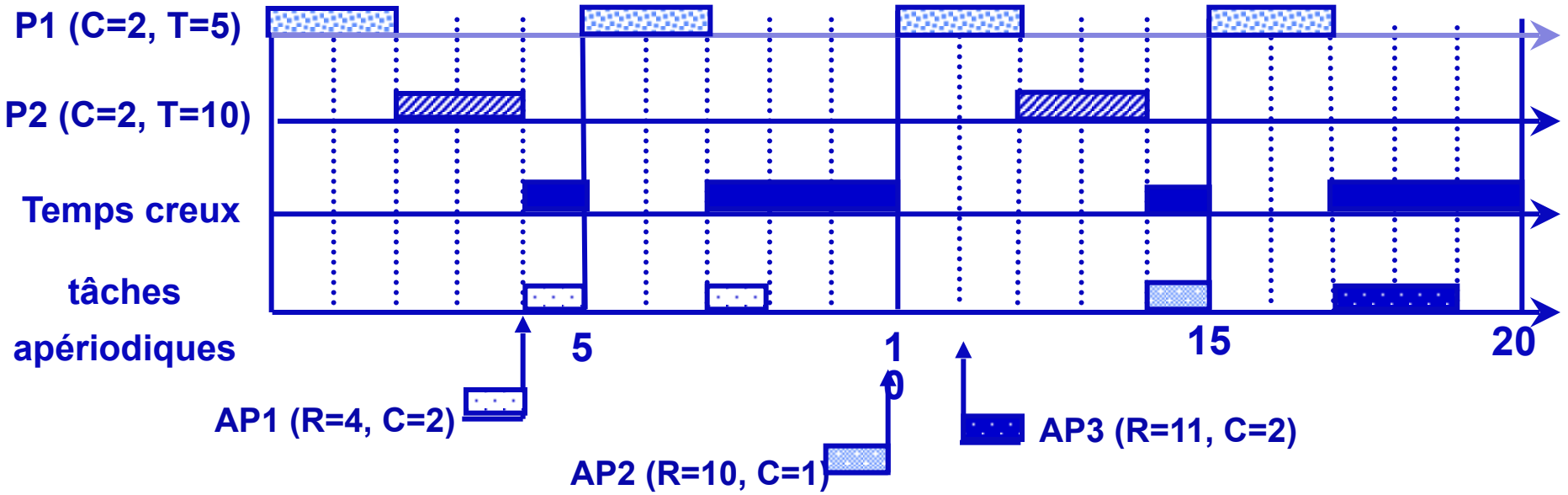
⇒ **Permet de garantir l'exécution des tâches périodiques dans le respect de leurs contraintes temporelles**

⇒ **Le temps de réponse des tâches apériodiques peuvent s'avérer très grands surtout quand la charge des tâches périodiques devient importante et s'approche de 1**

Traitement en arrière plan BG (background processing)

Exemple :

- 2 tâches périodiques P1 (C=2, T=5), P2 (C=2, T=10)
- 3 tâches apériodiques AP1 (R=4, C=2), AP2(R=10, C=1), AP3(R=11, C=2)



Les serveurs de tâches

Minimiser le temps de réponse des tâches apériodiques dans le contexte d'un ordonnancement conduit par priorités.

Faire prendre en charge par une tâche particulière qui possède une réelle priorité (statique ou dynamique) sans toutefois compromettre l'exécution des tâches périodiques : tâche serveur.

Serveur de scrutation (polling server)

⇒ C'est une tâche qui a généralement la plus forte priorité

⇒ A chacune de ses activations, le serveur de scrutation traite les T. APs. en suspens depuis son activation précédente et ce, soit jusqu'à épuisement de sa capacité, soit jusqu'à ce qu'il n' y a aucune T. AP à servir.

⇒ Le PS sert périodiquement les requêtes apériodiques en attente selon la technique FIFO.

⇒ Si au commencement de son exécution, il n' y a aucune T. AP en attente, le serveur se suspend jusqu'à sa prochaine occurrence en donnant la main au T. Ps.

Serveur de scrutation (polling server)

⇒ **Condition suffisante de RMA avec un serveur :**

$$\sum_{i=1}^{n} \frac{C_i}{T_i} + \frac{C_{serv}}{T_{serv}} \leq (n+1)(2^{\frac{1}{n+1}} - 1)$$

⇒ **Simple à mettre en oeuvre et test d'ordonnançabilité facilement adaptables**

⇒ **Le serveur relâche sa capacité C_{serv} si aucun événement déclencheur de tâche apériodique n'est arrivé durant la période de scrutation (temps de réponse aux tâches apériodiques pas très bon).**

Serveur de scrutation (polling server)

Exemple (ordonnancement par RM):

