

La Stabilisation Instantanée

Alain Cournier, S. Devismes

Master 2 – Informatique, SDD

Université de Picardie

Plan du Cours

- La genèse

Genèse

- Article de A. Bui et al.
- Un algorithme autostabilisant pour la Diffusion d'Informations avec Accusé Réception (en anglais PIF).
- Topologie du réseau : Un arbre enraciné
- But initial : Minimiser le nombre d'états de l'algorithme

Genèse : Présentation de l'algorithme

- Les états possibles :
 - B : Diffusion du message (Broadcast)
 - F : Envoie de l'accusé réception (Feedback)
 - C : Prêt pour une nouvelle diffusion

Genèse : Présentation de l'algorithme

- Pour les feuilles :
- Deux états possibles : F et C
- Règles de changement :

Règle	Etat feuille	Etat Père	Nouvel état
F.Propagation	C	B	F
F.Nettoyage	F	F ou C	C

Genèse : Présentation de l'algorithme

- Pour les nœuds internes :
- Trois états possibles : B, F et C
- Règles de changement :

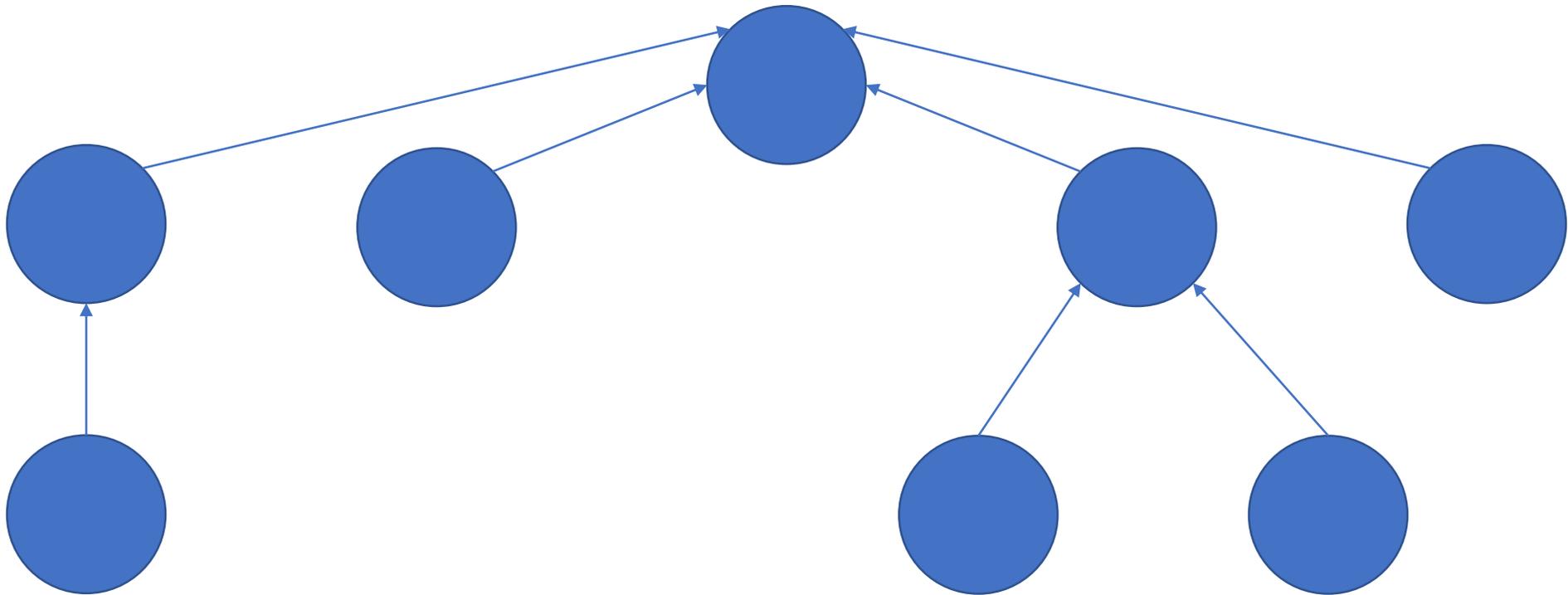
Règle	Etat nœud	Etat père	Etat des fils	Nouvel état
I.Propagation	C	B	C	B
I.AR	B	B, F ou C	F	F
I.Nettoyage	F	F ou C	F ou C	C

