

Chapitre 4

Algorithmes à vagues

- Capteurs de température
- Circulation de jeton
- Parcours du réseau exploitant le parallélisme
→ parcours d'arbre

Algorithmes à vagues - Définition

Un **algorithme à vague** vérifie :

- **Terminaison** : toute exécution est finie
- **Décision** : toute exécution contient un événement appelé « *décision* »
- **Dépendance** : tout événement de décision est causalement précédé d'au moins un événement de chaque processus

Une exécution = une vague

Problème de la circulation de jeton

Exemple de problème résolu par un algorithme à vague

- 1) Un nœud initiateur envoie un jeton dans le réseau
- 2) Le jeton doit faire le tour de tous les nœuds avant de revenir à l'initiateur (*décision*)

Spécification

Sûreté :

- 1) Au plus un jeton dans le réseau
- 2) Au plus une décision de prise
- 3) Si une décision est prise, tous les processus ont obtenu le jeton préalablement.

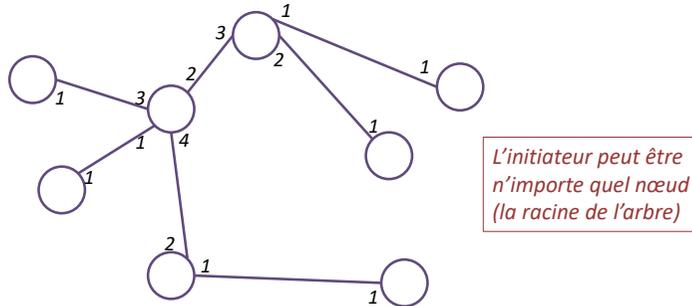
Vivacité :

- 4) La décision est prise en temps fini.
- 5) L'exécution termine.

Exemple de problème résolu par un algorithme à vague **car**

- 5) => **Terminaison** ; 2)+4) => **Décision** ; 3) => **Dépendance**

Circulation de jeton dans un arbre



Hypothèses

- Topologie = réseau en arbre bidirectionnel
- Pour chaque nœud p , les canaux sont étiquetés de 1 à $\delta(p)$
- Exactement un nœud initiateur
- Pas de faute, tout asynchrone

Les nœuds n'ont pas besoin d'être identifiés.

NB : l'étiquetage sur les canaux implique un ordre local sur les canaux.

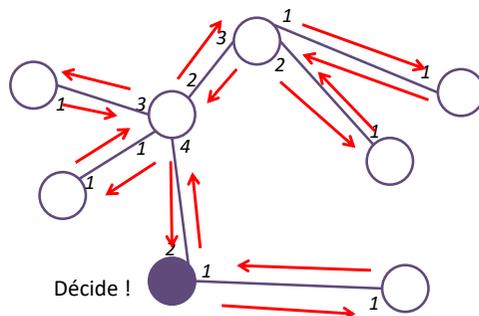
Circulation de jeton dans un arbre

Principe

- Le **jeton** est créé par l'*initiateur*
- Le jeton circule dans les processus,
 - Selon un **parcours en profondeur d'abord**
 - En utilisant l'ordre des canaux
- L'initiateur **décide** quand le jeton lui revient

→ Voir algorithme

Circulation de jeton dans un arbre Exemple d'exécution



Circulation de jeton dans un arbre - Correction

Remarque : Les deux premiers points sont vérifiés

- 1) Au plus un jeton dans le réseau
- 2) Au plus une décision de prise

Lemme A : Si p obtient le jeton pour la première fois d'un voisin q , alors q est le père de p dans l'arbre enraciné en l'initiateur.

Preuve au tableau

Lemme B : Si p obtient le jeton pour la première fois d'un voisin q , alors p envoie ensuite le jeton à tous ses autres voisins avant de le renvoyer à q .

Preuve au tableau

Circulation de jeton dans un arbre - Correction

Lemme C : le 3^e point est vérifié

Preuve au tableau

3) Si une décision est prise, tous les processus ont obtenu le jeton avant.

Lemme D : les 4^e et 5^e points sont vérifiés

Preuve au tableau

4) La décision est prise en temps fini. 5) L'exécution termine.

Théorème : l'algorithme résout le problème de la circulation de jeton.