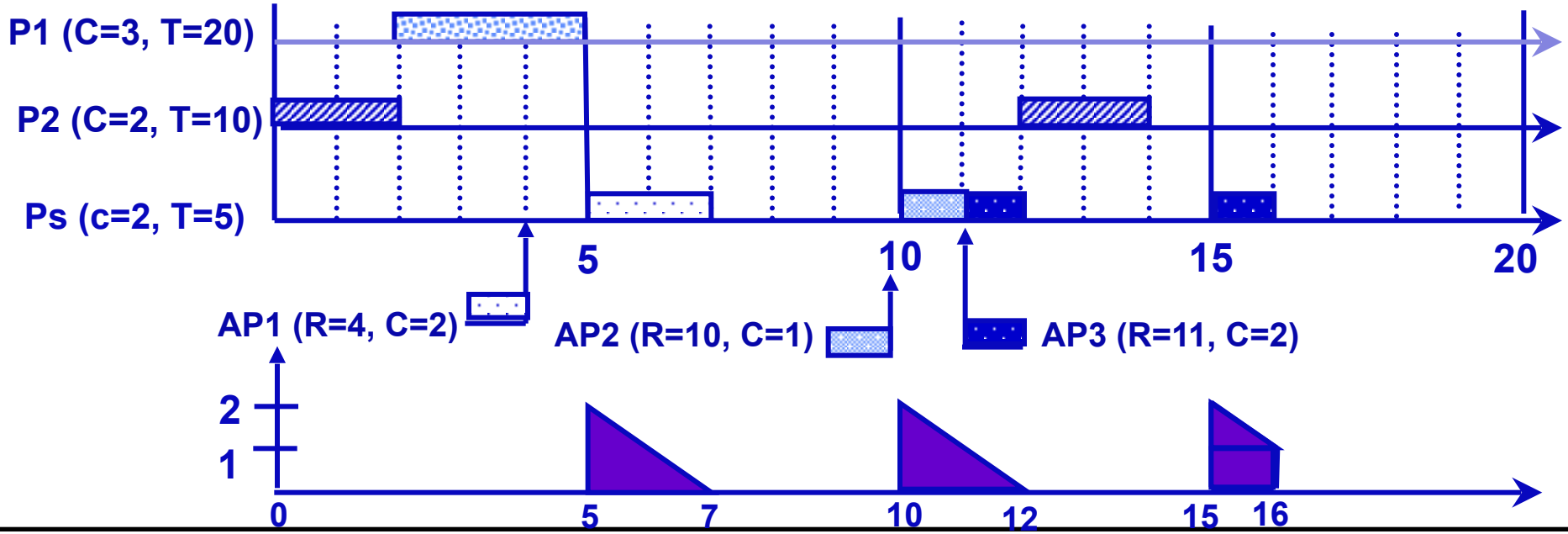
- **2 tâches périodiques P1 (C=3, T=20), P2 (C=2, T=10)**
- **la tâche serveur Ps (C=2, T=5) (de plus petite période et donc de plus forte priorité)**
- **3 tâches apériodiques AP1 (R=4, C=2), AP2(R=10, C=1), AP3(R=11, C=2)**

Serveur de scrutation (polling server)

Exemple (ordonnance par RM):

- 2 tâches périodiques P1 (C=3, T=20), P2 (C=2, T=10)
- la tâche serveur Ps (C=2, T=5) (de plus petite période et donc de plus forte priorité)
- 3 tâches apériodiques AP1 (R=4, C=2), AP2(R=10, C=1), AP3(R=11, C=2)

La condition de faisabilité (RM) est remplie :3/20 + 2/10 + 2/5 =0,75 < 0,77



Serveur de scrutation (polling server)

La méthode de scrutation souffre de plusieurs faiblesse dues à la manière dont est gérée sa capacité :

⇒ Lorsque le serveur est prêt, il se peut qu'il n'ait pas de T.APs à traiter et la capacité est alors perdue;

⇒ Lorsque des T. APs. surviennent sur le site, il se peut que le serveur se soit déjà suspendu, contraignant ainsi les T. APs. à attendre sa prochaine occurrence pour être servies.

Serveur différé (Defferable server DS)

⇒ **Cette méthode revient à abaisser le seuil du critère d'ordonnancement des tâches périodiques tout en fournissant un bon temps de réponse aux T. APs.**

⇒ **Le serveur différé est active quand il y a une T. AP. en attente. Le serveur est suspendu quand il a dépassé son quantum de temps. Son quantum est rempli quand il arrive en début de période.**

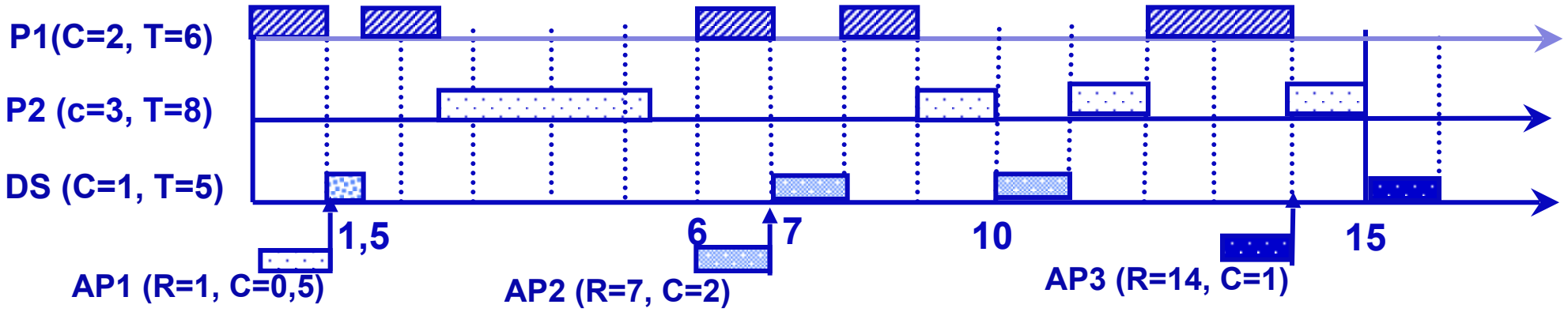
⇒ **En général, on affecte la plus haute priorité à DS (Tds < min (périodes des tâches périodiques)). Elle est définie suivant la méthode RM.**

