
SDR - Systèmes Distribués et Repartis**Notion de pannes***16 septembre 2025***Table des matières**

1	Types de pannes	1
1.1	Panne permantante	2
1.2	Panne récupérable	2
1.3	Panne arbitraire (Byzantine)	3
1.4	Autres types de pannes	3
2	Détecteur de pannes	3
2.1	Détecteur de pannes parfait	4
2.1.1	Propriétés	4
2.2	Heartbeat	4
2.2.1	Exemple	4
2.3	Détecteur de pannes parfait <u>un jour</u>	5
2.3.1	Propriétés	5
2.3.2	Timeout dynamique	6

1 Types de pannes

1.1 Panne permanente

Une panne permanente est une défaillance d'un composant du système qui persiste indéfiniment.

Dans notre cas, on dit qu'un processus est correct en terme de panne permanente quand il ne tombera jamais en panne permanente.



Fig. 1. – Capture des slides du cours – Panne permanente

i Info

L'aspect **correct** est une caractéristique théorique.

- Elle servira à analyser les algorithmes.
- Elle n'a aucun sens dans la vraie vie.

1.2 Panne récupérable

Une panne récupérable est une défaillance d'un composant du système qui peut être corrigée, permettant au composant de revenir à un état de fonctionnement normal.

Lors-que que notre processus revient d'une panne, il se peut qu'il ait perdu des informations, dans ce cas nous parlons d'**amnésie**.



Fig. 2. – Capture des slides du cours – Panne récupérable

On dit qu'un processus est correct en terme de panne récupérable quand il existe un instant T après lequel il ne tombera plus en panne.



Fig. 3. – Capture des slides du cours – Panne récupérable - correct

1.3 Panne arbitraire (Byzantine)

Une panne arbitraire, aussi appelée panne byzantine, est une défaillance d'un composant du système qui peut se comporter de manière imprévisible ou malveillante, affectant la fiabilité et la sécurité du système.



Fig. 4. – Capture des slides du cours – Panne byzantine

On dit qu'un processus est correct en terme de panne arbitraire quand il suivra toujours l'algorithme attendu.

Note

Les pannes byzantines sont les plus difficiles et couteuses à gérer dans les systèmes distribués, car elles peuvent impliquer des comportements malveillants ou erratiques.

1.4 Autres types de pannes

- **Panne d'omission:** lorsqu'un message devant être envoyé ne l'est pas.
- **Panne d'eavesdropping:** lorsqu'un message peut être lu par une entité extérieure au système.

2 Détecteur de pannes

2.1 Détecteur de pannes parfait

Un détecteur de pannes parfait est un mécanisme qui permet d'identifier avec certitude les processus défectueux dans un système distribué.

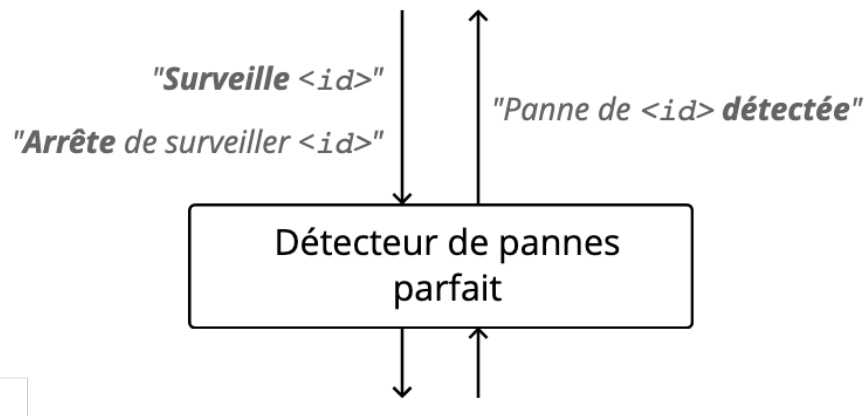


Fig. 5. – Capture des slides du cours – Détecteur de pannes parfait

2.1.1 Propriétés

- **Complétude:** Un jour, tout processus en panne sera détecté par tous les processus corrects.
- **Précision:** Si un processus p est détecté par un quelconque processus, alors p est effectivement en panne.

2.2 Heartbeat

Un détecteur de pannes de type **heartbeat** est un mécanisme utilisé dans les systèmes distribués pour surveiller l'état des processus en envoyant périodiquement des signaux (ou « battements de cœur ») entre eux.

