1. Transmission

```
from random import randint

def numerote(list):
    # ajoute un numéro aux éléments de la liste
    num = []
    for i in range(len(list)):
        num.append(str(i+1) + " " + list[i])
    return num

def permute(lst):
    # permute deux éléments de la liste
    rang = randint(0,len(lst)-2)
    lst[rang], lst[rang+1] = lst[rang+1], lst[rang]
    return lst

def supprime(lst):
    # supprime un élément de la liste
    del lst[randint(0,len(lst)-1)]
    return lst

recette = ["Emincer 100 g de truffe", "Casser 8 oeufs", "Battre les oeufs",
"Incorporer les truffes", "Faire cuire", "Servir"]

print("--- Recette de l'omelette aux truffes ---")
for instructions in recette:
    print(instructions)

print("\n--> Envoi de la recette...\n")

recette_recue = supprime(permute(numerote(recette)))

print("--- Recette de l'omelette aux truffes ---")
for instructions in recette_recue:
    print(instructions)
```

```
--- Recette de l'omelette aux truffes ---
Emincer 100 g de truffe
Casser 8 oeufs
Battre les oeufs
Incorporer les truffes
Faire cuire
Servir

--> Envoi de la recette...

--- Recette de l'omelette aux truffes ---
1 Emincer 100 g de truffe
2 Casser 8 oeufs
3 Battre les oeufs
6 Servir
5 Faire cuire
```

2. Simulation du protocole « envoyer et attendre »

```
from random import randint

time_out = 10
time_lost = 5

recette = ["1 Emincer 100 g de truffe", "2 Casser 8 oeufs", "3 Battre les
oeufs", "4 Incorporer les truffes", "5 Faire cuire", "6 Servir"]

time = 0
for i in range(len(recette)):
    t_emission = randint(1,5)
    t = randint(1,6)
    time = time + t + 10*randint(0,1) + t_emission
    while t > time_lost:
        print(recette[i], "( reçu à", time, "- temps acc. de réception", t,
")")
        t = randint(1,6)
        time = time + t_emission + t
    print(recette[i], "( reçu à", time, "- temps acc. de réception", t, ")")
```

1 Emincer 100 g de truffe (reçu à 3 - temps acc. de réception 4)
2 Casser 8 oeufs (reçu à 12 - temps acc. de réception 3)
3 Battre les oeufs (reçu à 30 - temps acc. de réception 2)
4 Incorporer les truffes (reçu à 35 - temps acc. de réception 1)
5 Faire cuire (reçu à 40 - temps acc. de réception 6)
5 Faire cuire (reçu à 48 - temps acc. de réception 3)
6 Servir (reçu à 53 - temps acc. de réception 5)

3. Simulation du protocole « bit alterné »

1 Emincer 100 g de truffe (reçu à 3 - temps acc. de réception 4) - **bit = 0**
2 Casser 8 oeufs (reçu à 12 - temps acc. de réception 3) - **bit = 1**
3 Battre les oeufs (reçu à 30 - temps acc. de réception 2) - **bit = 0**
4 Incorporer les truffes (reçu à 35 - temps acc. de réception 1) - **bit = 1**
5 Faire cuire (reçu à 40 - temps acc. de réception 6) - **bit = 0**
*5 Faire cuire (reçu à 48 - temps acc. de réception 3) - **bit = 0***
6 Servir (reçu à 53 - temps acc. de réception 5) - **bit = 1**

[Plus tard : Détecter les corruptions avec le « bit de parité »]