Réseaux 2

I. Membres du groupe

- Corentin JEZEQUEL
- Arnaud CHESNAY
- Alexis MERCERON
- Pierre PETILLON

II. Rappel du cahier des charges

II. A. La problématique

Comment faire circuler les paquets sur un réseau en anneau ?

II. B. Les Pré-requis

Chaque hôte est muni d'une « bouche » (sortie) et d'une « oreille » (entrée). Un seul paquet circule sur l'anneau. Le réseau est de taille modeste, c'est à dire 256 machines. L'identification se fais sur 8 bits. L'identifiant est défini à la mains par l'attribution du plus faible indice disponible.
Exemple, la première machine obtiendra l'identifiant 0, la deuxième 1.
Si la deuxième machine se déconnecte, la prochaine prendra l'identifiant 1 étant le plus petit disponible.

II. C. Les paquets

II. C. a. Description des paquets

Type de paquet (2 bits)	Machine source (8bits)	Machine destination (8 bits)	Données (1024 bits)

Type de paquet :
Le type de paquet est codé sur 2 bits car nous avons 4 types de paquets différents.

- « 0 » : libérer le canal
- « 1 » : prendre le canal
- « 2 » : paquet de données
- « 3 » : paquet d'accusé de réception

Machine source/destination :
représente la machine qui envoie/reçoit le paquet, dans notre réseau nous avons au maximum 256 machines il est donc nécessaire de coder cela sur 8 bits.

Données : chaque message contient au maximum 128 caractères qui sont tous codés sur 8 bits

Taille totale du paquet: 1042 bits

II.C.b.Exemple de paquets

Paquet de libération du canal :

00	ID_Machine source	ID_Machine source	000000...000000

Paquet de réservation du canal :

01	ID_Machine source	ID_Machine source	000000...000000

Paquet de données :

10	ID_Machine source	ID_Machine destination	Codage des données

Paquet d'accusé de réception :

11	ID_Machine destination	ID_Machine source	000000 …. 000000

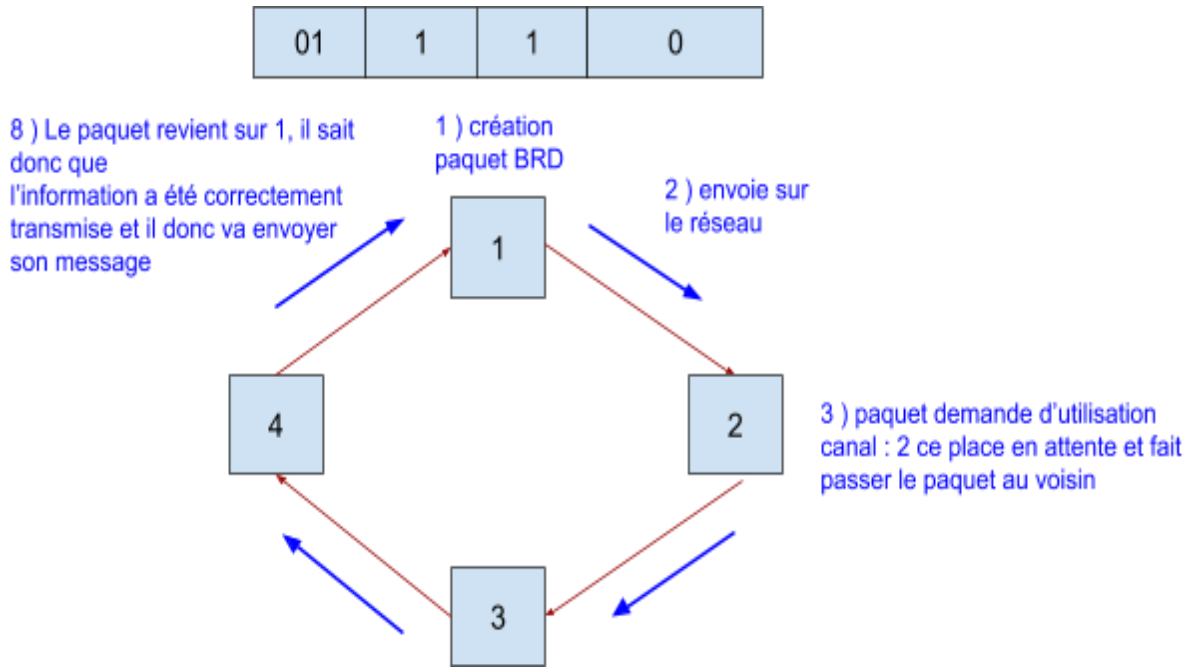
ID_Machine destination : la machine qui était sensé recevoir le paquet de données
ID_Machine source : la machine émettrice du paquet de données
La source et la destination sont inversés dans ce paquet afin de constituer le paquet d'accusé de réception.

