



# *Systemes distribués : beauté et difficulté*

*Vincent Villain*

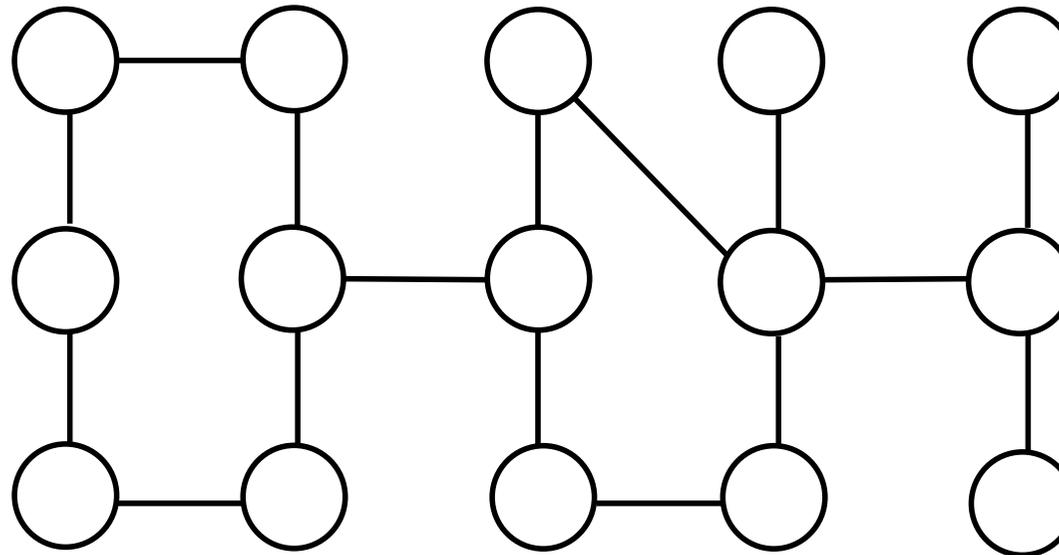
UTC

28 septembre 2010



# *Systeme distribué*

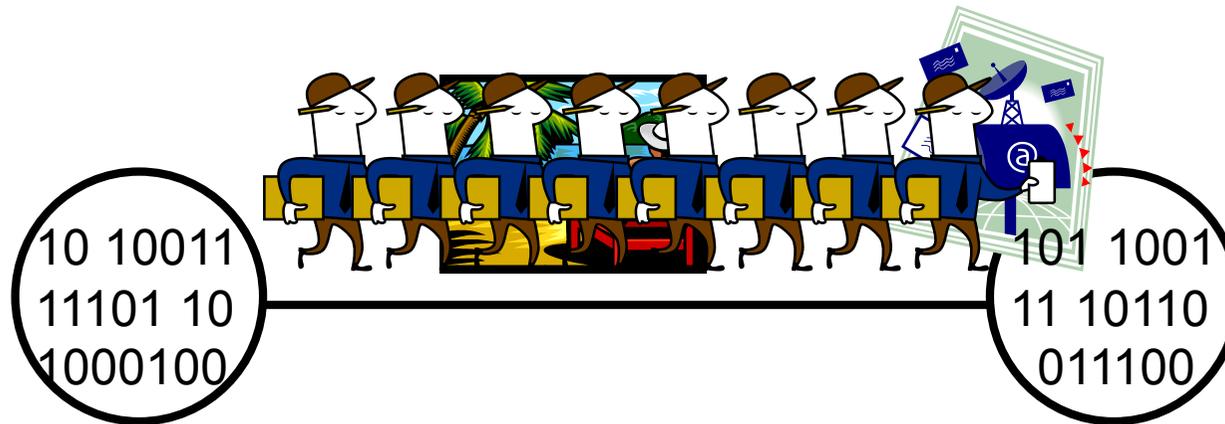
## Modèle pour les réseaux





# *Systeme distribué*

- horloge commune : NON
- mémoire partagée : NON
- communication par messages
- système asynchrone





# *Algorithme distribué*

Quel modèle ?

- Calculs locaux + envois de messages
- Initiateurs, non-initiateurs

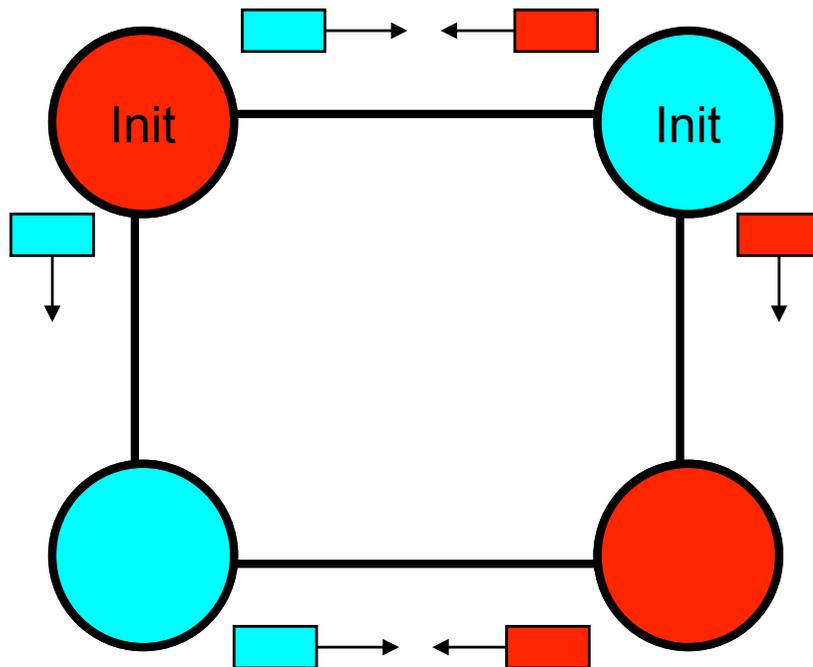
(Synchrone : fonctionnement par phase)

Asynchrone : ...