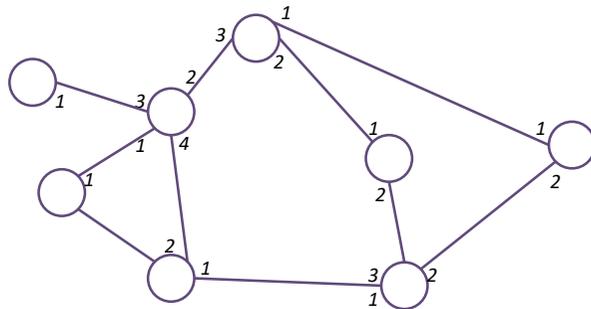
Complexité :

Combien de messages échangés avant décision ? $2(n - 1)$

Nombre de pas de calcul ?

Complexité en temps ?

Propagation d'information avec retour



Hypothèses

- Topologie connexe, bidirectionnelle (au moins 2 nœuds)
- Pour chaque nœud p , les canaux sont étiquetés de 1 à $\delta(p)$
- Exactement un nœud initiateur
- Pas de faute, tout asynchrone

Les nœuds n'ont pas besoin d'être identifiés.

NB : l'étiquetage sur les canaux implique un ordre local sur les canaux.

Propagation d'information avec retour

Une donnée d est propagée dans le réseau.

- 1) Depuis un initiateur
- 2) Tout nœud recevant d accuse réception
- 3) L'initiateur décide une fois qu'il a reçu les accusés de réception de tous les nœuds.

Spécification

Sûreté :

- 1) Au plus une décision de prise
- 2) Si une décision est prise, elle vient causalement après un accusé de réception de d envoyé par tous les nœuds non initiateurs.

Vivacité :

- 4) Une décision est prise en temps fini.
- 5) L'exécution termine.
- 6) Tout processus non initiateur a reçu d et en a accusé réception.

Exemple de problème résolu par un algorithme à vague car

5) => **Terminaison** ; 1)+4) => **Décision** ; 2) => **Dépendance**

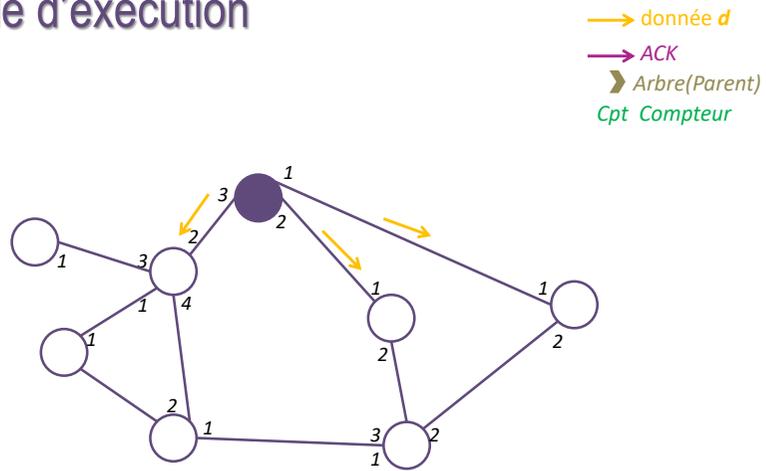
Propagation d'information avec retour

Principe

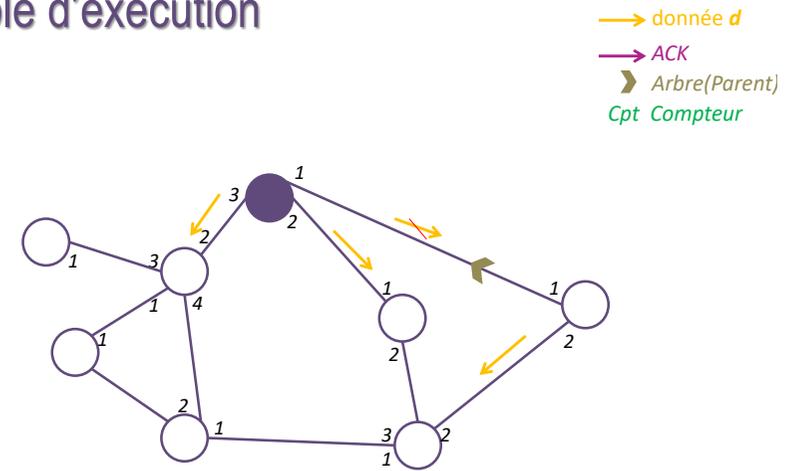
- La donnée d est broadcastée par l'initiateur, puis renvoyée par les nœuds
- Un arbre est construit au fur et à mesure : il sert à faire remonter les ACK jusqu'à la racine
- L'initiateur **décide** quand il a reçu des ACK de chacun de ses fils dans l'arbre