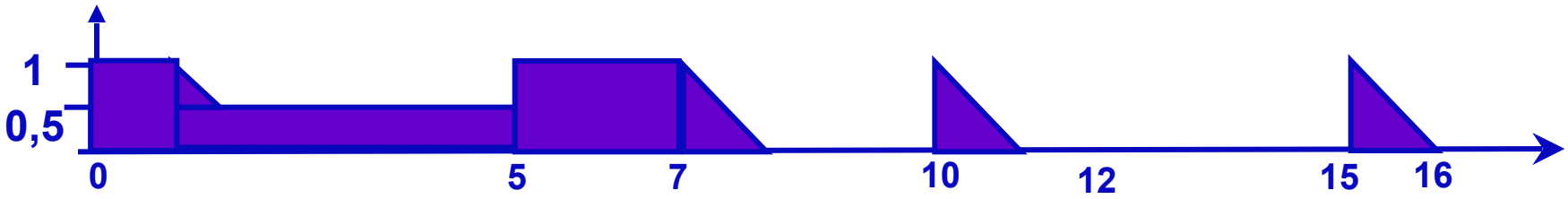
Serveur différé : exemple

- 2 tâches périodiques P1 (C=2, T=6), P2 (C=3, T=8)
- la tâche serveur DS (C=1, T=5) (de plus petite période et donc de plus forte priorité)
- 3 tâches apériodiques AP1 (R=1, C=0,5), AP2(R=7, C=2), AP3(R=14, C=1)

Serveur différé : exemple

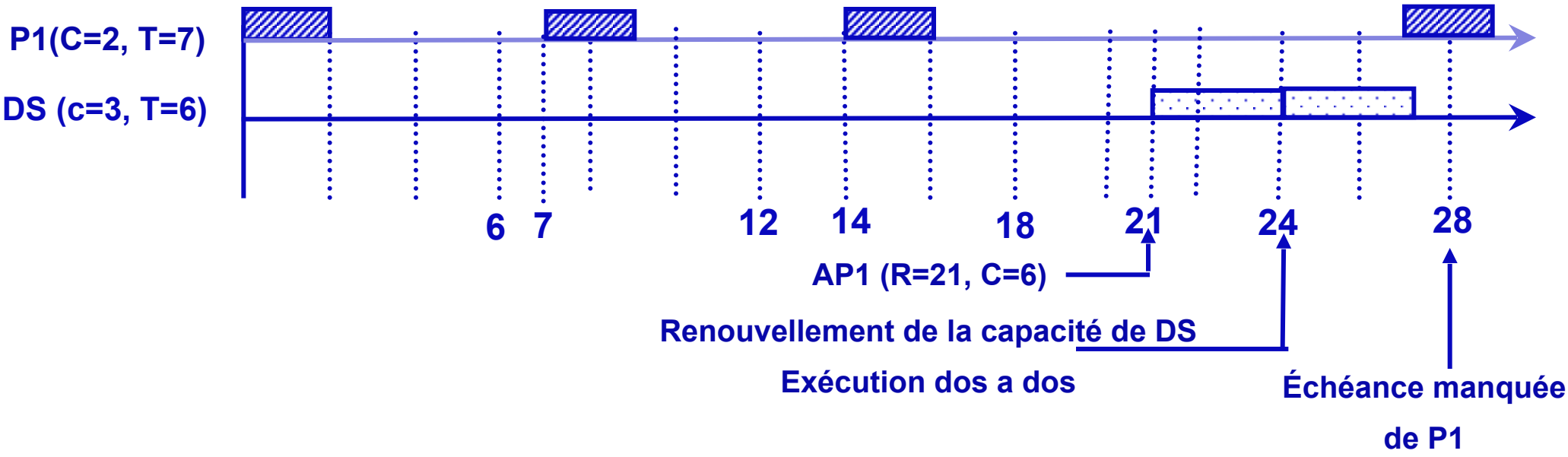
- 2 tâches périodiques P1 (C=2, T=6), P2 (C=3, T=8)
- la tâche serveur DS (C=1, T=5) (de plus petite période et donc de plus forte priorité)
- 3 tâches apériodiques AP1 (R=1, C=0,5), AP2(R=7, C=2), AP3(R=14, C=1)





Serveur différé : comportement indésirable

- 1 tâche périodique P1 (C=2, T=7),
- la tâche serveur DS (C=3, T=6)
- 1 tâche apériodique AP1 (R=21, C=6)



