

Introduction à la communication

Module

Systèmes Communicants et Synchronisés

Master Informatique 1ère année

L. Philippe & V. Felea

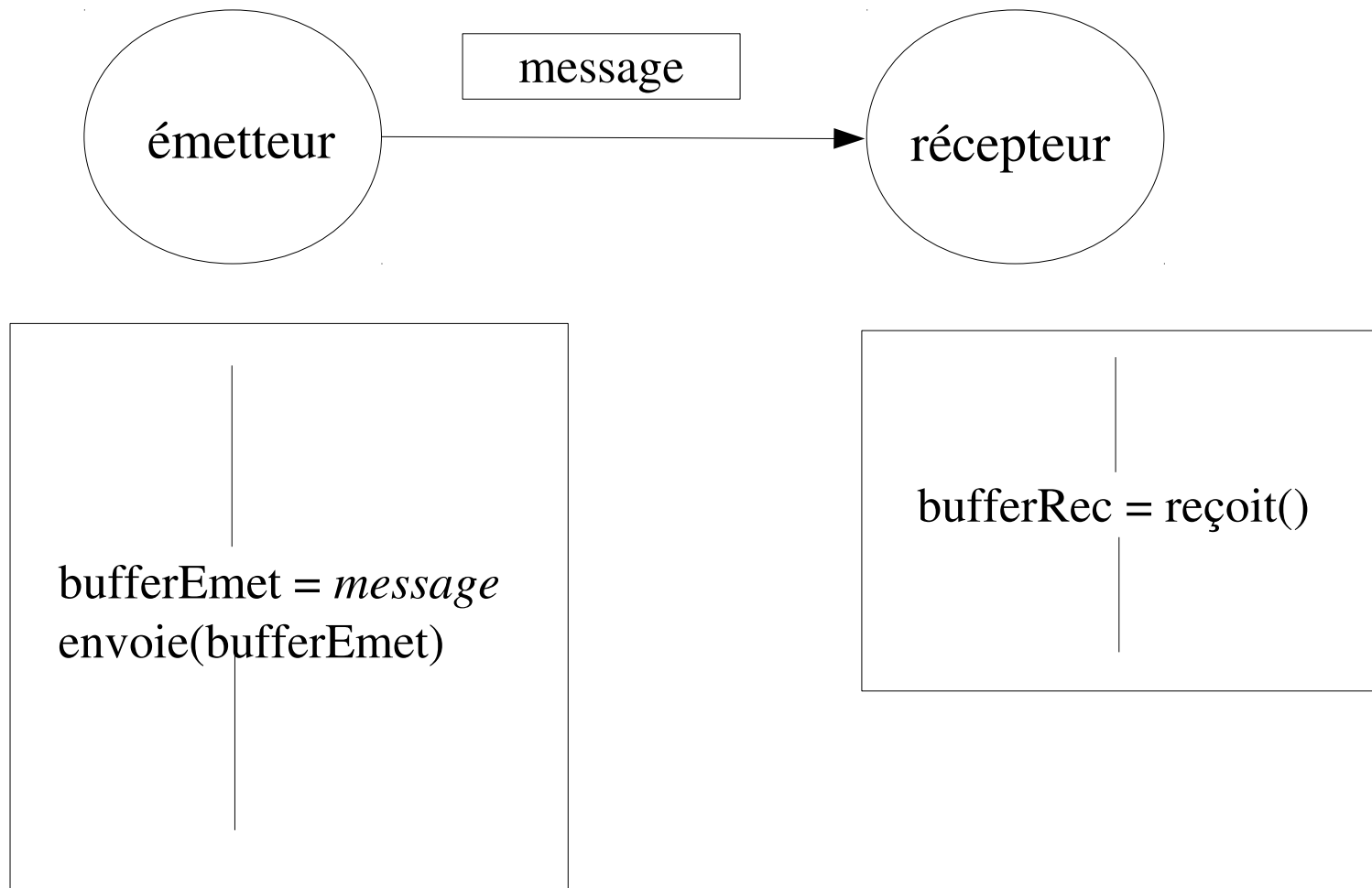
Systèmes communicants

- programmation traditionnelle
 - ▶ exécution d'un programme sur des données
 - ▶ données « centralisées » (disponibles localement)
 - ▶ fin du programme = résultat
- programmes communicants
 - ▶ exécution simultanée de plusieurs programmes
 - ▶ données « distribuées » (indisponibles localement)
 - ▶ serveur = programme infini
 - ▶ communication = ensemble des échanges entre programmes