→ Voir algorithme

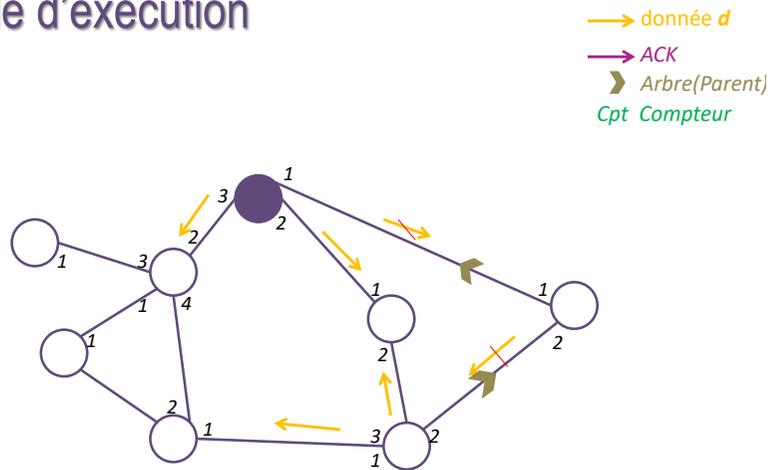
Propagation d'information avec retour Exemple d'exécution



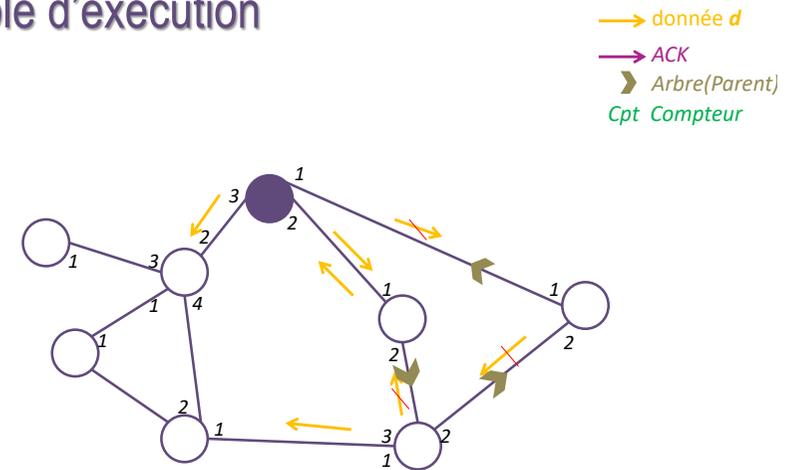
Propagation d'information avec retour Exemple d'exécution