Serveur sporadique (Sporadic server SS)

⇒ **Comme le DS, le serveur sporadique est défini par sa priorité d'exécution (fixée par exemple à l'inverse de sa période dans le RM), et sa capacité.**

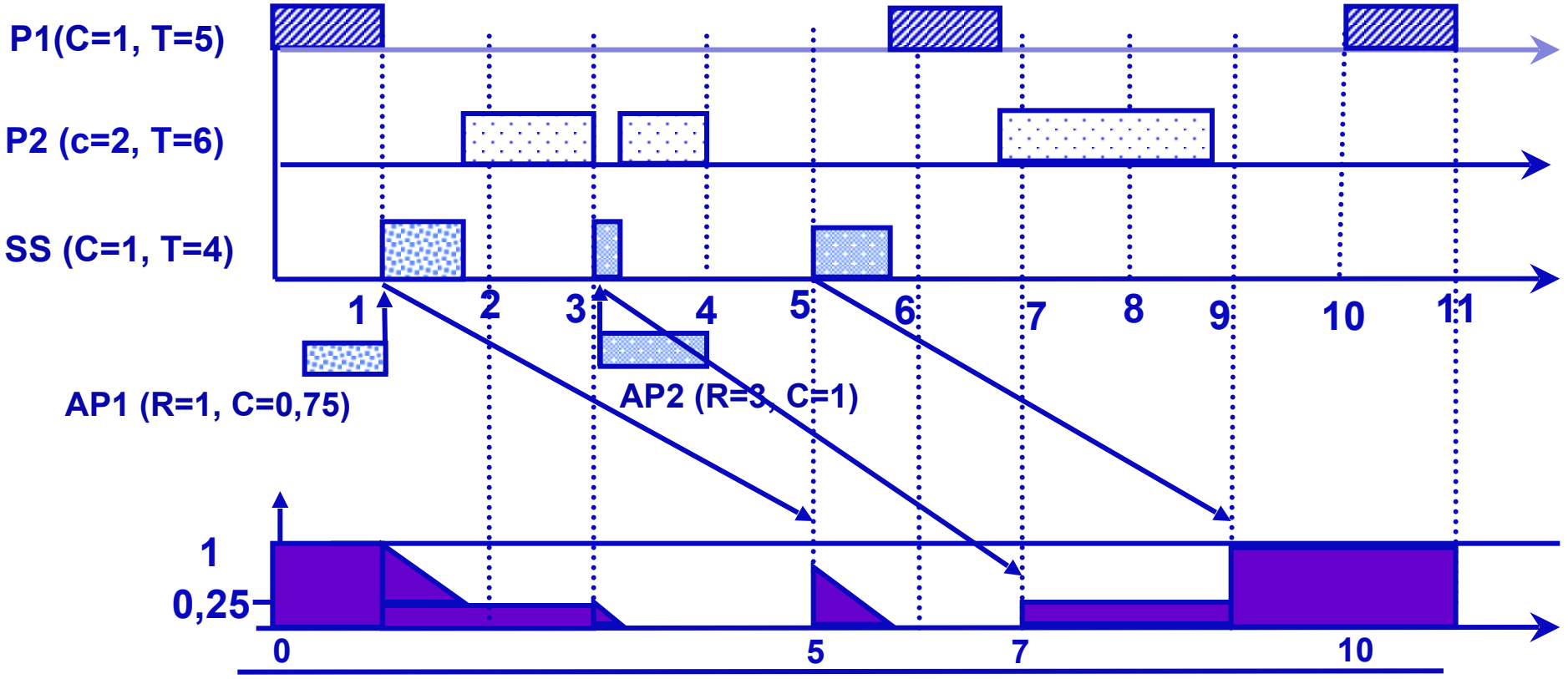
⇒ **Contrairement au DS, le SS ne retrouve pas sa capacité originelle à chaque frontière de période mais attend un instant de réapprovisionnement qui se ramène au temps de son dernier démarrage avec une capacité pleine + sa période. lL n'y a plus d'exécution dos à dos possible**

⇒ **La capacité d'approvisionnement du serveur est égale à celle qu'il a consommé lors de son activation**

⇒ **La capacité du serveur est consommée au fur et à mesure de l'arrivée des événements à servir dans le système et la capacité restante est conservée. Au lieu de reconsidérer le budget alloué au serveur périodiquement (comme dans DS), on le fait lorsque les requêtes du serveur sont servies.**

Serveur sporadique : exemple

- 2 tâches périodiques P1 (C=1, T=5), P2 (C=2, T=6)
- la tâche serveur SS (C=1, T=4)
- 2 tâches apériodiques AP1 (R=1, C=0,75), AP2(R=3, C=1),



Variante du serveur sporadique

Une variante consiste à transformer le serveur sporadique en serveur en arrière plan (faible priorité) lorsque sa capacité est épuisée afin de récupérer le temps non-alloué

- 2 tâches périodiques P1 (C=1, T=5), P2 (C=2, T=6)
- la tâche serveur SS (C=1, T=4)
- 2 tâches apériodiques AP1 (R=1, C=0,75), AP2(R=3, C=1),

Serveur périodique et ordonnancement dynamique EDF

Les ordonnancements dynamiques permettent, à priori, une meilleure utilisation du processeur. Leur difficulté de mise en oeuvre provient du fait qu'ils ne peuvent pas distinguer les tâches qui ont des échéances temporelles dures, des tâches qui ont moins de contraintes. L'idée est de servir les tâches aux contraintes plus lâches au moyen de serveurs périodiques.