III. Traitement logique du paquet

Protocole d'envoi et de réception d'un paquet de données :

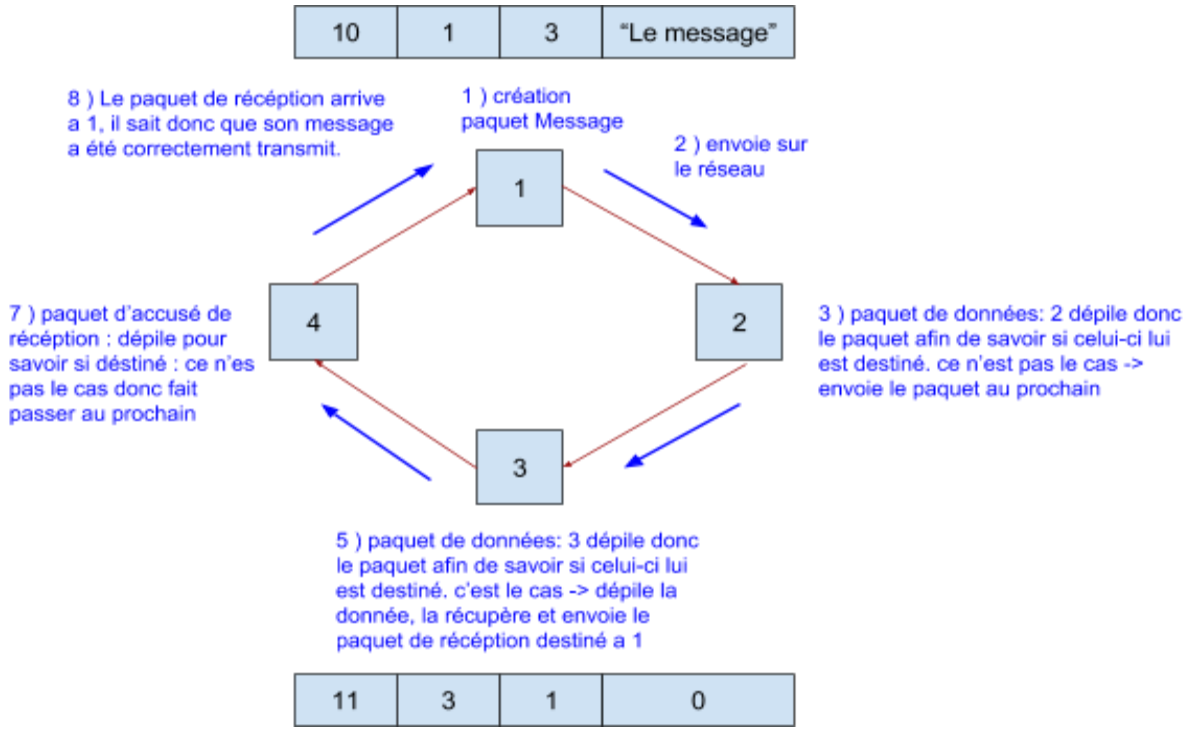
1) La machine source envoie un paquet de type "réservation de canal" (paquet en broadcast de type 01).
    a) Soit il y a un conflit de réservation, les deux machines sources attendent un temps aléatoire puis retentent de réserver le canal. Et les machines ayant reçu deux paquets de réservation du canal et n'étant pas impliqués dans le conflit stockent le fait qu'il y a un conflit dans celui ci et deviennent muettes jusqu'à la prochaine libération.
    b) Soit il n'y a pas de conflit, toutes les machines enregistrent le fait que le canal est occupé.