Genèse : Présentation de l'algorithme

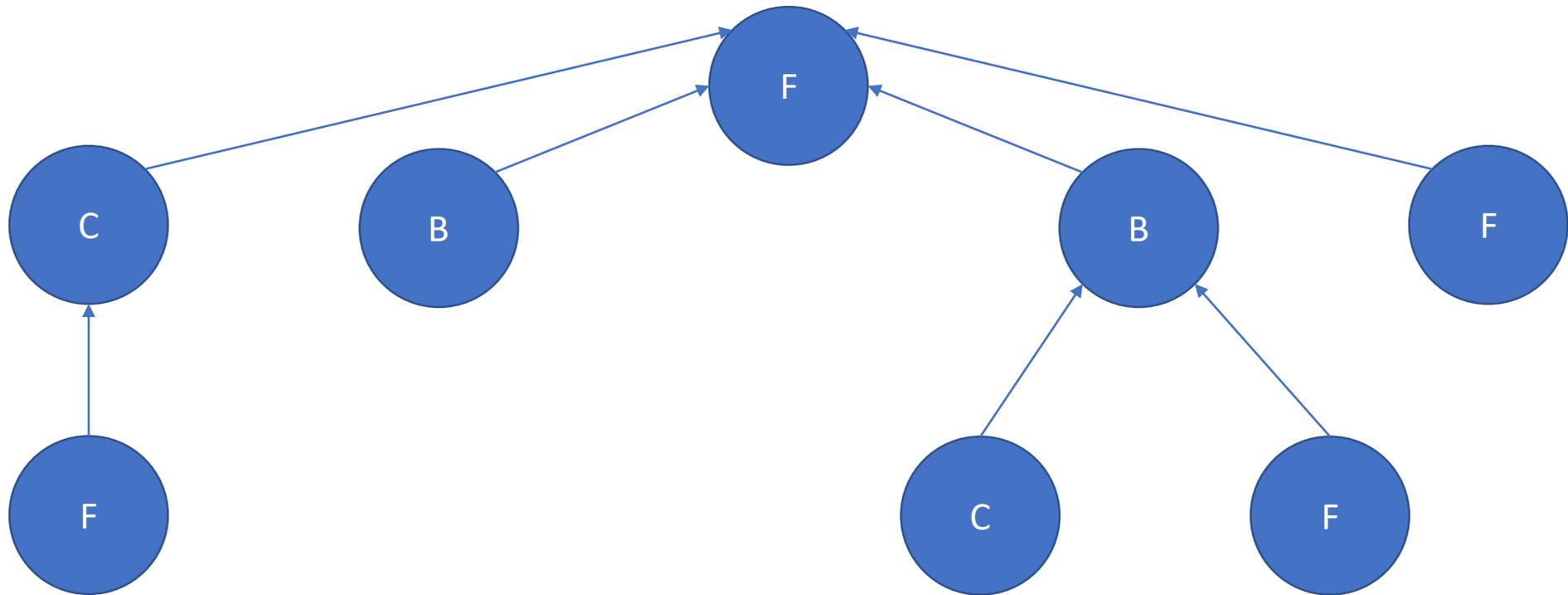
- Pour la racine :
- Deux états possibles : B et F
- Règles de changement :

Règle	Etat racine	Etat des fils	Nouvel état
R.Propagation	F	C	B
R.AR	B	F	F

Exercice 1 : Initialement tous en F



Exercice 2



Exercice 3

- Démontrez que tout message diffusé par la racine (par l'exécution de R.Propagation) arrivera une fois et une seule à chaque nœud de notre réseau.
- Démontrez que pour tout message diffusé par la racine (par l'exécution de R.Propagation), l'exécution suivante de l'action R.AR signe un accusé réception global du message par tous les nœuds du réseau.

Points forts de la stabilisation instantanée

- Si un service est demandé, il sera rendu, indépendamment des conditions initiales (en l'absence de nouvelles fautes).
- C'est une réponse à la question :
 - « Combien de fois dois-je demander un service pour avoir la certitude qu'il soit rendu au moins une fois ? »

Conséquences

- Le temps de stabilisation n'a plus de sens.
- Le **délai** est le temps nécessaire pour que l'algorithme puisse initier sa première demande de service.
- Passé ce délai nous ne sommes pas obligatoirement dans une configuration « légitime ». Mais dans une configuration qui permet au service d'être rendu.