Communication

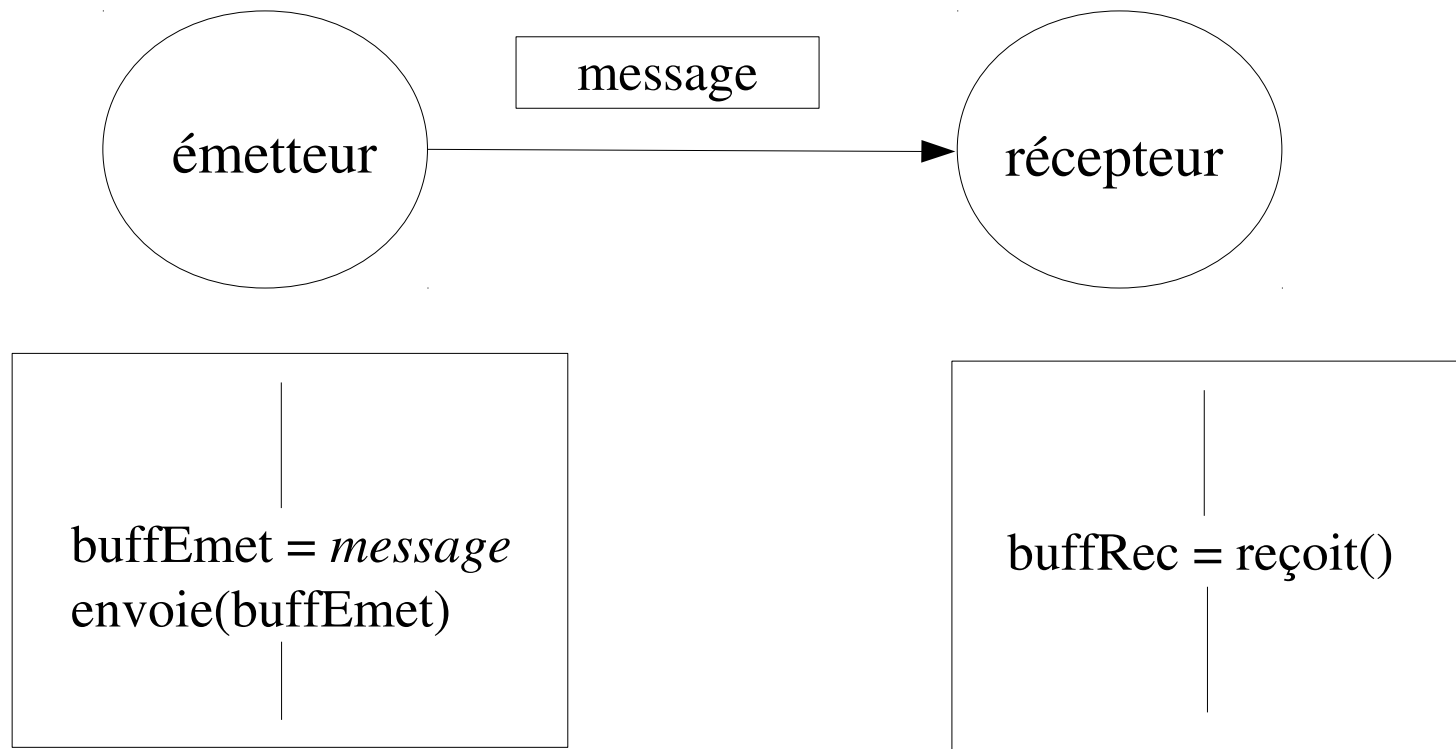
- nombre de participants : 2 à n , souvent 2
- échange entre les participants
 - ▶ généralement, des données
 - ▶ sans données = contrôle
- rôles
 - ▶ émetteur
 - ▶ récepteur
- support de communication dans les programmes
 - ▶ fonctions (système/bibliothèques)
 - ◆ envoi (émetteur)
 - ◆ réception (récepteur)
 - ▶ message