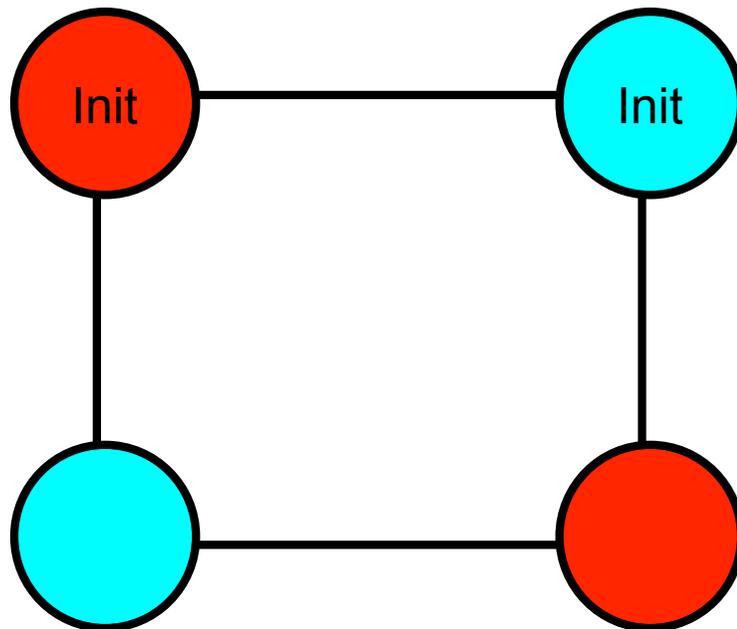
# Algorithme distribué

Synchrone :

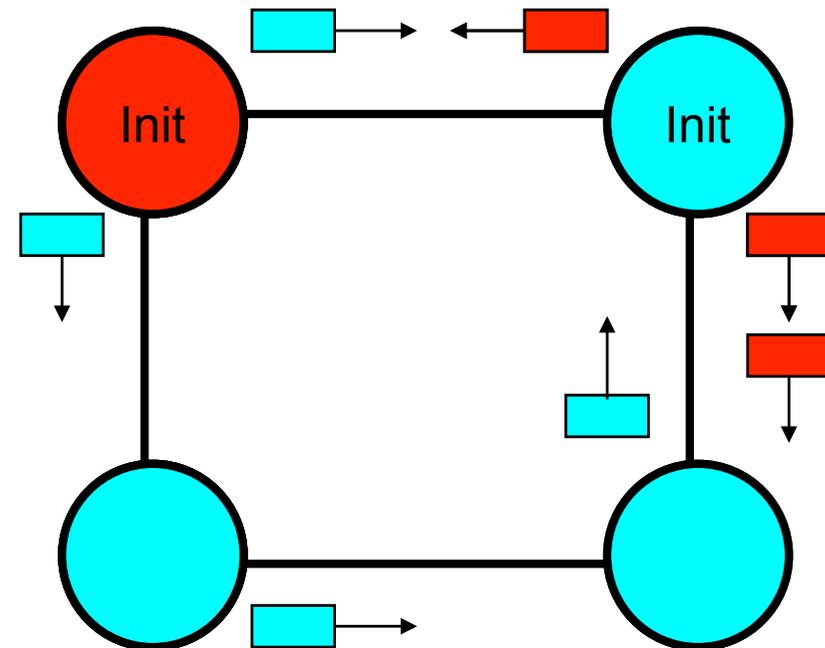


# Algorithme distribué

Synchrone :



Asynchrone :





## *Algorithme distribué asynchrone : conception et preuve de validité*

Modèle asynchrone :

- temps de calcul local « nul »
- temps d'acheminement des messages fini (non borné)
- plutôt difficile :  
**explosion combinatoire du nb de config.**  
**⇒ bons prédicats pour contourner ce pb ...**





# *Algorithme distribué asynchrone : complexité*

Modèle :

temps d'acheminement des messages borné ...

**PAR 1 !!!**



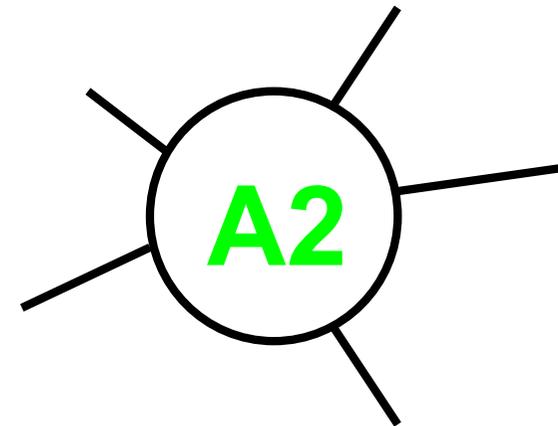
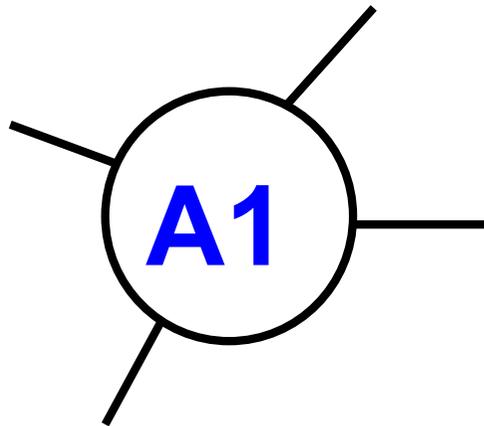
## *Algorithme distribué asynchrone : performances*

Quel degré de parallélisme ?

2 actions peuvent-elles se dérouler simultanément ?

Modèle :

temps de calcul local potentiellement infini !!!





# *Algorithme distribué asynchrone : comment poser les problèmes ?*

## Problème et spécification :

- Spécification : énoncé d'un pb permettant de prouver la validité de l'algo correspondant
- Relation entre pb et spécif :
  - Adéquation non prouvable
  - Pas d'unicité de la spécif
  - Équivalence de 2 spécif



## *Algorithme distribué asynchrone : comment poser les problèmes ?*

Équivalence de 2 spécifications :

Soit E un environnement quelconque...

