#### **Exercice 1**

1. On considère dans un premier temps que le canal de transmission peut corrompre les données transmises mais qu’il ne peut pas perdre ou détruire des données. Ce protocole stop-and-wait utilise 2 types d’acquittements:

* acquittements positif → permet d’acquitter un segment non corrompu
* acquittements négatif qui permet de signaler que le segment reçu est corrompu et donc de demander sa retransmission

Compléter l’automate en réception et en émission de l’entité de protocole transport afin de prendre en compte les pbs de retransmission. Vous disposez des fonctions suivantes:

* makeseg(buffer)
* isACK(SegRcv)
* corrupt(SeqRcv)
* extract(SegRcv, DataInd)

Réponse :

**Champs 1:**

?DataReq

-------------

SegSnd = makeseg(DATAREQ)

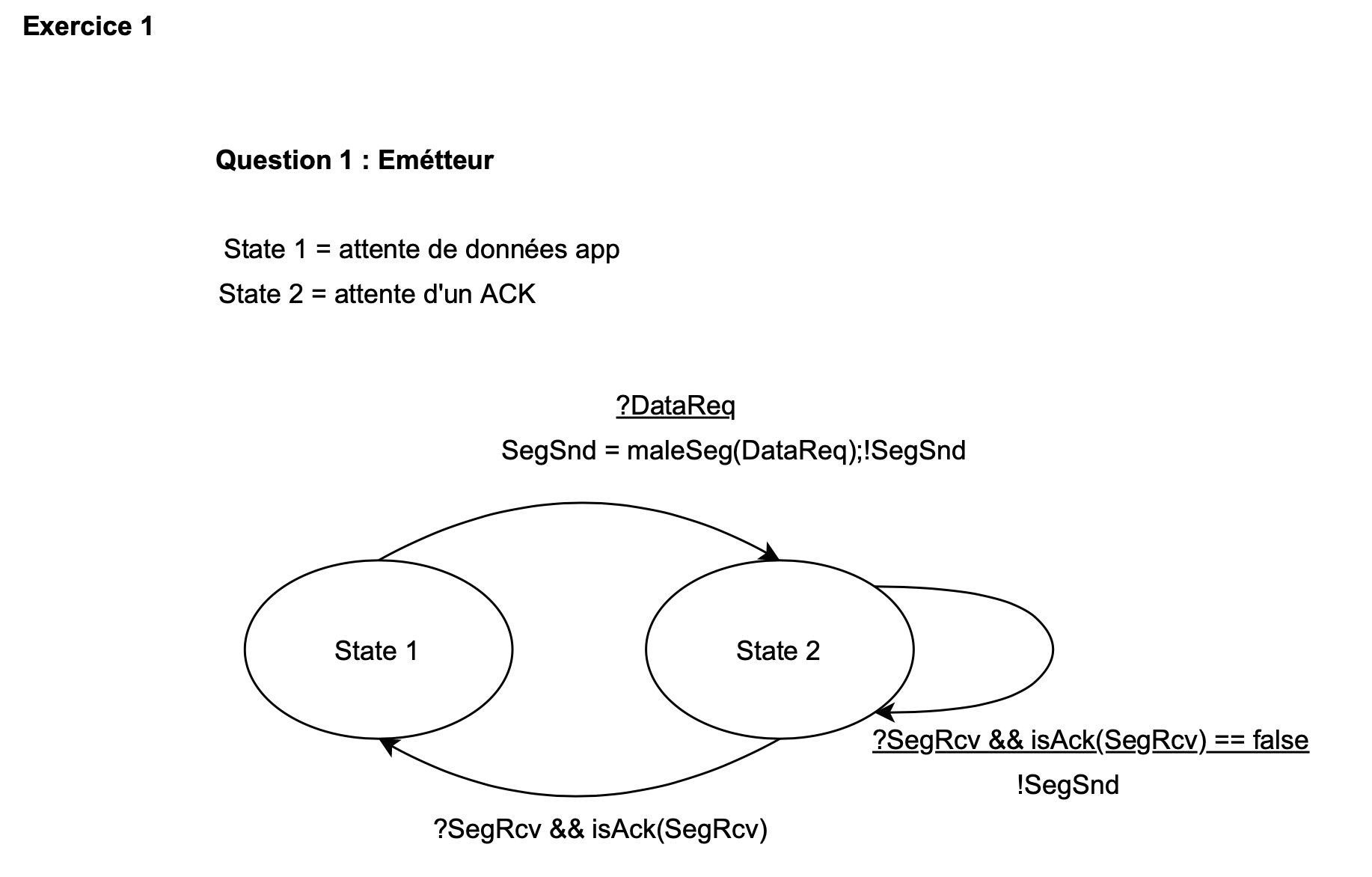
!SegSnd

**Champs 2:**

? isACK(SegRcv) && SegRev == false

-------------

!SeqSnd



1. Le canal peut être corrompu les acquittements. Remplir les automates:

testnum : vérifie le numéro de séquence

Réponse:

**Champs 1:**

?SegRev && corrupt(SegRev) == true

--------------------------------------------

SegSnd = makeseg(NACK)

!SegSnd

**Champs 2 (entre State 1 et State 1):**

?SegRev && corrupt(SegRev) == false && testnum(SegRev)==1

--------------------------------------------

SegSnd = makeseg(ACK)

!SegSnd

**Champs 3 (entre State 2 et State 1):**

?SegRev && corrupt(SegRev) == false && testnum(SegRev)==1

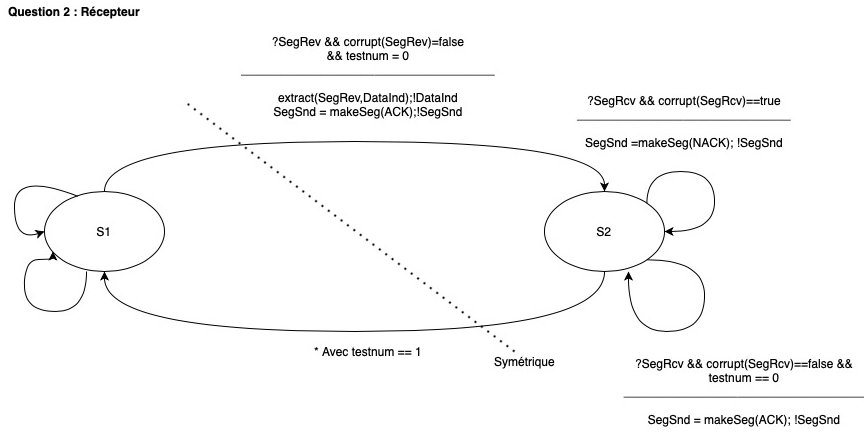
--------------------------------------------

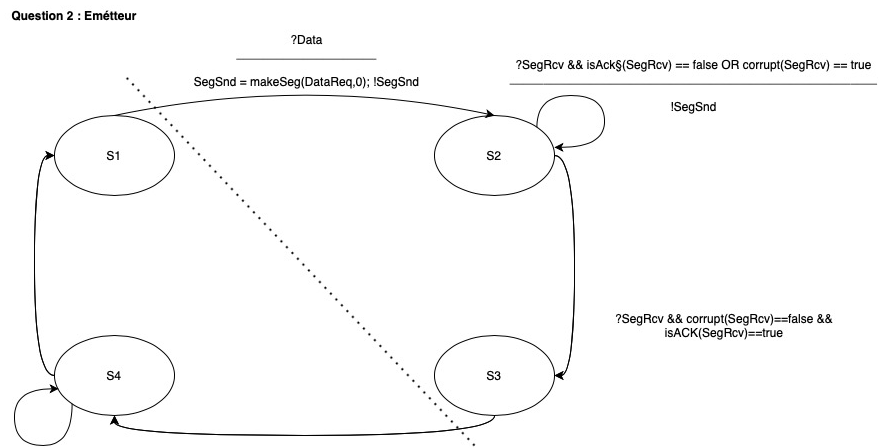
extract(SegRev, DataInd)

!DataInd

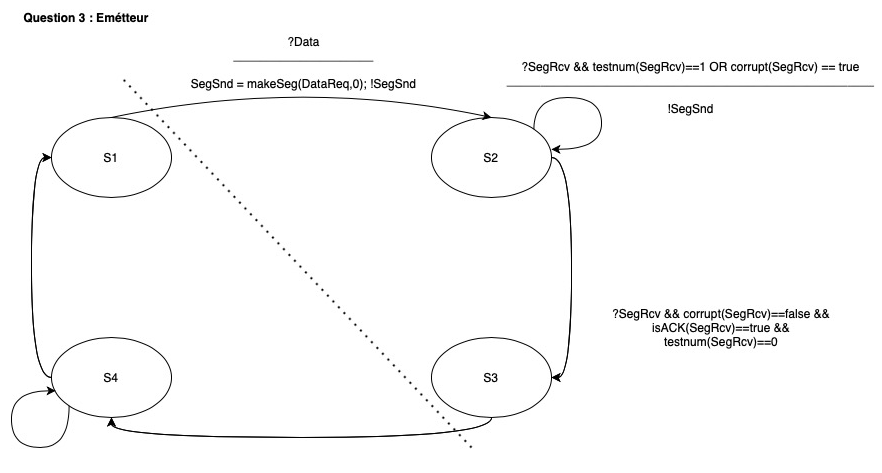
SagSnd = makeSeg(ACK)