2.2.1 Exemple

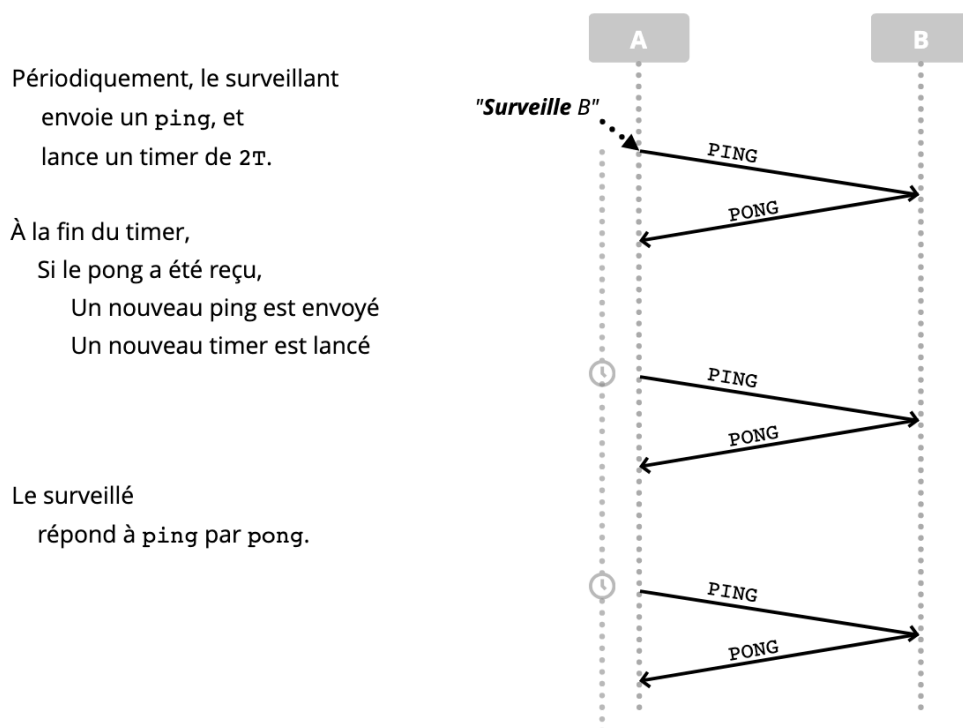


Fig. 6. – Capture des slides du cours – Détecteur de pannes Heartbeat

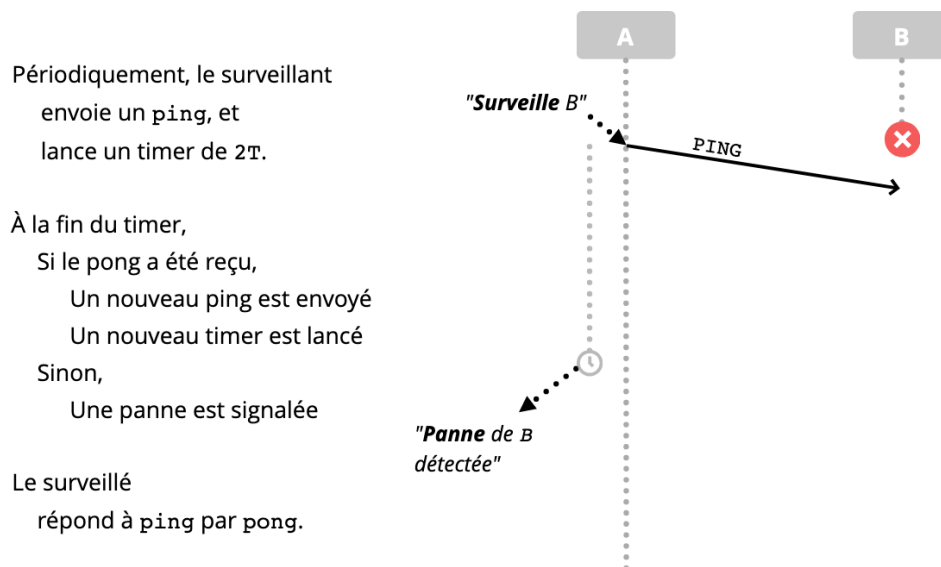


Fig. 7. – Capture des slides du cours – Détecteur de pannes Heartbeat - exemple

Dans ces exemples, nous voyons que notre système répond aux deux propriétés d'un détecteur de pannes parfait.