–  $S1 \Rightarrow S2$  dans E ssi

  tout algo résolvant S1 dans E résout S2 dans E

–  $S1 \Leftrightarrow S2$  dans E ssi

$S1 \Rightarrow S2$  dans E et  $S2 \Rightarrow S1$  dans E



## *Algorithme distribué asynchrone : comment poser les problèmes ?*

Exclusion mutuelle : gestion de l'utilisation d'une ressource dans un système.

S1 (classique) :

Sûreté : **Jamais plus d'un processeur en SC,**

(Vivacité : **Tout processeur demandeur finit par entrer en SC.**)

S2 (moins classique) :

Sûreté : **Si il y a plus d'un proc en SC, alors aucun n'est demandeur,**

(Vivacité : **Tout processeur demandeur finit par entrer en SC.**)



## *Algorithme distribué asynchrone : comment poser les problèmes ?*

Structure séquentielle :

1. Demande de la SC (Section Critique)
2. Entrée en SC (et exécution)
3. Libération de la SC

Il est donc facile de montrer que :

$$S1 \Leftrightarrow S2$$

dans un « Syst. Dist. asynchrone sauf ».



## *Algorithme distribué asynchrone : comment poser les problèmes ?*

Dans le contexte de la stabilisation

- Configurations initiales quelconques
- **Auto-stabilisation** : respecte la spécif en 1 temps fini (Dijkstra 1974)
- **Stabilisation instantanée** : respecte toujours la spécif (Bui, Datta, Petit, Villain 1999)



## *Algorithme distribué asynchrone : comment poser les problèmes ?*

S1 et S2 ne sont plus équivalentes :

S2 permet d'obtenir des algos instantanément stabilisants,  
**S1 non !**

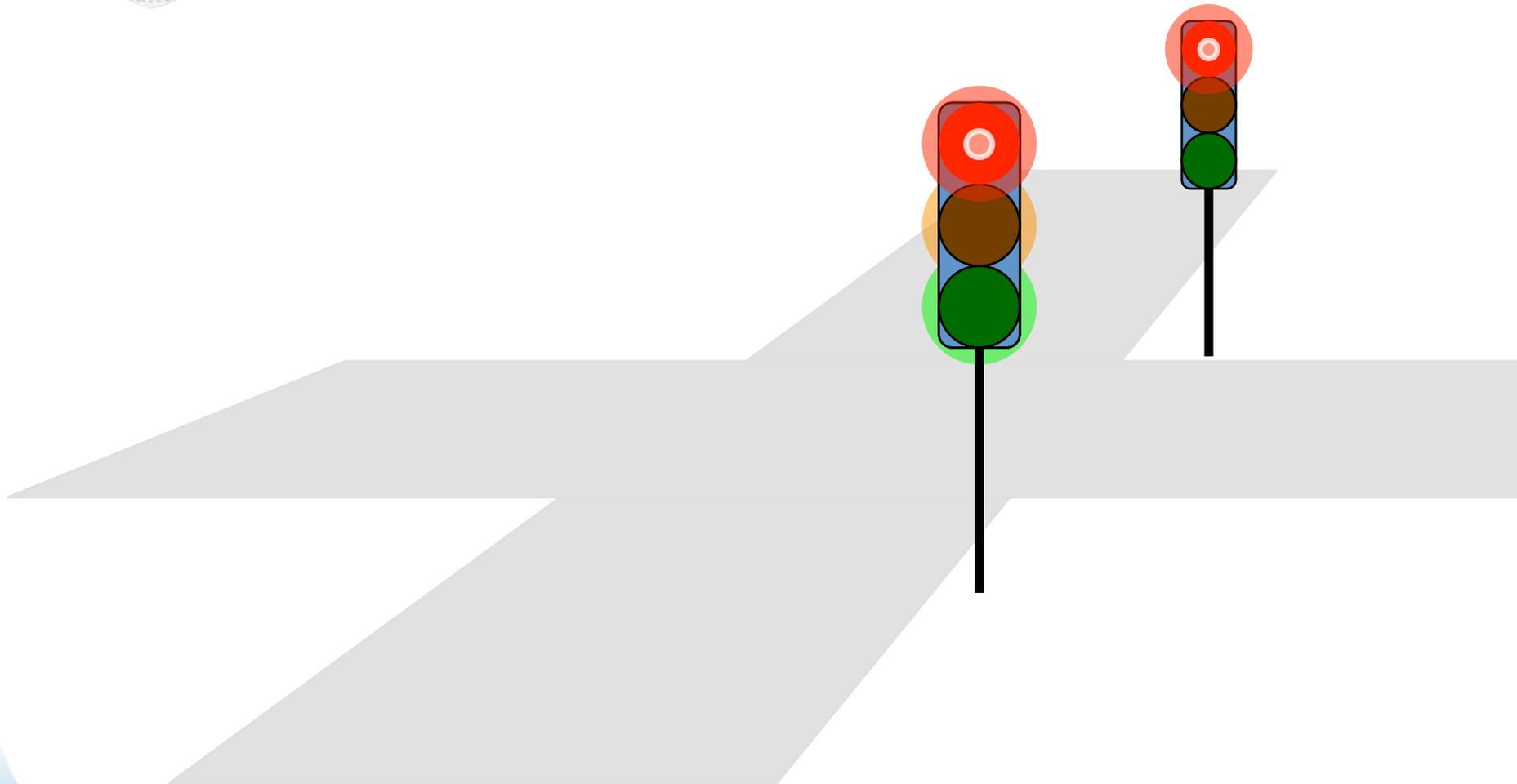
Remarques :

- Les algos instantanément stabilisants pour S2 sont auto-stabilisants pour S1
- S2 n'est pas la seule spécif à autoriser des algos instantanément stabilisants

Sûreté : Si un proc demandeur entre en SC, il exécute seul la SC,  
(Vivacité : Tout processeur demandeur finit par entrer en SC.)