Propagation d'information avec retour Exemple d'exécution

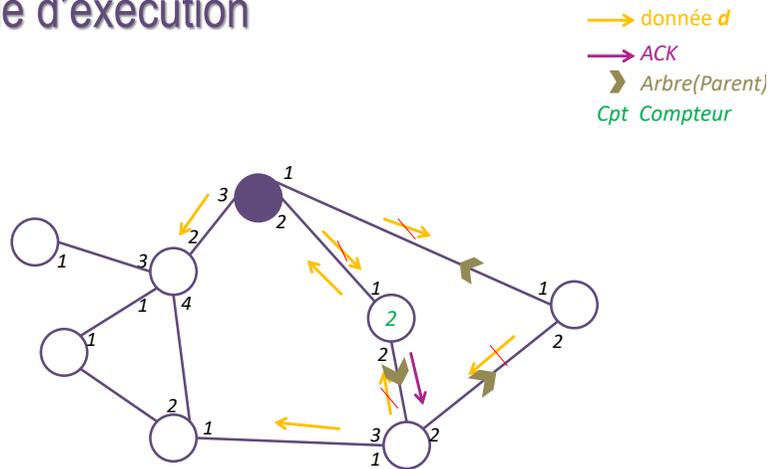


Propagation d'information avec retour Exemple d'exécution



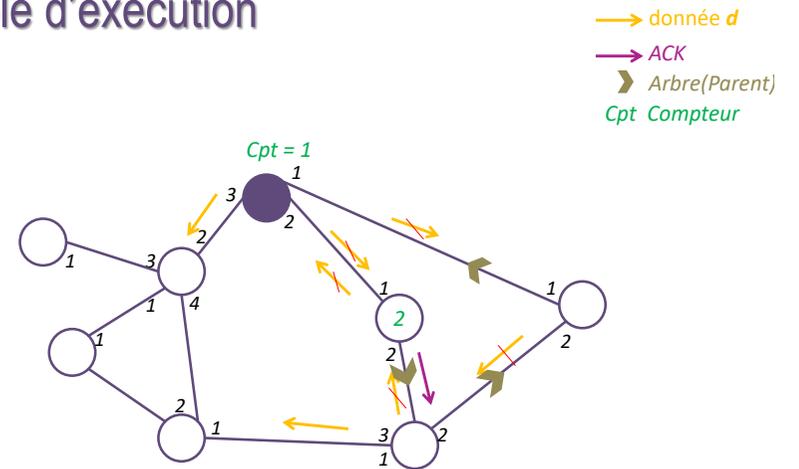
Propagation d'information avec retour

Exemple d'exécution



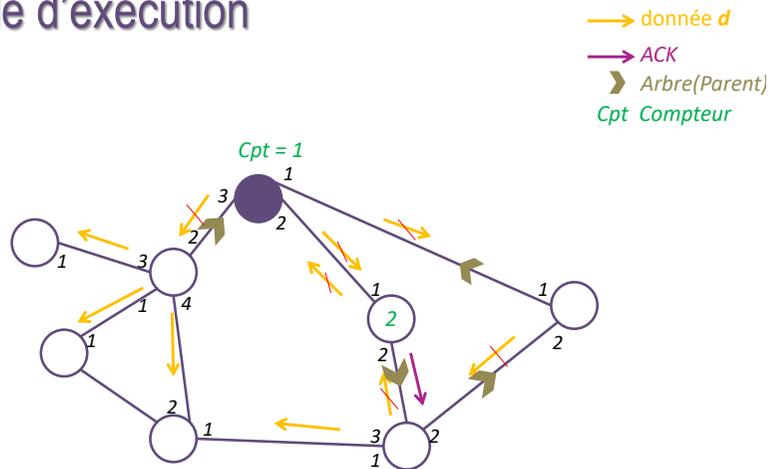
Propagation d'information avec retour

Exemple d'exécution