On retrouve la même mise en oeuvre de serveur sporadique et serveur différé dans le cas de EDF.

Serveur périodique et ordonnancement dynamique EDF

Exemple EDF/DS

Soit un système composé de deux tâches périodiques et d'un serveur d'événements apériodiques, la tâche 1 est définie par T1= 8, C1 = 3, la tâche 2 par T2=6,C2=2, DS par Tds=4, Cds=1.

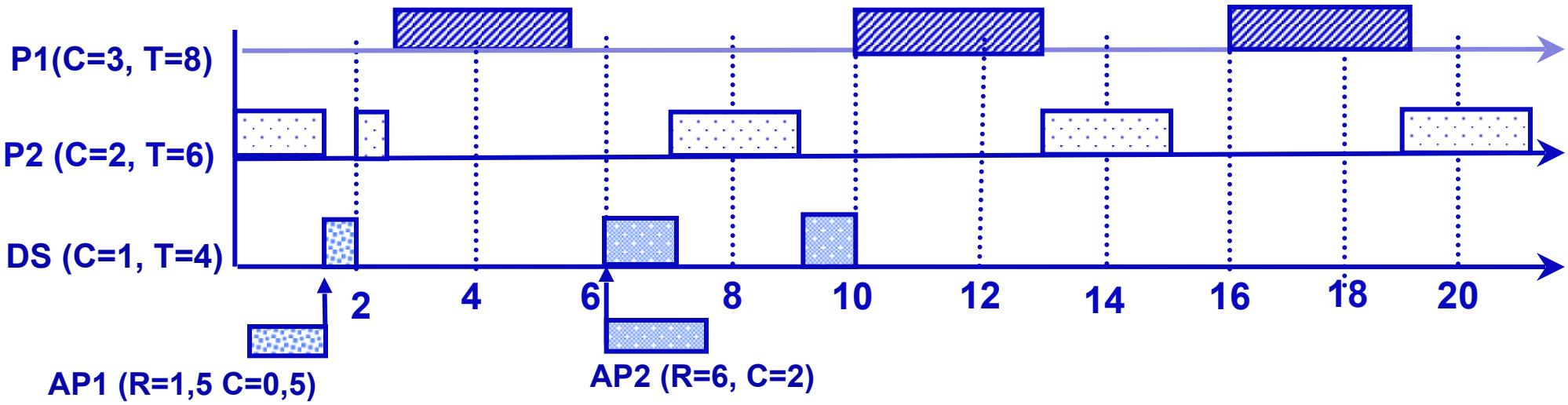
Les tâches 1 et 2 sont prêtes à l'instant 0. Le premier événement apériodique arrive à t = 1.5 et réclame 0.5 unités de traitement. Le deuxième événement arrive à t = 6 et réclame 2 unités de traitement

1. Faire le test de faisabilité de l'ordonnancement EDF
2. Donner le diagramme d'ordonnancement

Serveur périodique et ordonnancement dynamique EDF

Exemple EDF/DS

1. le test de faisabilité : 3/8 + 2/6 + 1/4 = 23/24 < 1
2. le diagramme d'ordonnancement



Algorithme "Slack Stealer"

