**Travail à rendre avant le 25/01/2021**

**Algorithmes Distribués**

**Problème**

Un parking possède p portes par lesquelles des voitures peuvent entrer ou sortir. Le parking ne doit jamais contenir plus de N voitures. À chaque porte se trouve un gardien ; les gardiens communiquent en s’envoyant des messages (de façon asynchrone, par exemple à l’aide de pigeons voyageurs supposés fiables) ; ils doivent faire en sorte que la capacité du parking ne soit jamais dépassée. Il faut par ailleurs que, s’il y a régulièrement des sorties, toute voiture qui attend pour entrer par la porte de son choix y parvienne en un temps fini. Décrire un algorithme réparti, qui sera appliqué par chacun des gardiens, afin résoudre le problème. Cet algorithme utilisera des principes présentés en cours. Le nombre de messages échangés entre deux entrées de voitures devra être borné.

Mises en gardes :

* Une idée (incorrecte) serait de partager initialement les N places du parking en p lots de N/p places, à la responsabilité de chaque gardien. Dans ce cas le gardien peut laisser rentrer N/p voitures, plus autant de voitures qu’il a laissé sortir. Cependant, avec ce schéma si toutes les voiture sortent parla même porte, cette dernière porte devient la seule entrée possible et les voitures qui cherchent à rentrer par une autre porte y restent bloquées (ce qu’on ne veut pas)
* Une autre idée (tout aussi incorrecte) serait d’imaginer que les gardiens tentent de garder le compte des places libres en diffusant (aux autres gardiens) des messages “entrée” ou “sortie” à chaque fois qu’une voiture entre ou sort par leur porte. Le problème est que ces diffusions ne sont pas délivrées instantanément et qu’elles peuvent aussi se croiser. Cela devient critique lorsqu’il n’y a plus beaucoup de places libres. Par exemple s’il ne reste plus qu’une place libre, deux gardiens pourraient simultanément laisser rentrer chacun une voiture.