⚠ Warning

Cependant la supposition d'un système synchrone est irréaliste. Les vrais réseaux ne nous donnent pas de garantie sur la durée de transit du message. C'est pourquoi ce genre de détecteur de type Heartbeat donnera potentiellement des faux positifs, la notion de τ unité de temps n'existe pas.

2.3 Détecteur de pannes parfait un jour

Un détecteur de pannes parfait un jour est un mécanisme qui garantit que, après un certain temps, tous les processus corrects auront détecté tous les processus en panne, tout en maintenant la précision.

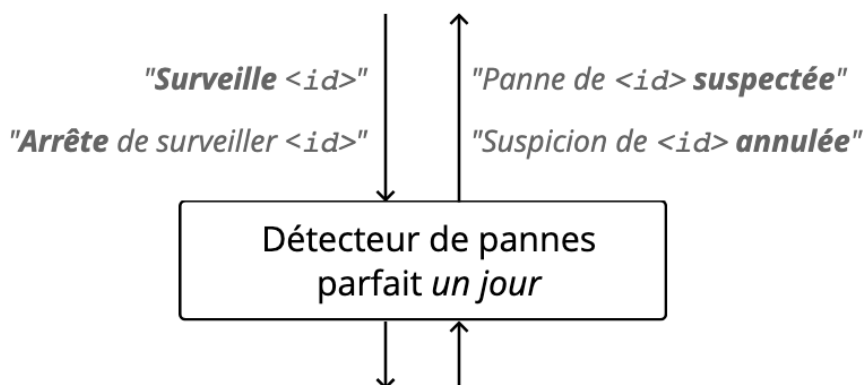


Fig. 8. – Capture des slides du cours – Détecteur de pannes parfait un jour

2.3.1 Propriétés

- **Complétude:** Un jour, tout processus en panne sera détecté par tous les processus corrects.
- **Précision un jour:** Un jour, aucun processus correct ne sera suspecté par processus correct.

2.3.2 Timeout dynamique

Initialement,
envoie un ping, et
lance un timer de durée Δ

À la fin du timer,
Si le pong n'a pas été reçu,
 Δ est incrémenté d'une constante
Une panne est suspectée
Envoie un ping, et
lance un timer de durée Δ

Si un pong est reçu d'un suspecté
La suspicion est annulée.

Le surveillé
répond à ping par pong.

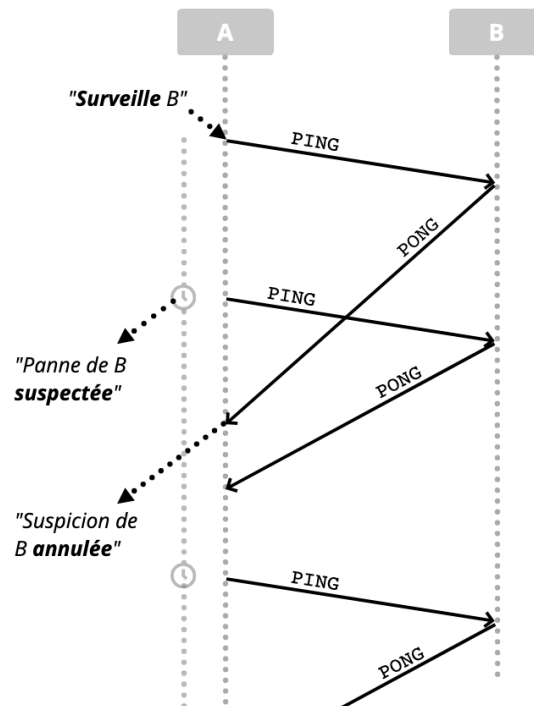


Fig. 9. – Capture des slides du cours – Détecteur de pannes parfait un jour - timeout dynamique

Grâce à cette nouvelle définition, nous pouvons donc affirmer les informations suivantes:

- Complétude: Un jour, tout processus en panne sera détecté par tous les processus corrects.
 - Oui, un processus en panne permanente ne répondra plus aux pings, et après Δ , cette panne sera découverte.
- Précision un jour: Un jour, aucun processus correct ne sera suspecté par processus correct.
 - Oui, un processus correct (en terme de panne permanente comme récupérable) répondra aux pings en un temps fini ; quand Δ sera assez grand, plus de panne ne sera suspectée : tout processus sera soit « en panne » et suspecté pour toujours, ou « correct » et plus jamais suspecté.

Note

Pour résoudre ce problème, était-il nécessaire d'avoir un timeout dynamique?

- Oui, sinon on risquerait de constamment suspecter un processus qui est simplement lent mais bien correct.