Schéma phase de réservation du canal (par la machine 1) :



2) La machine source prépare le “paquet de données” (type 10) en encapsulant le type, la machine machine source, la machine destination et les données.
3) La machine source envoie le “paquet de données” et attend un “paquet d’accusé de réception”. En cas de retour du paquet de données, la machine source comprend que la machine destination n’existe pas ou est déconnectée du réseau.
4) Dans le cas où le successeur (la machine suivante sur le canal) reçoit le paquet, décapsule le type de paquet, l’adresse de la machine source et de la machine de destination, il compare l’adresse destination et la sienne :
    a) Soit le paquet lui est destiné, la machine décapsule les données et les lit. Elle prépare le paquet d’accusé de réception et l’envoie à la machine source.
    b) Soit le paquet ne lui est pas destiné, la machine réencapsule le paquet de données tel qu’elle l’a reçu et le renvoi au suivant.

Schéma d'envoi de message (par la machine 1 a la machine 3) :



5) Une fois que l'émetteur a reçu son accusé de réception, il envoie un paquet de libération du canal. Les autres machines stockent le fait que le canal est de nouveau disponible. La machine source attend alors un certain temps, afin de laisser aux autres machines la prise de parole, avant de pouvoir demander à nouveau d'envoyer un paquet
6) Et donc le processus peut recommencer avec la machine ou une machine différente.

IV. Algorithme de traitement

```
// Couche 1 : Type de données

def couche1():
      typePaquet = Dépile(Tampon)
      Si typePaquet == "00" // Paquet de libération de canal
            canalDisponible = True
            couche2(typePaquet)
      Sinon si typePaquet == "01" // Paquet de reservation de canal
            canalDisponible = False
            couche2(typePaquet)
      Sinon si typePaquet == "10" // Paquet de données
            couche2(typePaquet)
      Sinon si typePaquet == "11" // Paquet accusé de reception
            couche2(typePaquet)

// Couche 2 : Machine Source

def couche2(typePaquet):
      ID_Machine_Source = Dépile(Tampon)
      Si typePaquet == "00" ou typePaquet == "01"
            Si ID_Machine_Source != ID_Machine_Courante
                  Tampon.Empile(ID_Machine_Source)
                  Tampon.Empile(typePaquet)
                  envoyerPaquet()
      Sinon Si typePaquet == "10"
            Si ID_Machine_Source == ID_Machine_Courante
                  Exception("Paquet revenu à l'envoyeur")
            Sinon
                  couche3(typePaquet,ID_Machine_Source)
      Sinon Si typePaquet == "11"
            Si ID_Machine_Source == ID_Machine_Courante
                  Exception("Machine emettrice à quitté le reseau")
            Sinon
                  couche3(typePaquet,ID_Machine_Source)


// Couche 3 : Machine Destination

def couche3(typePaquet,ID_Machine_Source):
      ID_Machine_Destination = Depile(Tampon)
      Si ID_Machine_Destination != ID_Machine_Courante
            Tampon.Empile(ID_Machine_Destination)
            Tampon.Empile(ID_Machine_Source)
            Tampon.Empile(typePaquet)
            envoyerPaquet()
      Sinon
            couche4(typePaquet,ID_Machine_Source,ID_Machine_Destination)
```

```
// Couche 4 : Message

def couche4(typePaquet,ID_Machine_Source,ID_Machine_Destination):
      Message = Depile(Tampon)
      afficher(Message)
      Si typePaquet == "10"
            typePaquet = "11"
            ID_Machine_Source = ID_Machine_Destination
            ID_Machine_Destination = ID_Machine_Source
            Message = "0000...000"
            Tampon.Empile(Message)
            Tampon.Empile(ID_Machine_Destination)
            Tampon.Empile(ID_Machine_Source)
            Tampon.Empile(typePaquet)
            envoyerPaquet()
      Sinon si typePaquet == "11"
            messageRecu = True
            typePaquet = "00"
            ID_Machine_Source = ID_Machine_Destination
            Message = "0000...000"
            Tampon.Empile(Message)
            Tampon.Empile(ID_Machine_Destination)
            Tampon.Empile(ID_Machine_Source)
            Tampon.Empile(typePaquet)
            envoyerPaquet()

def traiterPaquet():
      couche1()

// Algorithme Principal

initialisation()
tant que True
      recevoirPaquet()
      traiterPaquet()
fin tant que
```