Message



Aspects de la communication (1)

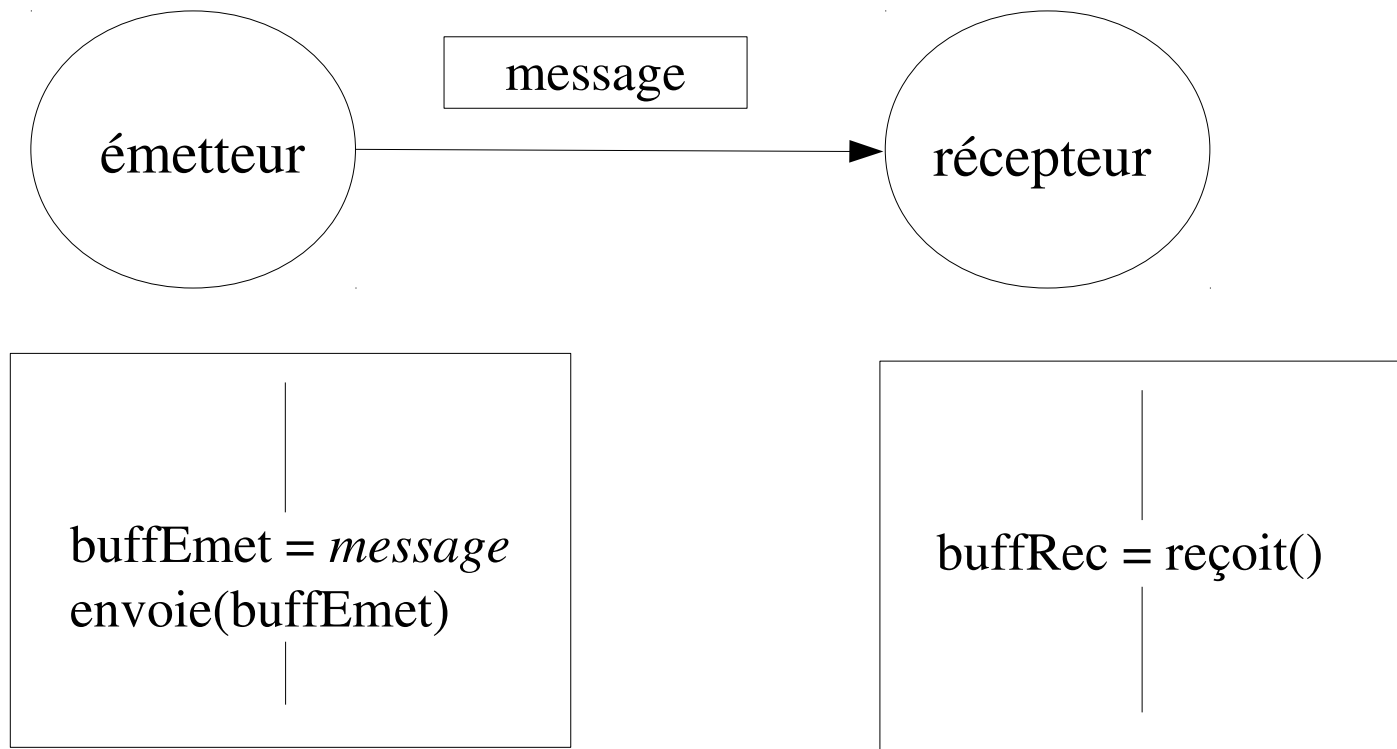
- cas 1 : récepteur actif (prêt) ?