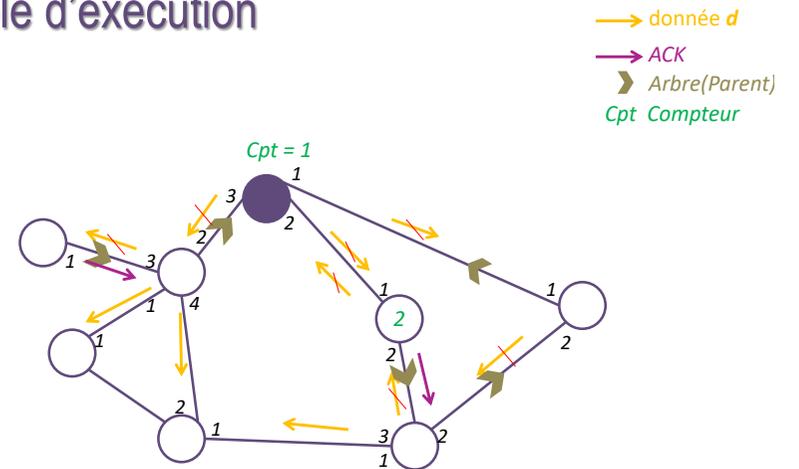
Propagation d'information avec retour

Exemple d'exécution



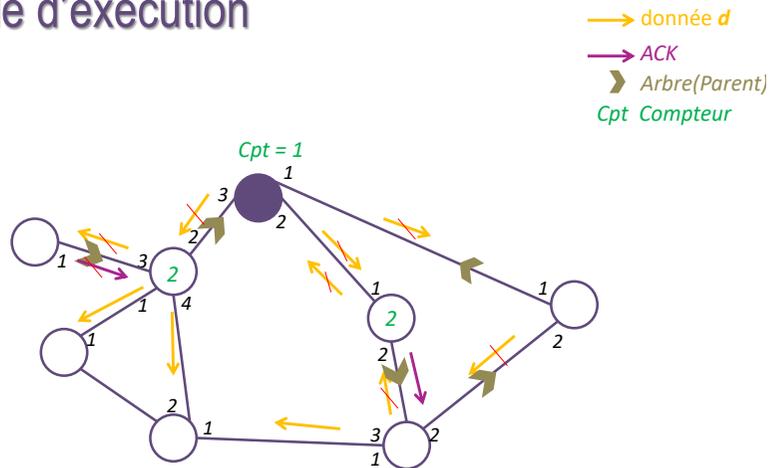
Propagation d'information avec retour

Exemple d'exécution



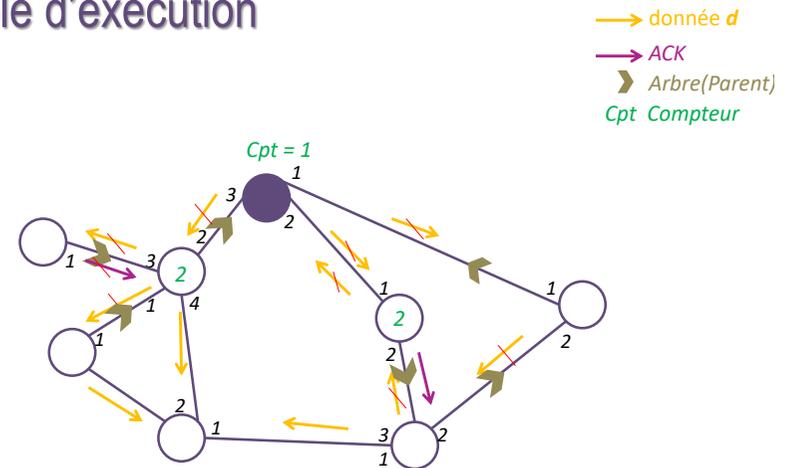
Propagation d'information avec retour

Exemple d'exécution



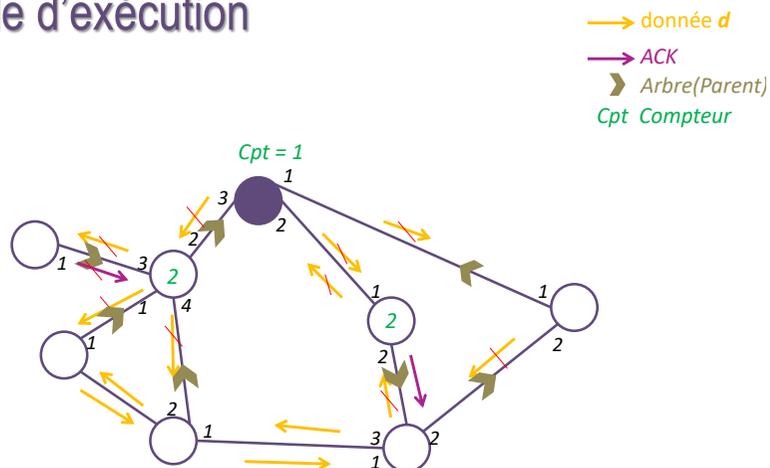
Propagation d'information avec retour