Conséquences

- Combien de temps mettra l'algorithme pour rendre le service pour la première fois ?
- Combien de temps mettra l'algorithme pour rendre le service pour la en dehors de la première fois ?
- Calculez ces coûts sur l'algorithme proposé.

Conséquences

- Combien de temps mettra l'algorithme pour rendre le service lorsqu'il débute dans une configuration « normale » ?
- Quel est dans ce cas le surcoût ?
- Calculez ces coûts sur l'algorithme proposé.

Quand on parle de qualité de service

- Nous avons regardé la qualité de service du côté de celui qui ordonne le service (ici celui qui initie le PIF).
- Que se passe-t-il du côté de celui qui subit le service (Il n'a rien demandé mais il reçoit une demande, une requête) ?

Quand on parle de qualité de service

- Combien de fois un nœud peut-il être traversé par un protocole non initié par la racine ?
- Les protocoles non initiés par la racine sont-ils tous reçus consécutivement ?
- Répondez à ces questions avec notre algorithme.
- Peut-on améliorer notre algorithme selon ces critères ?

Quand on parle de qualité de service

- Combien de fois un nœud peut-il être traversé par un protocole non initié par la racine sans pouvoir détecter une anomalie ?
- Les protocoles non initiés par la racine indétectable sont-ils tous reçus consécutivement ?
- Répondez à ces questions avec notre algorithme.
- Peut-on améliorer notre algorithme selon ces critères ?

Quand on parle de qualité de service

- En l'absence de toute nouvelle requête au bout de combien de temps notre algorithme n'effectuera plus aucune action ?
- Répondez à cette question avec notre algorithme.
- Peut-on améliorer notre algorithme selon ce critère ?

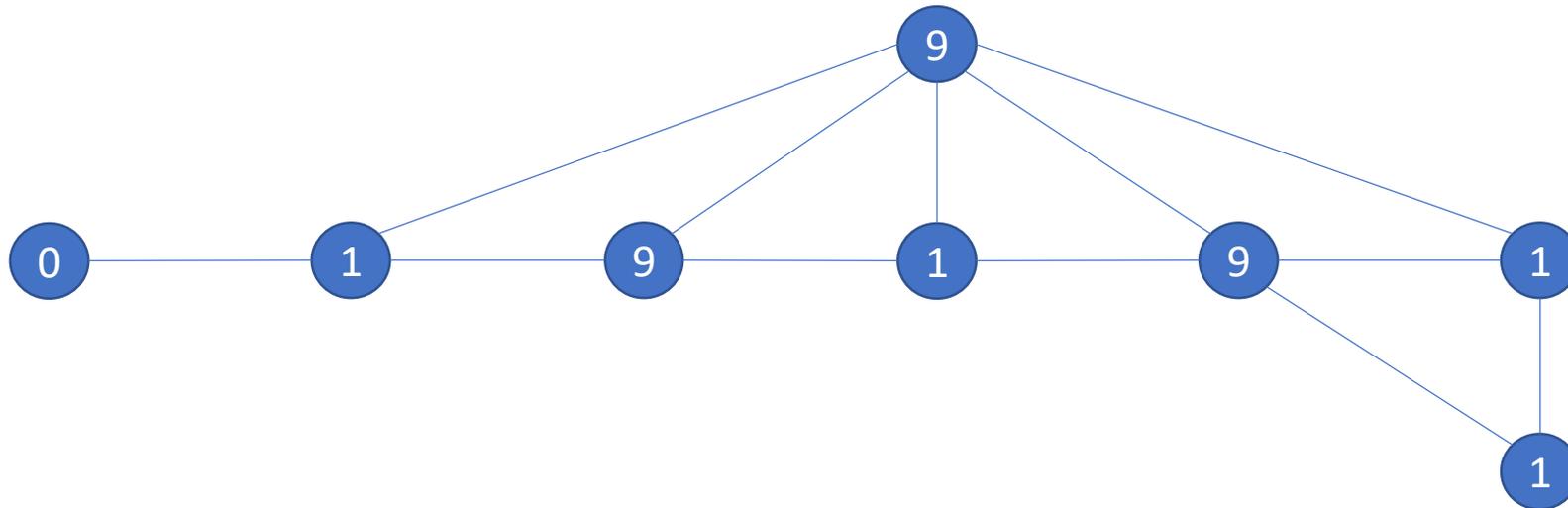
Sur la dualité du temps de calcul

- Devons nous le calculer en nombre de pas de calcul (lorsque c'est possible) ?
- Devons nous le calculer en nombre de ronde ?
- Devons nous calculer un travail (nombre d'actions dans le pire des cas) ?
- Ces mesures s'opposent-elles les une au autres ?