- problème : réception aléatoire (message éventuellement perdu)

Aspects de la communication (2)

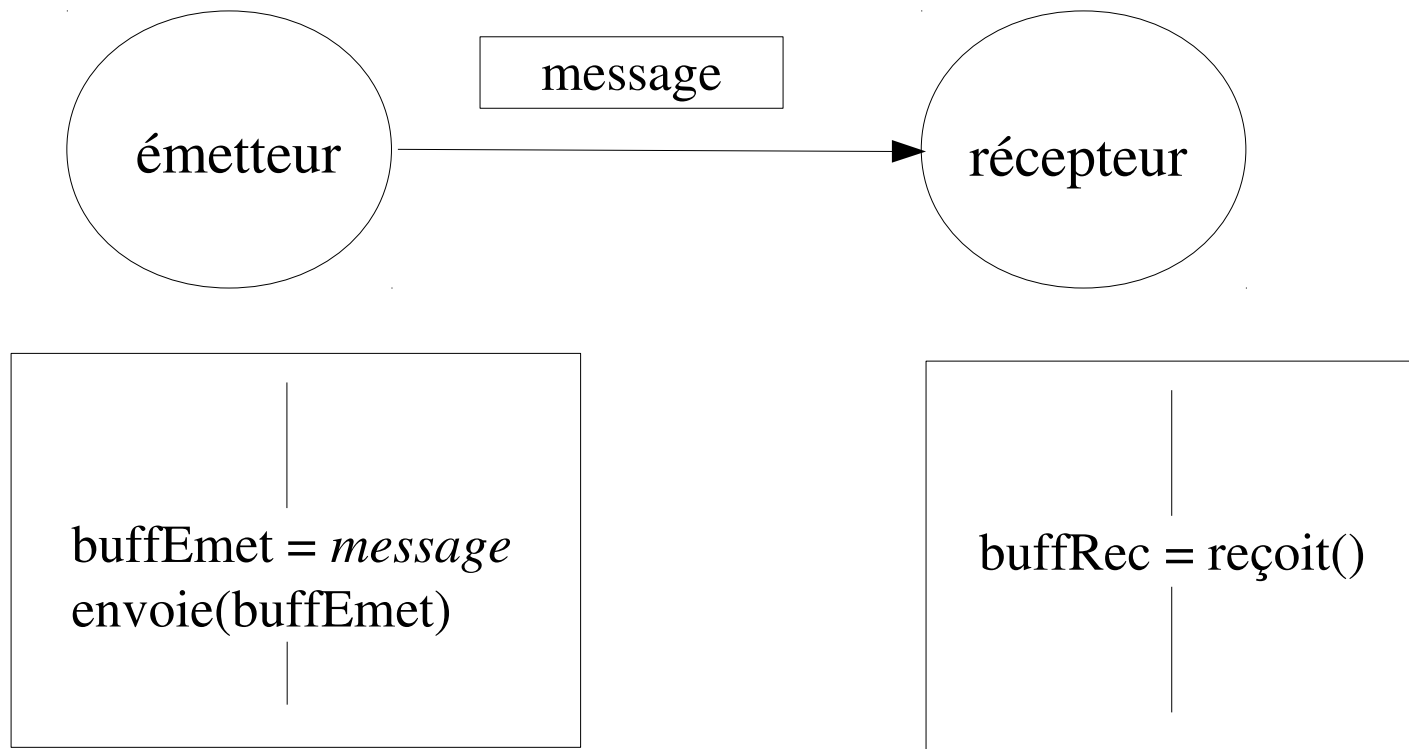
- cas 2 : récepteur actif, en attente (permanente)



- problème : attente infinie

Aspects de la communication (3)