Exemple d'exécution



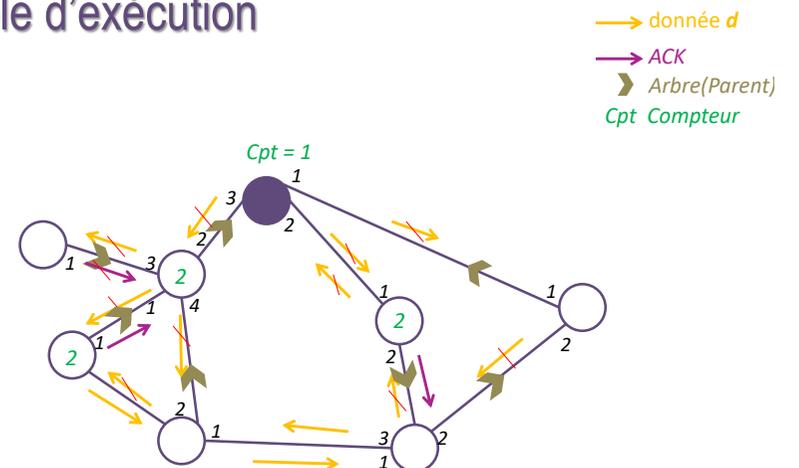
Propagation d'information avec retour

Exemple d'exécution

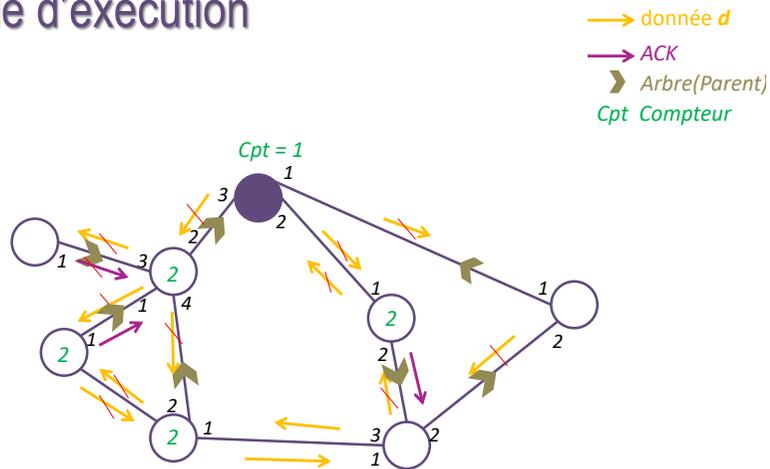


Propagation d'information avec retour

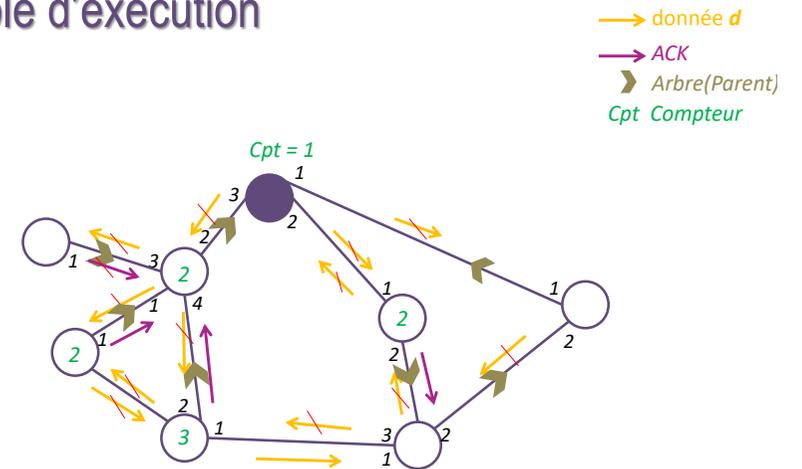
Exemple d'exécution



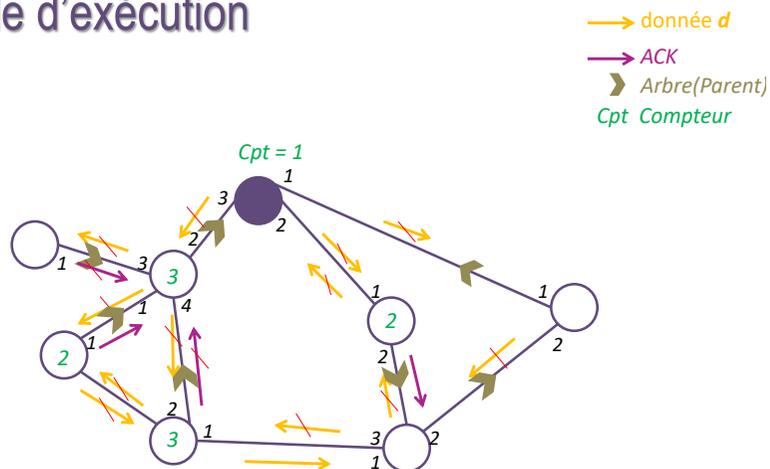
Propagation d'information avec retour Exemple d'exécution