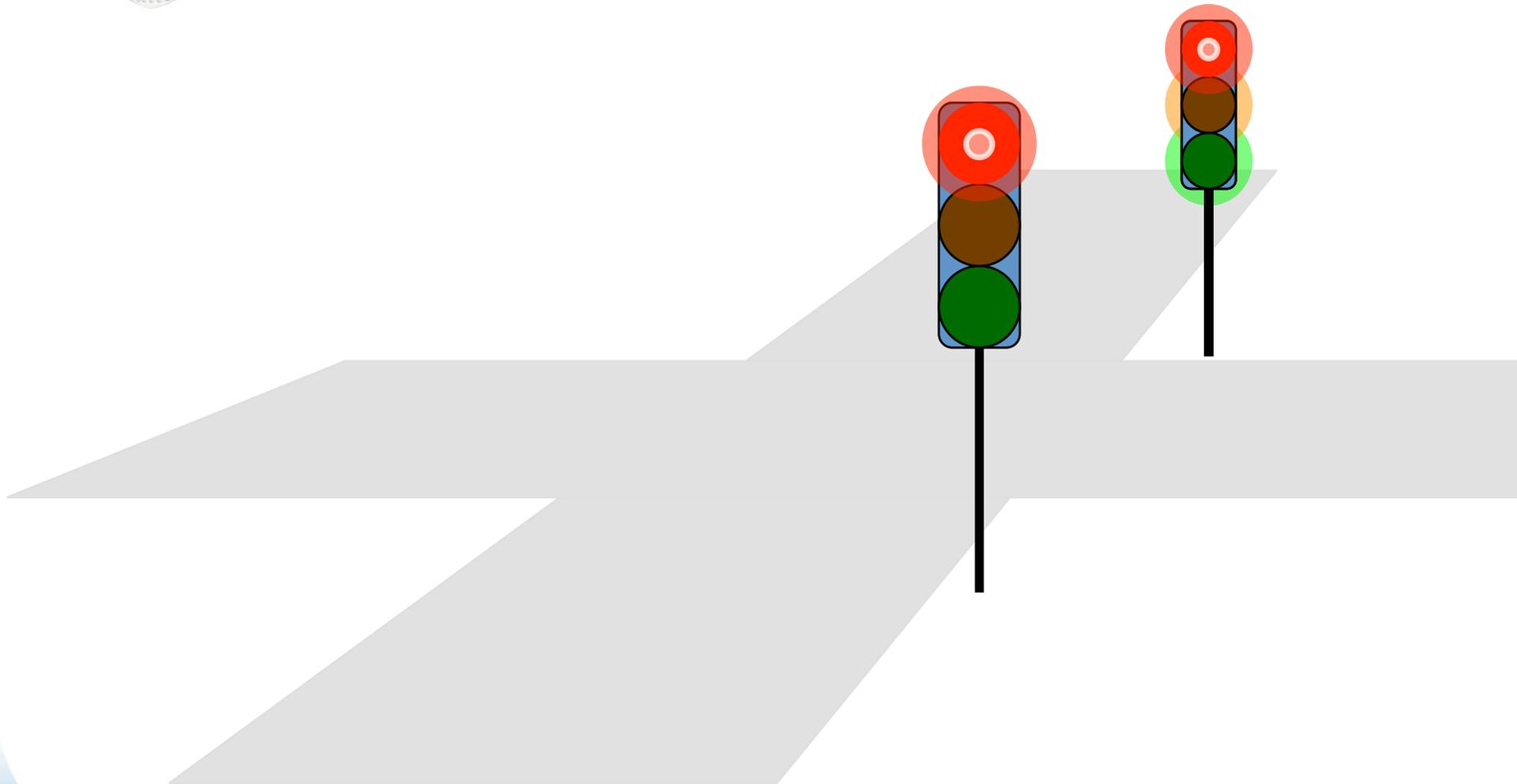


## *Exemple des feux tricolores*



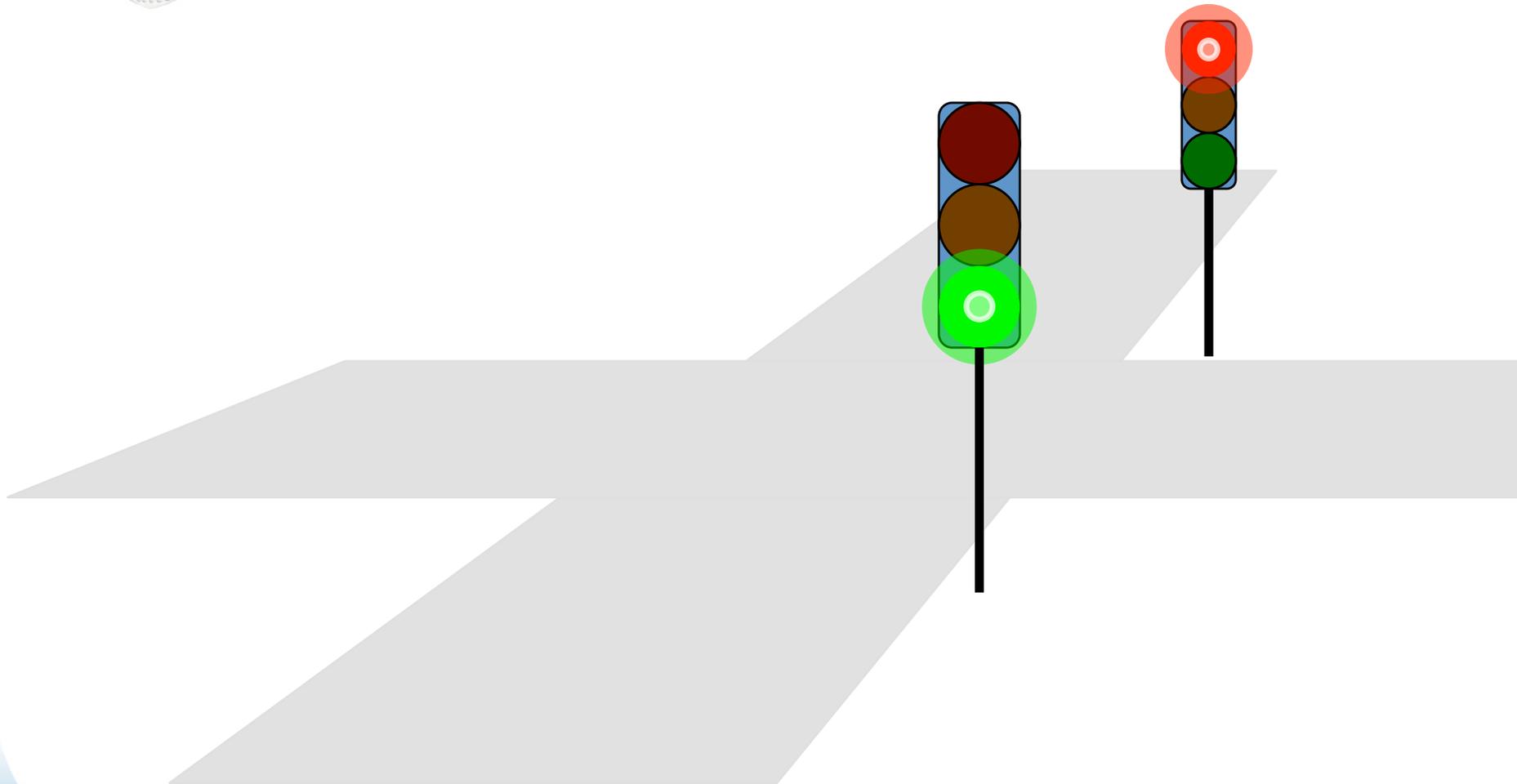


## *Exemple des feux tricolores*



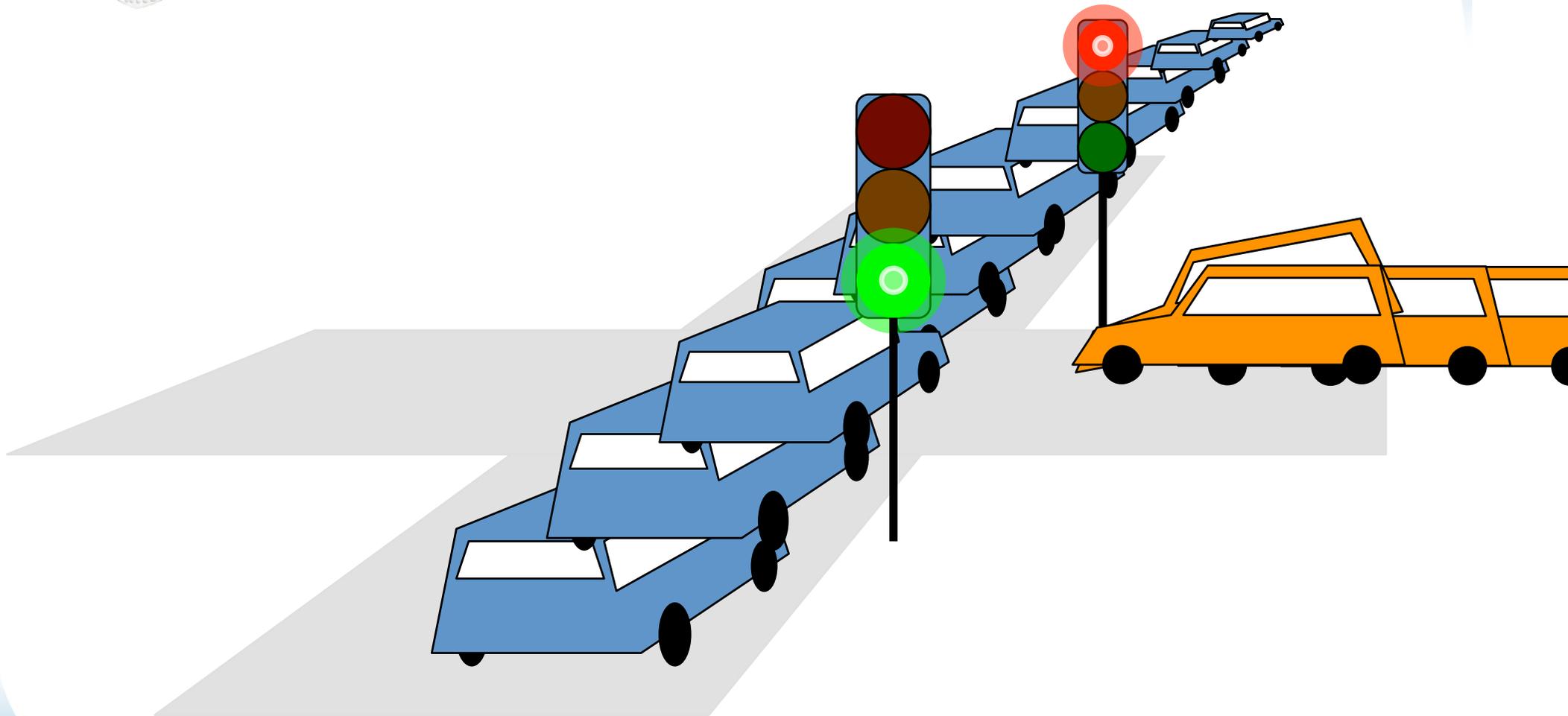