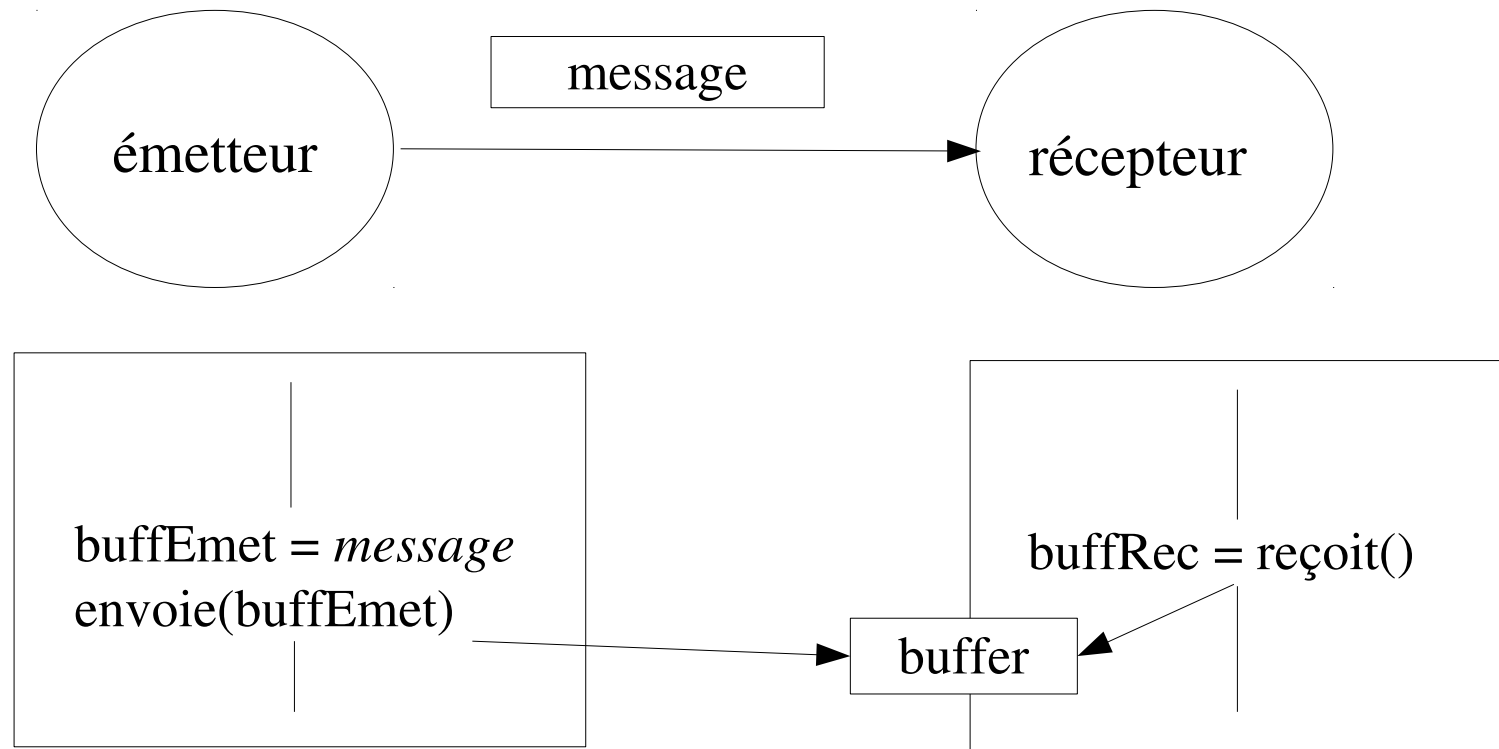
- cas 3 : récepteur non actif, cherche le message à la demande



- problème : demandes répétées, choix du destinataire

Aspects de la communication (4)

- cas 4 : récepteur non actif, mais éviter la perte de messages



- problème : saturation buffer, validation réception
- modèle le plus courant (sockets)

Questions

- entente temporelle entre l'envoi et la réception ?
- quand la réutilisation de la donnée envoyée est-elle possible ?
- d'autres actions peuvent se dérouler *en même temps* que la communication ?
- les communications sont bufferisées ? chez l'émetteur / chez le récepteur ?