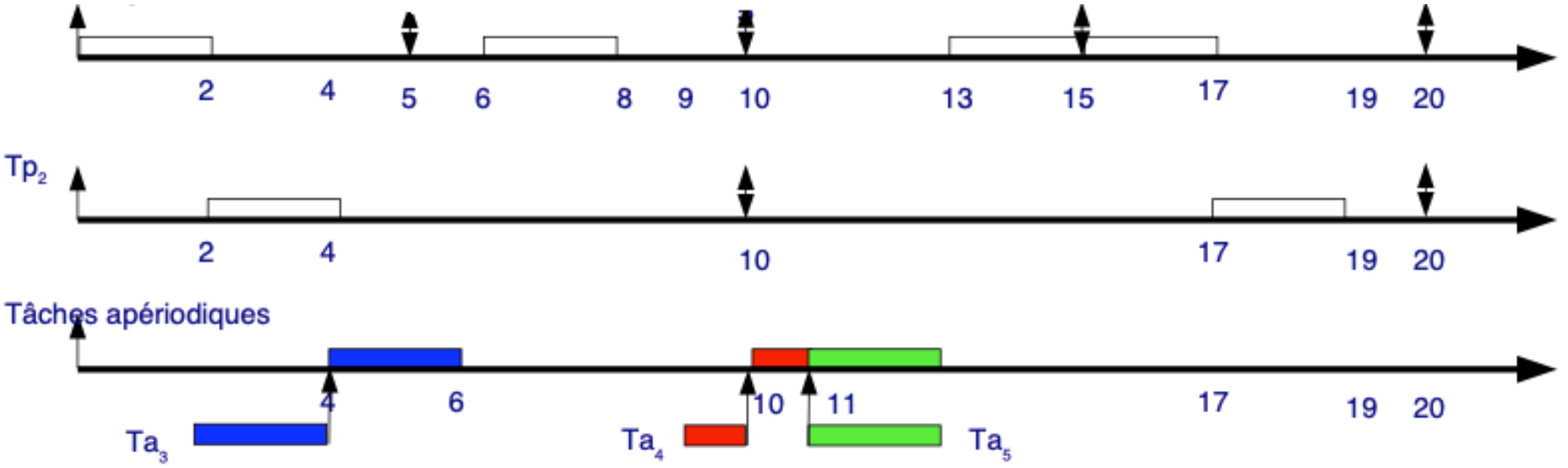
⇒ **Slack Stealer s'appuie sur RMA**

⇒ **Pas de serveur pour les tâches apériodiques**

⇒ **Lorsque une tâche apériodique est activée, le système recule au maximum le démarrage des tâches périodiques pour autant qu'elles continuent à respecter leurs contraintes temporelles. On utilise la laxité des tâches périodiques pour ordonnancer les tâches apériodiques au plus tôt**

Exemple d'algorithme "Slack Stealer"

Tp_1 (r_0=0, C=2, P=5), Tp_2 (r_0=0, C=2, P=10)
Ta_3 (r=4, C=2), Ta_4 (r=10, C=1), Ta_5 (r=11, C=2)



- à t=4 : Ta_3 est activée
- à t=5 Tp_1 est réveillée et $L(Tp_1) = 3$ on peut reculer l'exécution deTp1 jusqu'à t=6 pour laisser Ta3 se terminer
- de même à t=10 et t=11, recul de Tp1 jusqu'à t=13 et Tp2 jusqu'à t=17 pour laisser Ta_4 et Ta_5 s'exécuter

Tâches apériodiques à contraintes strictes

2 approches

⇒ **Les traiter comme des tâches périodiques avec une pseudo-période représentant l'intervalle minimal entre 2 occurrences.**

- **utilisé dans le cas RMA**
- **mais difficile d'évaluer la pseudo-période**
- **engendre une sous-utilisation du processeur**

⇒ **Ordonnancer les tâches en EDF. A chaque nouvelle tâche apériodique, faire tourner une "routine de garantie" pour vérifier que toutes les contraintes temporelles seront respectées. Si non, refuser la tâche.**

- **2 politiques d'acceptation dynamique**
- **favorise les tâches périodiques**

Acceptation dans les temps creux d'une séquence rigide de tâches

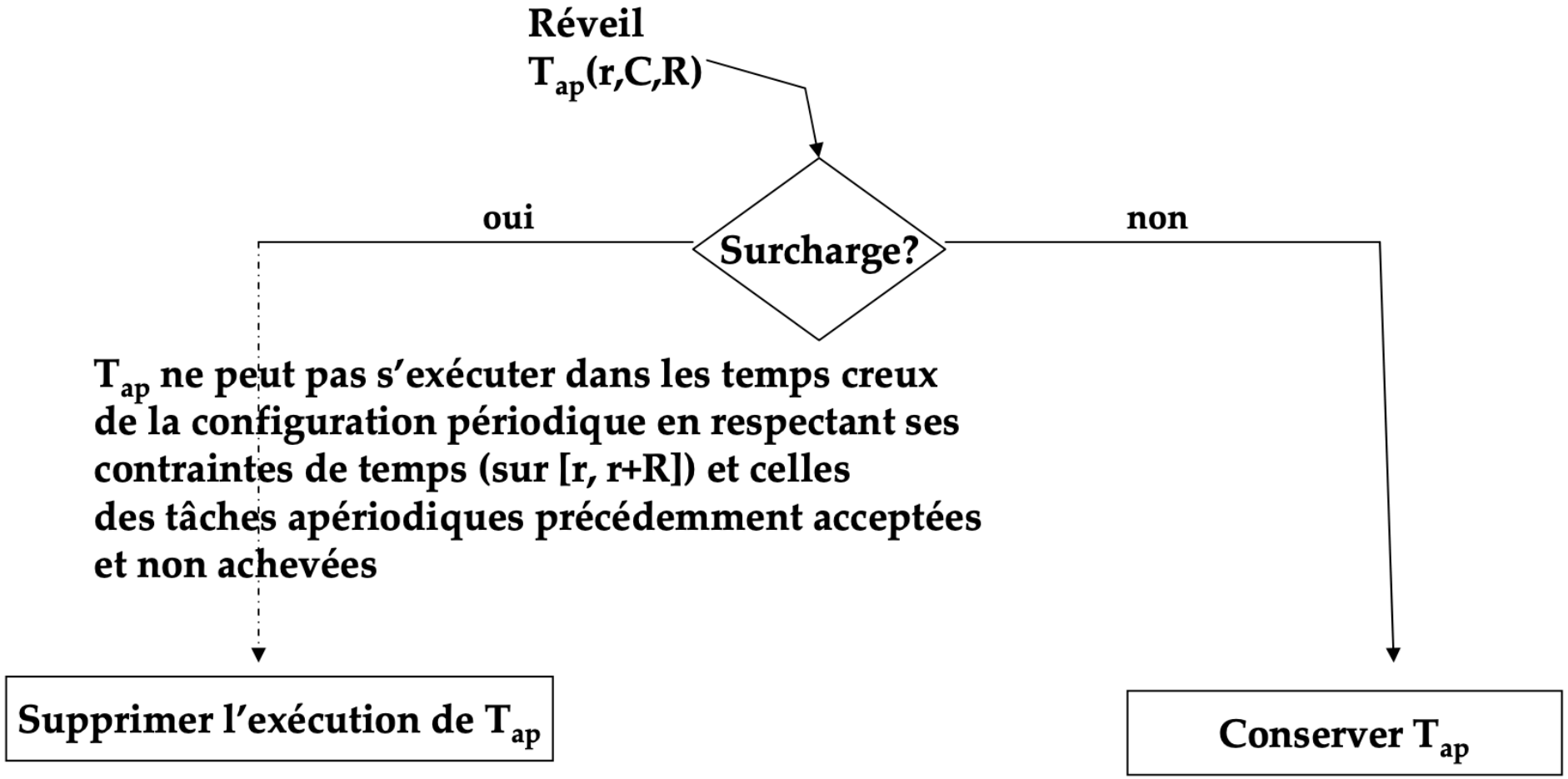
⇒ **Ordonnancement EDF des tâches périodiques**