## *Exemple des feux tricolores*



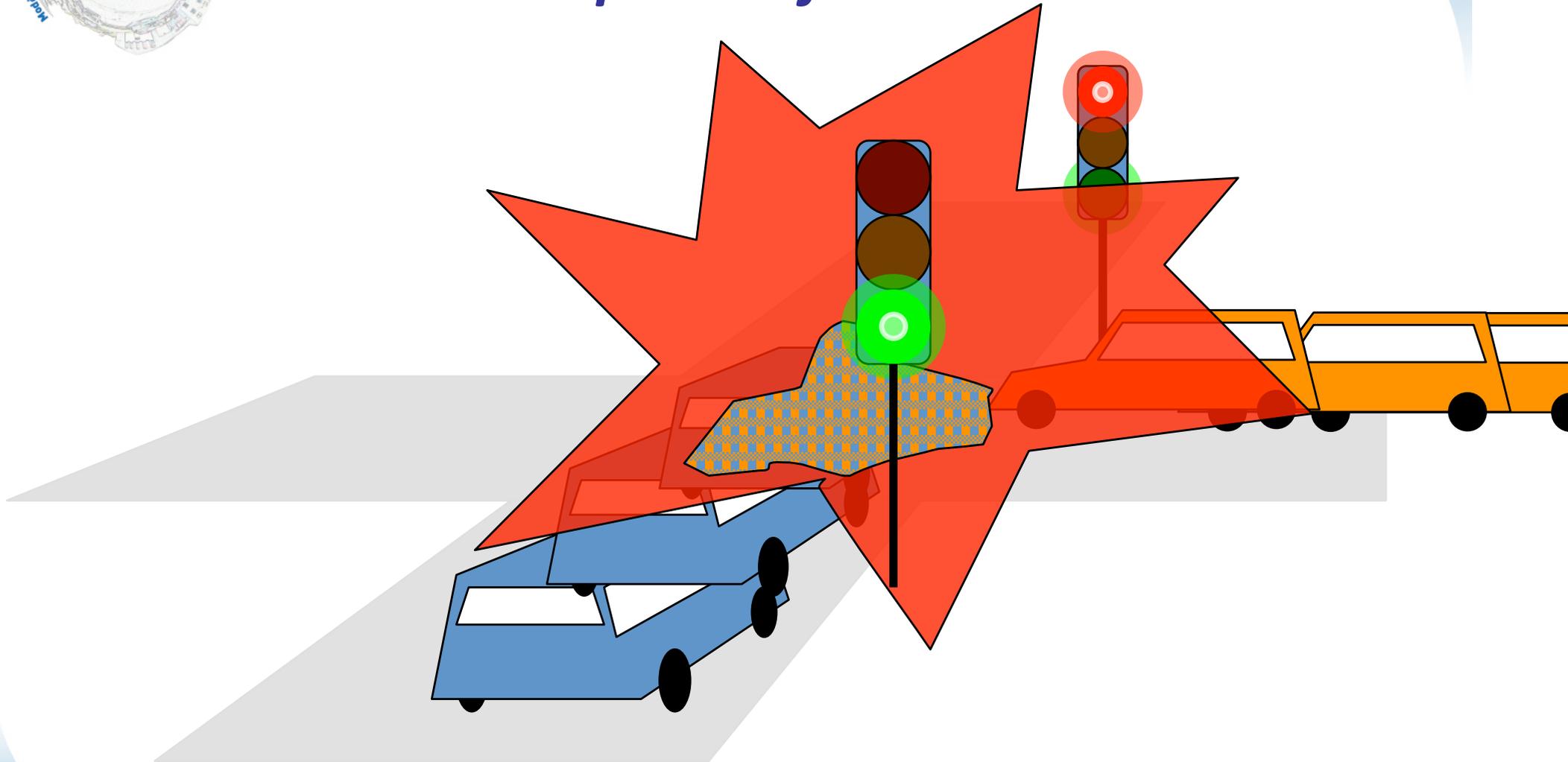


## *Exemple des feux tricolores*



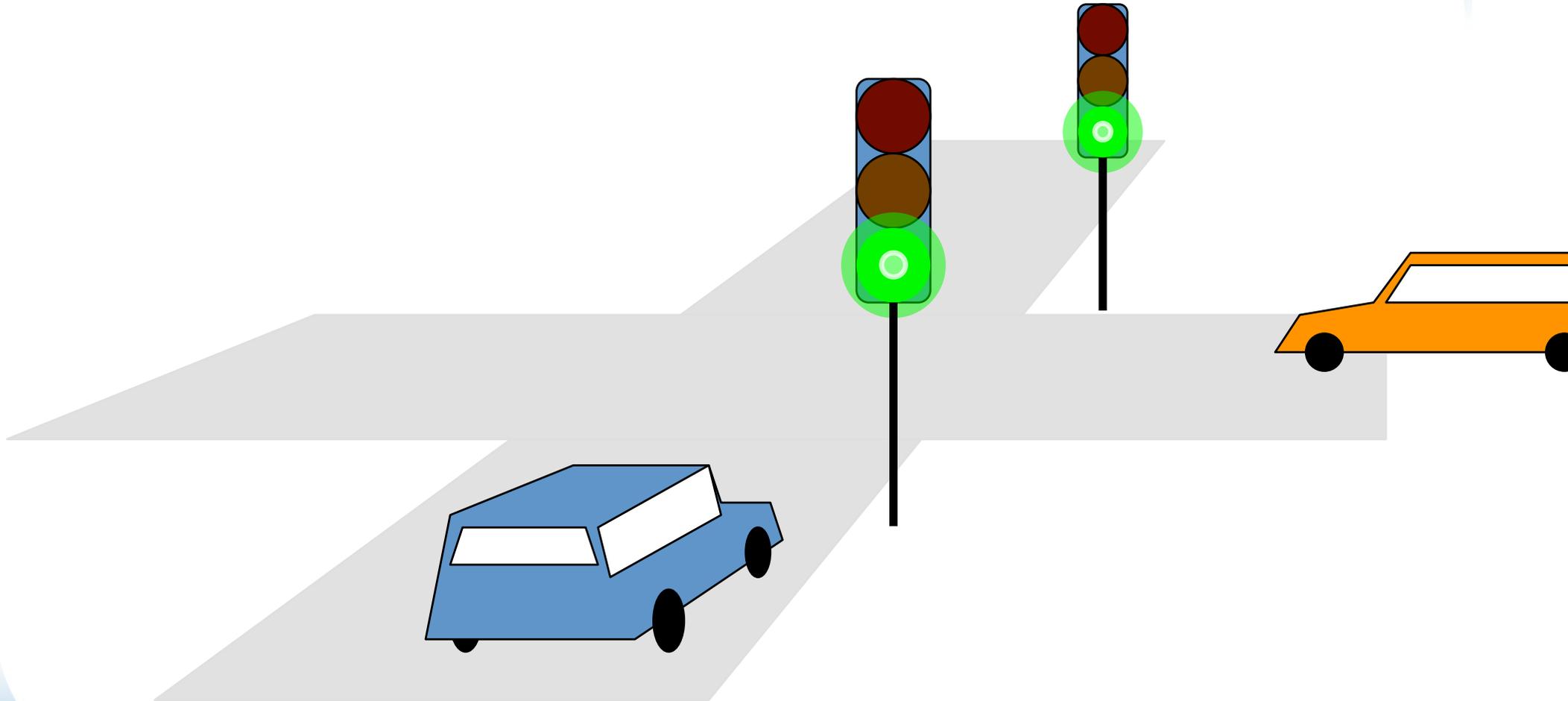


## *Exemple des feux tricolores*



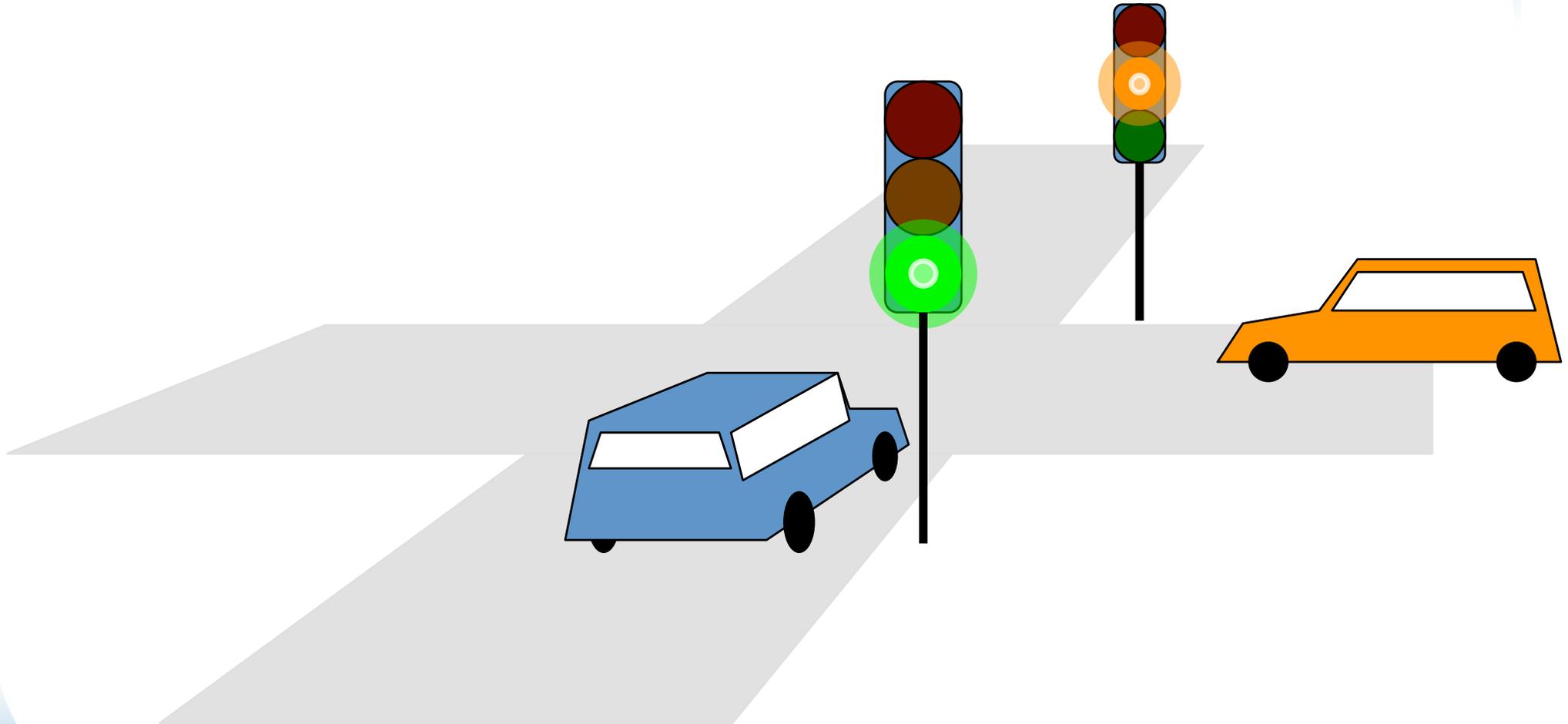