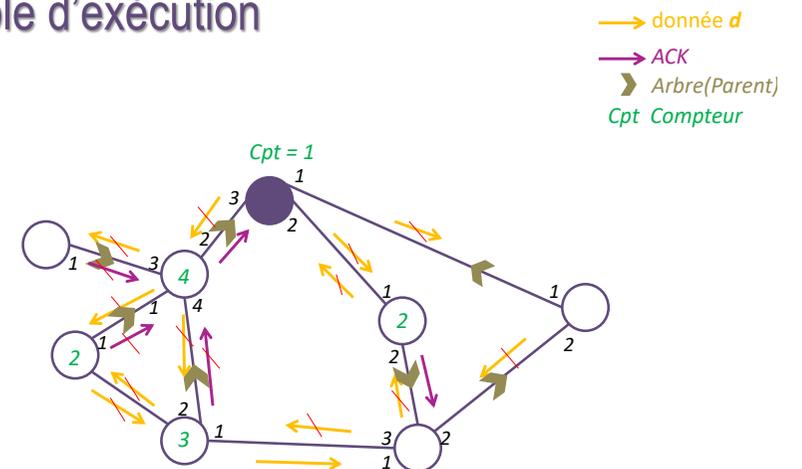
Propagation d'information avec retour Exemple d'exécution



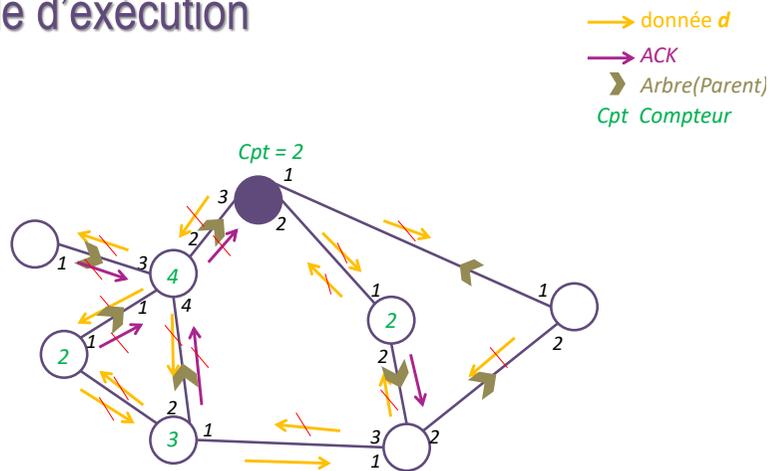
Propagation d'information avec retour Exemple d'exécution



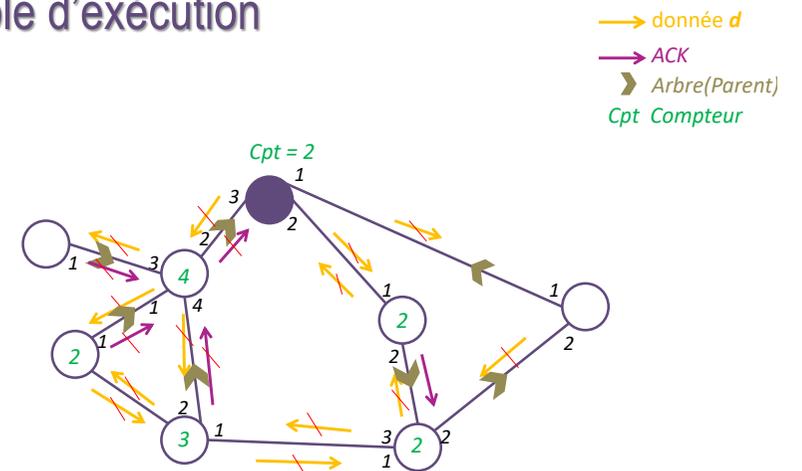
Propagation d'information avec retour Exemple d'exécution



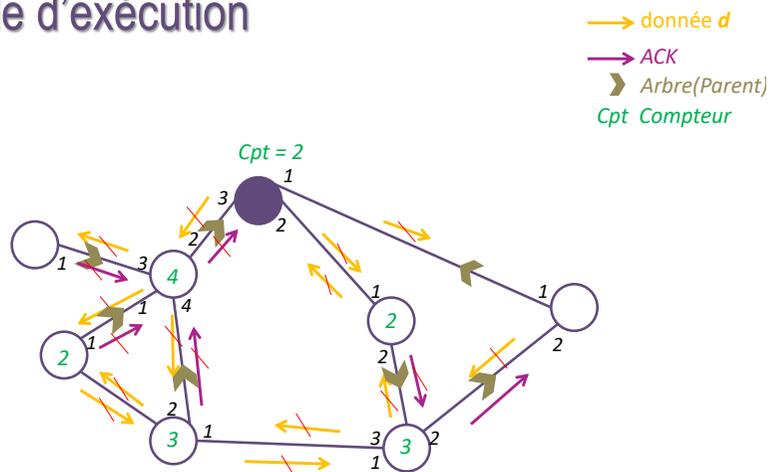
Propagation d'information avec retour Exemple d'exécution