Aspects de la communication

- participants

- ▶ émetteur : qui est le destinataire (identification),
adresse du message à envoyer
- ▶ récepteur : qui est l'émetteur (identification), adresse
où stocker le message reçu

- déroulement de la communication

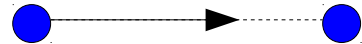
- ▶ canal (direction)
- ▶ synchronisation (temps)
- ▶ bloquante/non-bloquante (contrôle)

- protocole : règles qui régissent la communication

Canal de communication

- simple (simplex) : une seule direction

- ▶ exemples : radio, télévision



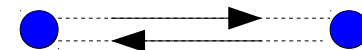
- semi-duplex (half-duplex) : deux directions, mais non simultanément

- ▶ exemple : talkie-walkie



- duplex (full-duplex) : deux directions simultanément

- ▶ exemple : téléphone

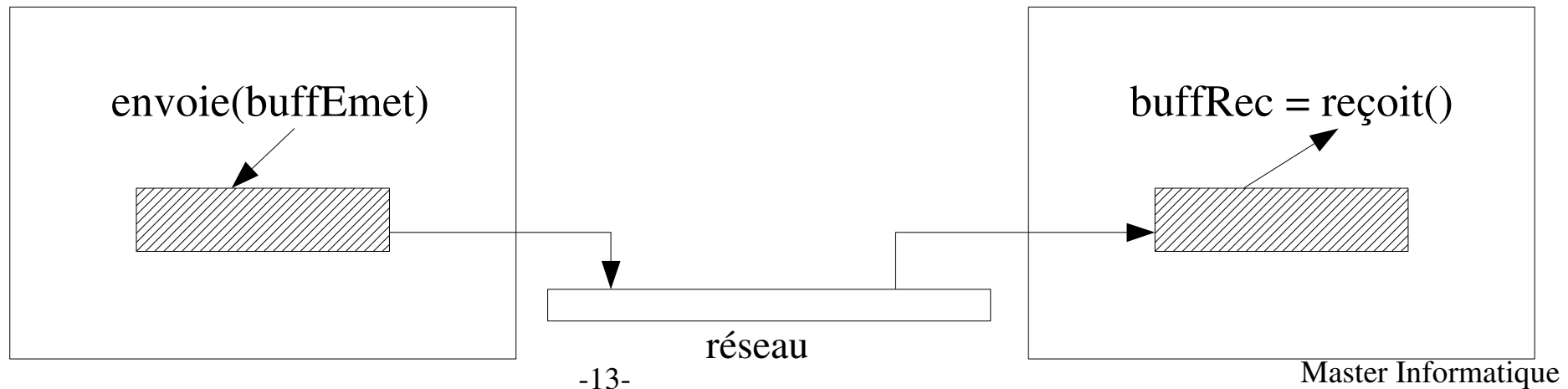


Synchronisation des communications (1)

- synchrone / asynchrone
 - ▶ définit une action dont le traitement est exécuté / n'est pas exécuté au moment où on l'appelle
- envoi/réception synchrones
 - ▶ attente
- envoi/réception asynchrones
 - ▶ émetteur et récepteur ne s'attendent pas (ils ne sont pas nécessairement actifs au même moment)
- l'émetteur et le récepteur peuvent ne pas être les deux synchrones / asynchrones

Synchronisation des communications (2)

- où vont les données envoyées ?
- bufférisation (chez l'émetteur, chez le récepteur, caractéristique du lien)
 - ▶ capacité
 - ♦ 0 : aucun message en attente (besoin de synchronisation)
 - ♦ limitée : émetteur peut être retardé si buffer plein
 - ♦ illimitée : émetteur jamais retardé
 - ▶ buffers gérés par le programmeur ou internes au système



Communications bloquantes / non bloquantes

- fait référence au flux de contrôle
- envoi/réception bloquants : la ressource est disponible en retour du traitement (définit une action qui ne termine pas tant que son traitement n'est pas terminé)
- réception non-bloquante simple : un paramètre de retour indique si l'information est disponible
- envoi/réception non-bloquants généraux : retour de la procédure sans garantir que la donnée ait été envoyée/reçue
 - ▶ l'utilisateur ne peut pas réutiliser l'espace mémoire associé au risque de changer ce qui sera envoyé
 - ▶ il faut donc pouvoir tester/attendre la libération (si envoi) ou la réception effective de la donnée