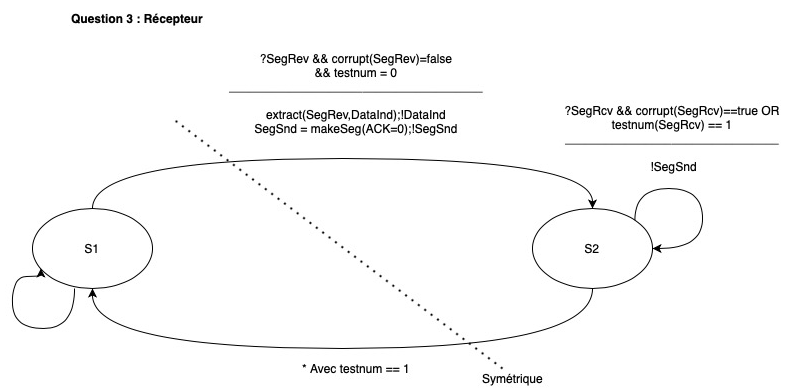
!SegSnd

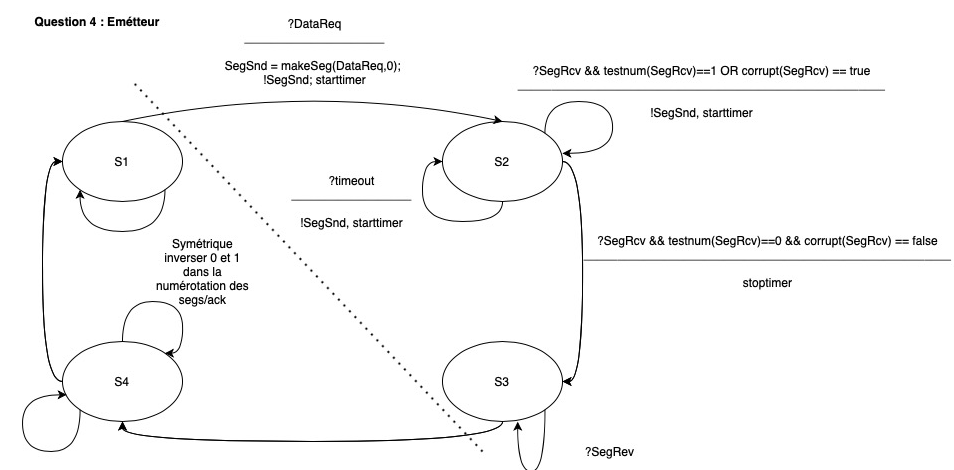




1. Supprimer le NACK Comment dans ce cas, l’émetteur peut savoir que la donnée est corrompue ?







#### **Exercice 2**

1. Donner une formule générique du taux d’utilisation U

U = Ttrans / T

T = Ttrans + Tprop + Tproc + Tack + Tprop + Tproc

On suppose que Tack et Tproc (attente buffer) négligeable

T = Trans + 2\*Tprop (2\*Tprop = RTT)

U = Ttrans / (Ttrans + 2\*Tprop)

#### a = Tprop/Ttrans et U = 1/(1+2\*a)

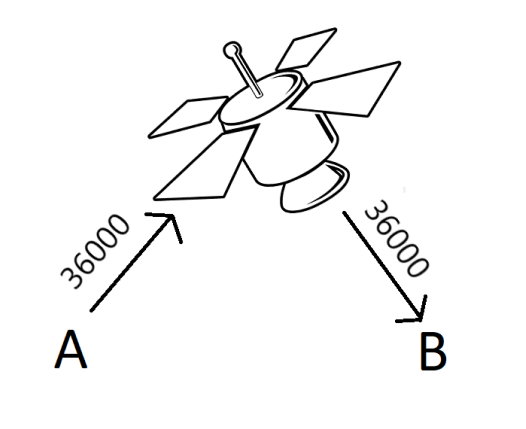
1. Soit une liaison géostationnaire…

Application numérique : Liaison satellite

Taille de données = 4000 bits

Débit = 1 Mbit/s

Vitesse prop = 300 000 km/s



Tprop = Distance/VitesseProp = 2\*36 000 / 300 000 = 0,24s

Ttrans = 4000/1 000 000 = 0,004s

a=0,24/0,004==60

U=1/(1+2\*60)=1/121=0,826%

1. Soit une liaison câble coaxiale ...

Tprop = 10/200 000 = 0.05 ms

Ttrans = 4 ms

U = 97,5%

1. Conclusion

Le protocole Stop & Wait pas adapté avec Tprop important. il faut utiliser un protocole à fenêtre.

#### **Exercice 3**

1. Bah on fait pas

#### **Exercice 4**

Quelles caractéristiques ne sont pas vraies pour UDP

1. Contrôle de flux : NON
2. Sans connexion : OUI
3. Avec connexion : NON
4. Démultiplexage : OUI
5. Ordre des paquets : NON

#### **Exercice 5**

Parmi les éléments suivants, quels sont ceux qui sont inclus dans un en-tête TCP mais pas dans un en-tête UDP ?

* numéro d’accusé de réception
* données
* numéro de séquence (si c’est pas le checksum)

1. Port d’origine : NON
2. Numéro de séquence : OUI (que TCP)
3. Taille des fenêtres : OUI (que TCP)
4. Données : NON
5. Port de destination : NON
6. Numéro d’accusé de réception : OUI (que TCP)

**Exercice 6**

Numérotation séquentielle des paquets qui indique le nombre d’octets transmis dans la connexion.