⇒ **Les tâches apériodiques acceptées sont ordonnancées dans les temps creux des tâches périodiques (~ méthode d'arrière-plan) selon l'algorithme EDF**

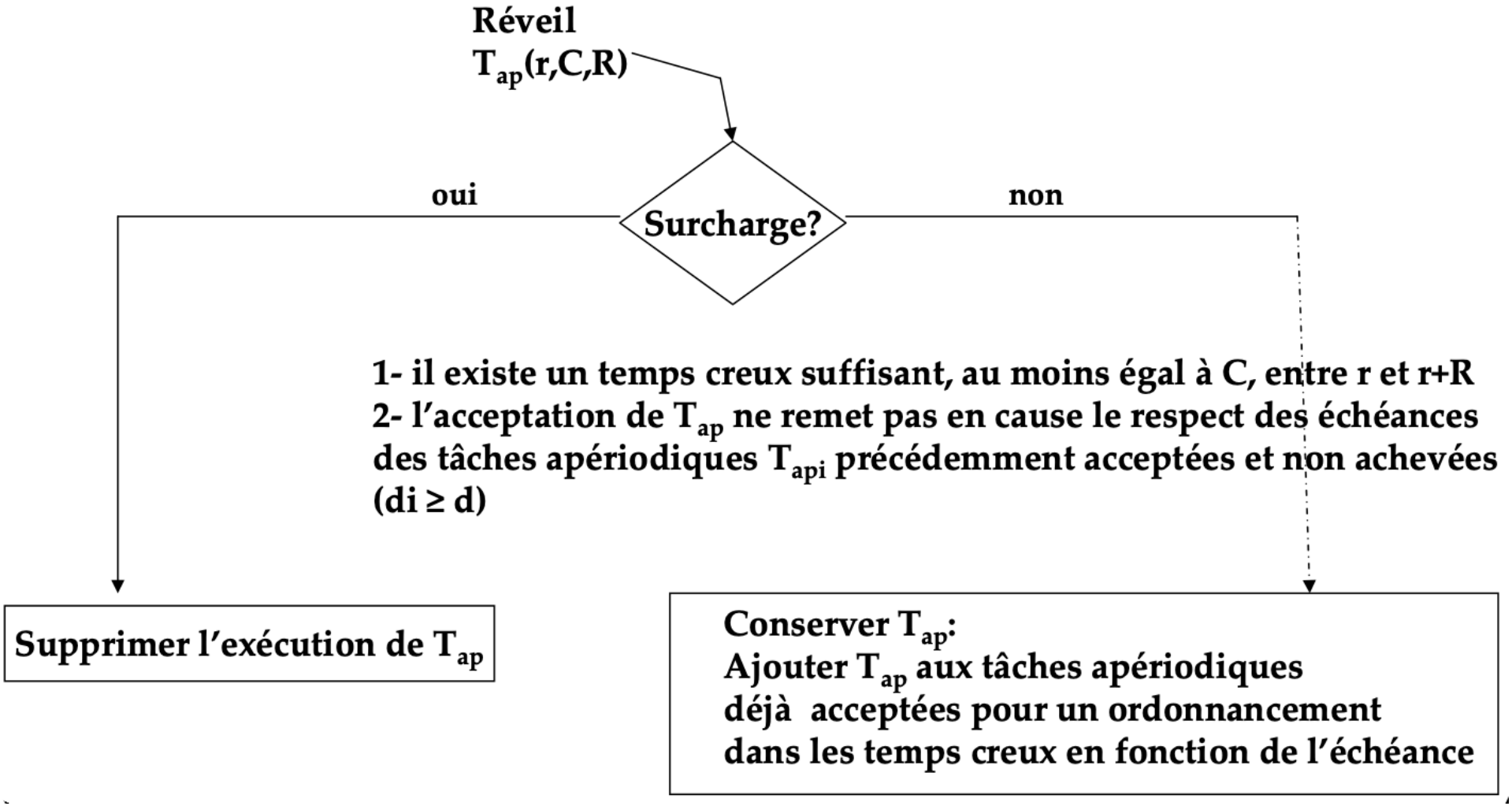
⇒ **Routine de garantie (au réveil d'une tâche apériodique):**