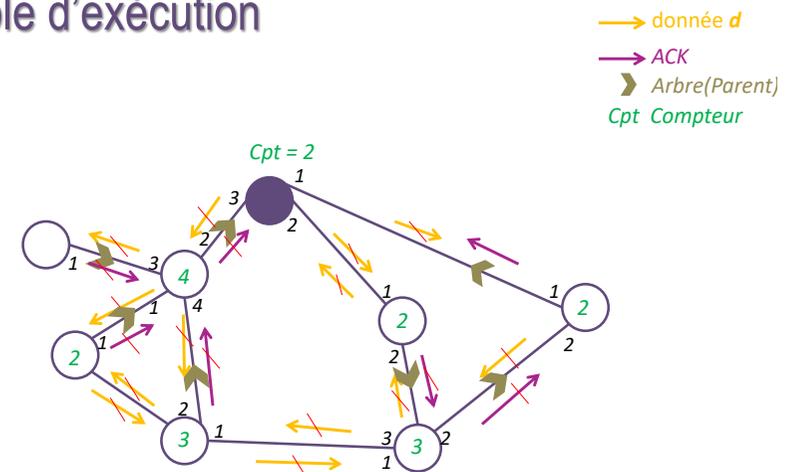
Propagation d'information avec retour Exemple d'exécution



Propagation d'information avec retour Exemple d'exécution

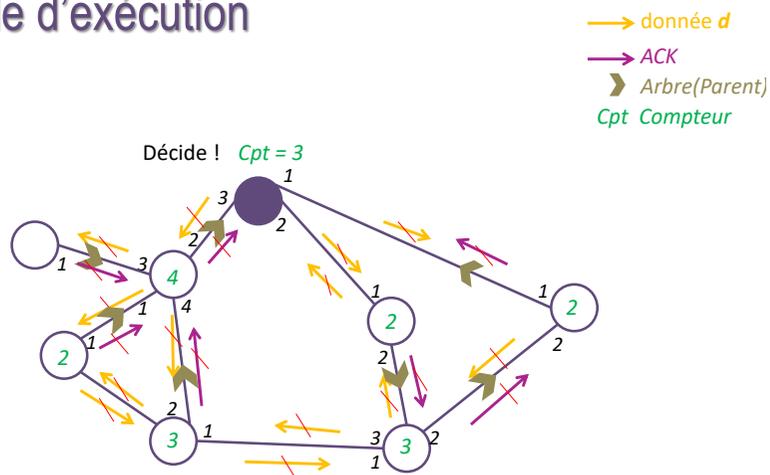


Propagation d'information avec retour Exemple d'exécution



Propagation d'information avec retour

Exemple d'exécution



Propagation d'information avec retour

Correction

Lemme A : Tout processus non initiateur reçoit un message $\langle Brd, d \rangle$.

Preuve au tableau

Lemme B : Soit T le graphe orienté dessiné par les pointeurs *Parents* :

T est un arbre couvrant enraciné à l'initiateur.

Preuve au tableau

Lemme C : Tout processus non initiateur envoie un message $\langle Ack \rangle$ vers son *Parent*.

Preuve au tableau

Corollaire : L'initiateur décide en temps fini.

Preuve au tableau

Lemme D : La décision dépend causalement des *Ack* de tous les nœuds non initiateurs.

Preuve au tableau

Propagation d'information avec retour

Complexité

Théorème : L'algorithme résout le problème de la propagation d'information avec retour.

Complexité :

Combien de messages échangés avant décision ?

$2 \times \text{nombre de liens}$

Nombre de pas de calcul ?

$1 + 2 \times \text{nombre de liens}$

Complexité en temps ?

$2 \times \text{diamètre}$
 $\text{Diamètre} + \text{nombre de noeuds}$