Un pb simple : construire un arbre couvrant

- Premier algorithme :
 - Une variable pour chaque processus h
 - La racine r :
 - $h_r \neq 0 :: h_r \leftarrow 0$
 - Un nœud $p \neq r$:
 - $(\forall q \in V(p), h_p \neq h_q + 1) \text{ ou } (\exists q \in V(p), h_p > h_q + 1) :: h_p \leftarrow \text{Min}_{q \in V(p)}\{h_q\} + 1$

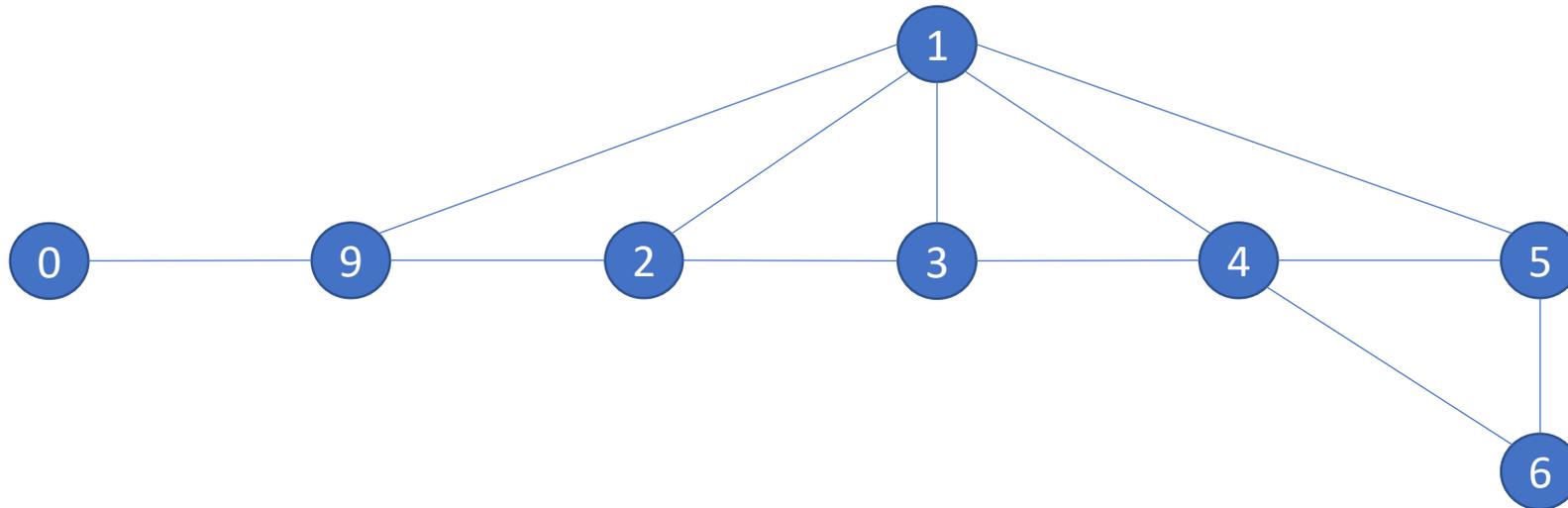
- Exécutez ce premier algorithme sur le réseau suivant :



Un pb simple : construire un arbre couvrant

- Second algorithmme :
 - Une variable pour chaque processus h
 - La racine r :
 - $h_r \neq 0 :: h_r \leftarrow 0$
 - Un nœud $p \neq r$:
 - $(\forall q \in V(p), h_p < h_q + 1) :: h_p \leftarrow \text{Max}_{q \in V(p)}\{h_q\} + 1$

- Exécutez ce second algorithme sur le réseau suivant :



Dualité du temps de calcul

- Calculez les complexités (rondes et pas de calcul) de ces deux algorithmes

- Que remarquez vous

Dualité du temps de calcul

- Est-il possible de réconcilier ces deux complexité ?
- Un algorithme distribué autostabilisant est dit totalement polynômial si et seulement si les deux conditions suivantes sont vérifiées :
 - Sa complexité en nombre de rondes est polynômiale dans le diamètre du réseau
 - Sa complexité en nombre de pas de calcul est polynômiale dans le nombre de nœud du réseau