- **Teste l'existence d'un temps creux suffisant entre le réveil et l'échéance de la tâche apériodique)**
- **Vérifie que l'acceptation de la nouvelle tâche ne remet pas en cause le respect des contraintes temporelles des autres tâches apériodiques déjà acceptées et non encore terminées**
  - ✦ **Si OK, la tâche est ajoutée à la liste des tâches apériodiques acceptées**.

Routines de garantie: méthode des temps creux

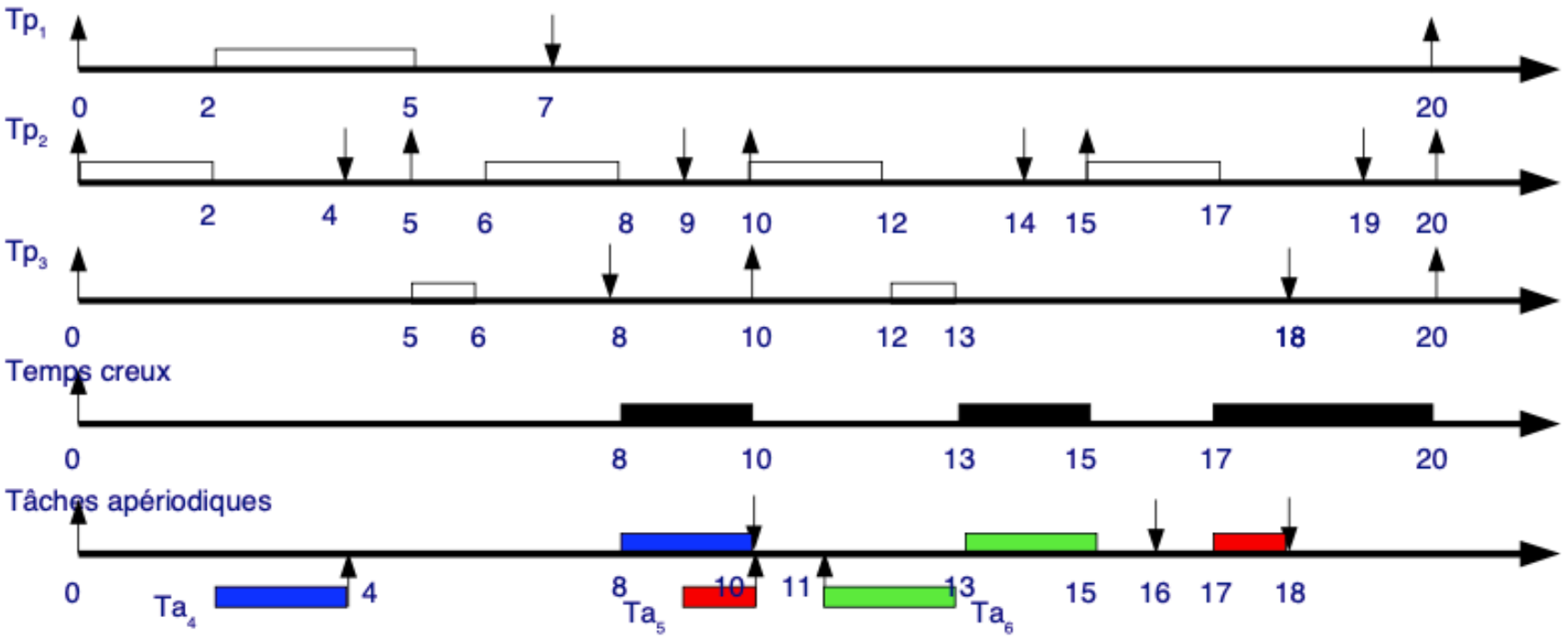


Routines de garantie: méthode des temps creux



Exemple de l'acceptation dans les temps creux

- Tp_1 (r_0=0, C=3, D=7, P=20), Tp_2 (r_0=0, C=2, D=4, P=5), Tp_3 (r_0=0, C=1, D=8, P=10)
- Ta_4 (r=4, C=2, d=10), Ta_5 (r=10, C=1, d=18), Ta_6 (r=11, C=2, d=16)



Exercices :

Soient deux tâches P_1 et P_2 et un serveur différé S ordonnancés par RMA.

On a $C_1 = 4$, $T_1 = 12$, $C_2 = 6$, $T_2 = 16$, $C_{serv} = 2$ et $T_{serv} = 10$. Soient les événements e_1,e_2 et e_3 arrivant aux instants t=2, t=14 et t=28 et déclenchant des tâches apériodiques de capacités 1, 4 et 2 respectivement.

Donner le chronogramme et représenter la capacité de S.