

# Réseaux informatiques

## Série 1

### 1.

Pour les deux premiers protocoles ARQ – Stop-and-wait et Go-back-N, écrivez les pseudocodes des programmes exécutés par l'émetteur et le récepteur. L'émetteur doit traiter 3 événements différents : réception d'une trame de données, réception d'une trame d'acquiescement, alerte en provenance d'un temporisateur.

Une proposition est d'implémenter les méthodes `linklayer.send(trame)`, `linklayer.receive(trame)` et `linklayer.handlerAlarm()`. Ces méthodes appellent les méthodes `hardware.send(trame)` et `reseaux.receive(trame)` pour communiquer avec les protocoles des couches voisines.

Vous pouvez coder la méthode `linklayer.send(trame)` comme non-bloquante. Cette méthode retourne `true` si la dernière trame envoyée a été acquiescée et `false` sinon. On laisse le soin à la méthode appelante de ré-invoquer `linklayer.send(trame)` si elle obtient `false`.

Vous utilisez deux types de trames : contrôle et données. Pour les distinguer utilisez une notation du type `trame.type==données` ou `trame.type=ACK`. Normalement un ACK est un acquiescement positif.

De même pour accéder aux champs de l'entête de la trame. Par exemple, pour accéder au numéro de séquence utilisez `trame.numACK` pour une trame `trame.type=ACK`

### 2.

Pour le protocole Stop-and-wait montrez que numéroter les trames est essentielle. Pour cela, supposez que les trames n'ont pas de numéro de séquence et trouver une séquence pour laquelle une trame n'est pas réceptionnée.