V. Multiplexage

Pour pouvoir multiplexer les paquets, nous avons choisi de créer un nouveau paquet avec un champ type, et avec des champs pour stocker les différents paquets. On limitera le nombre de slots pour stocker les paquets à 4. Pour le type, nous aurons besoin d'un bit supplémentaire pour pouvoir coder le nouveau type, on a donc désormais 3 bits pour le type.
La taille de ce paquet est donc de 3 + 4 * 1043 = 4175 bits.

Description d'un paquet multiplexé:

Type de paquet multiplexé (3 bits)	Slot n°1 (1043 bits)	Slot n°2 (1043 bits)	Slot n°3 (1043 bits)	Slot n°4 (1043 bits)

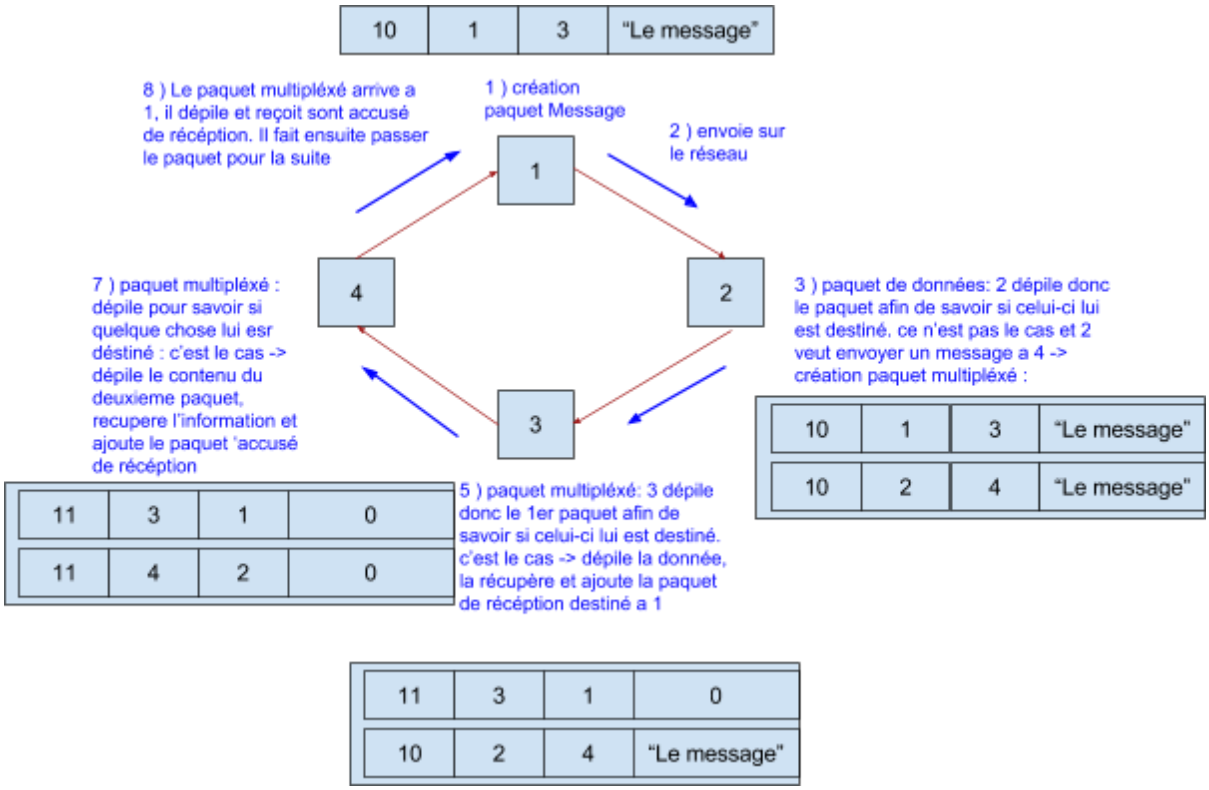
Exemple de paquet de multiplexage:

100	10	id_A	id_B	Data	11	id_C	id_D	Accusé	Ø	Ø

Ce paquet, de type multiplexé (100), contient 2 paquets: le premier est un paquet de données de A vers B, et le second est un accusé de réception de C vers D.

Protocole de création d'un paquet multiplexé:

1) Une machine reçoit un paquet non multiplexé:
    a) Le paquet lui est destiné, la machine renvoie alors un accusé de réception.
    b) Cette machine n'est pas en attente pour envoyer un message, elle renvoie simplement le paquet.
    c) Cette machine est en attente pour envoyer un paquet: elle crée alors un paquet multiplexé (type 100), dans le premier slot elle place le paquet qu'elle a reçu, et dans le deuxième slot elle place le paquet qu'elle souhaitait envoyer.
2) Une machine reçoit un paquet multiplexé, elle dépile chaque paquet contenu dans les différents slot afin de vérifier si un paquet lui est destiné.
    a) Si c'est le cas, elle le traite selon la méthode décrite précédemment. Ce paquet est alors retiré de son slot. Si un paquet doit être envoyé comme réponse (accusé de réception), il sera prioritaire sur le paquet en attente afin d'envoyer une réponse le plus vite possible.
    b) Si la machine est en attente, et que le nombre limite de paquet dans le paquet multiplexé n'est pas atteint, la machine encapsule son paquet à la suite des autres paquets données. S'il n'y a pas assez de place, elle reste en attente.
        i) Si après traitement, il ne reste plus qu'un seul paquet dans le paquet multiplexé, il est alors retiré du paquet multiplexé et envoyé au suivant.
        ii) Sinon, la machine remet en ligne le paquet multiplexé.
    c) Si la machine n'est ni en attente ni concernée par un des paquets, elle renvoie directement le paquet multiplexé au suivant.



VI. Mode connecté

Pour le mode connecté, on reprend le même principe qu'avant, mais avant qu'une machine n'envoie ses données, elle envoie d'abord un paquet de demande de connexion. On a ainsi les étapes suivantes:

1) La machine source réserve le canal ou se met en attente si le canal est occupé
2) Elle envoie son paquet de demande de connexion après avoir réservé le canal ou en multiplexage
3) Le destinataire répond (en multiplexage ou non) pour confirmer la connexion
4) La machine source envoie ses données
5) Le destinataire envoie son accusé de réception
6) La machine source envoie un paquet de fin de connexion
7) Le destinataire répond pour confirmer la fin de connexion

Description des paquets liés à la connexion :

Paquet de connexion :

101	ID_Machine source	ID_Machine destination	000000 …. 000000

Paquet de deconnexion:

110	ID_Machine source	ID_Machine destination	000000 …. 000000

Paquet d'acquittement de connexion/déconnexion:

111	ID_Machine destination	ID_Machine source	000000 …. 000000

Rappel de la description des paquets avec un codage du type de paquets sur 3 bits :

Paquet de libération du canal :

000	ID_Machine source	ID_Machine source	000000...000000

Paquet de réservation du canal :

001	ID_Machine source	ID_Machine source	000000...000000

Paquet de données :

010	ID_Machine source	ID_Machine destination	Codage des données

Paquet d'accusé de réception :

011	ID_Machine destination	ID_Machine source	000000 …. 000000