## *Exemple des feux tricolores*



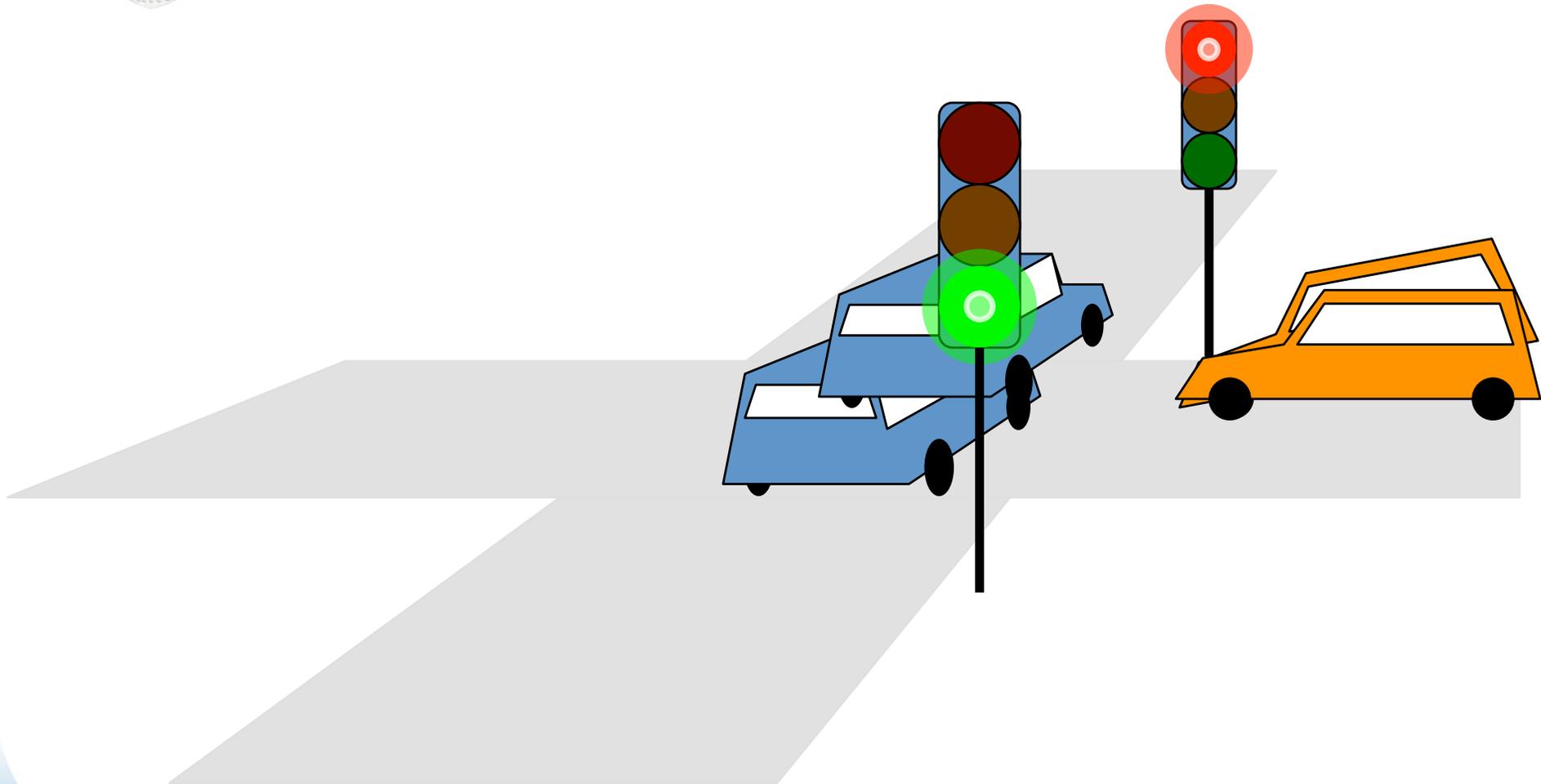


## *Exemple des feux tricolores*





## *Exemple des feux tricolores*





## Conclusion

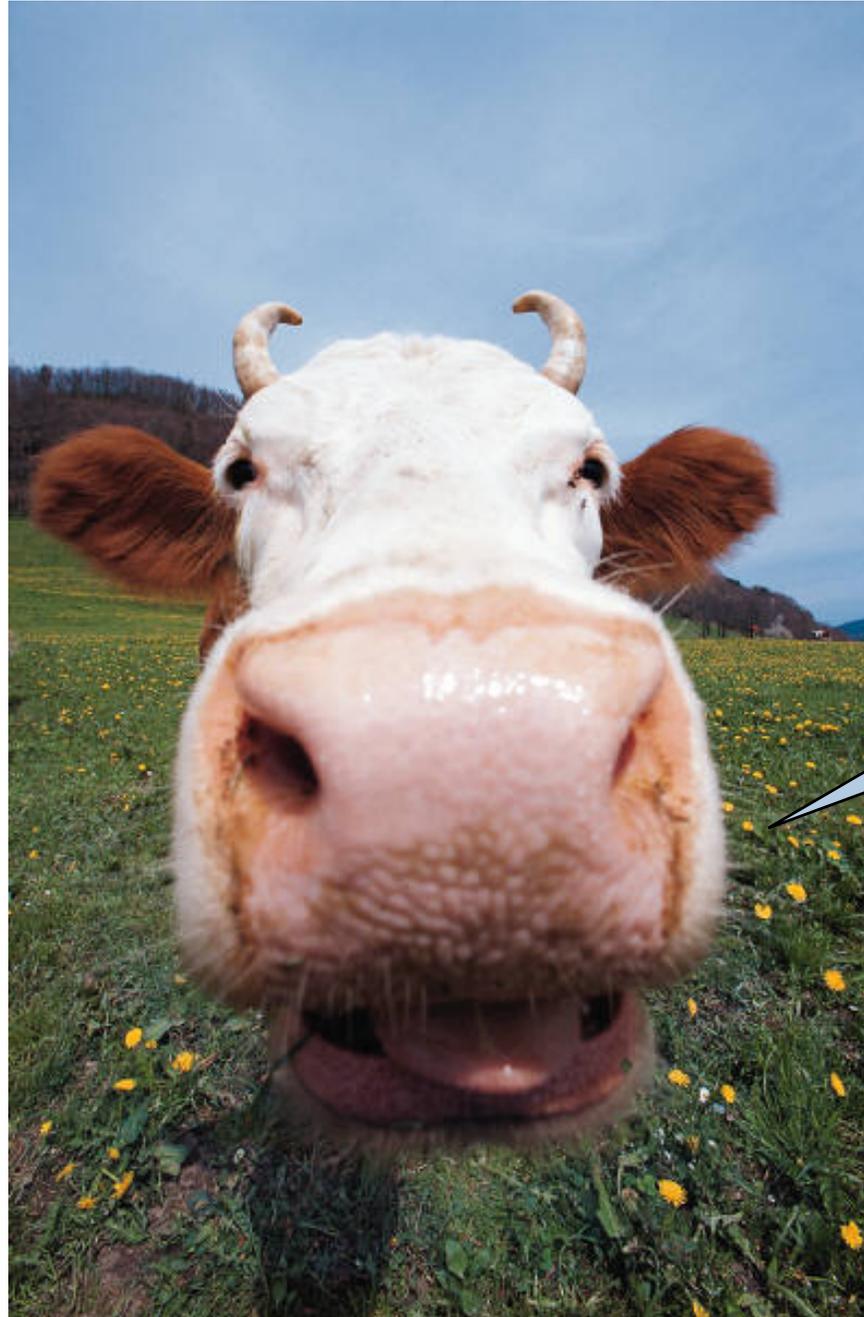
- Il est grand temps de réfléchir au problème posé et à son contexte, car beaucoup des étudiants sont trop pressés.





## *Conclusion*

- Il est grand temps de réfléchir au problème posé et à son contexte !! (à l'attention des étudiants trop pressés ...)
- Côté recherche :
  - y-a-t'il encore de bonnes questions ?
  - y-a-t'il de bonnes réponses à ces questions ?



C'est fini !