# IFt3325 Téléinformatique

# Devoir 2

Automne 2021

- Ce devoir est à rendre au plus tard le **29 Novembre à 23:59** (sur Studium)
- Ce devoir est à faire en équipe de deux.
- Le devoir est à programmer en langage JAVA
- Il est nécessaire que votre implémentation puisse être exécutée sur la plateforme du DIRO (jdk sous Linux) quel jdk
- Les présentations de ce devoir se tiendront à une date qui sera fixée avec votre démonstrateur. En moyenne, il faut prévoir 10-15 min par présentation.
- La politique du retard et de plagiat est énoncée dans le plan du cours.

**Objectifs :**

- Utilisation des sockets en java,
- Implanter une version simplifiée du protocole HDLC.

Deux entités communiquent à travers un réseau, la première entité envoie des données (émetteur) et la deuxième entité reçoit des données (récepteur). Nous supposons qu'il pourra y avoir des erreurs lors de la transmission des trames, des pertes de trames ou des pertes des accusés de réception.

Vous devez, dans ce devoir, simuler un sous ensemble du protocole HDLC. Au début de la communication, l'émetteur propose l'utilisation de Go-Back-N (REJ). Le récepteur est obligé d'accepter la demande de l'émetteur en lui envoyant un RR. Le polynôme générateur utilisé dans ce devoir est CRC-CCITT ($x^{16} + x^{12} + x^5 + 1$).

On considère que les numéros des trames sont sur 3 bits, mais durant la communication, les numéros de trames seront envoyés sur un octet avec la valeur en char de leur numéro comme cela est présenté dans la structure de la trame. n=3

Le bit stuffing sera introduit sur tous les champs sauf les fanions. Il faut introduire les bits avant l'envoie et les retirer à la réception. ?

La communication est unidirectionnelle dans le sens ou l'émetteur envoie une ou plusieurs trames de données (dépendamment de la taille de la fenêtre) et reçoit les accusés de réception. La durée du temporisateur utilisé pour la perte des trames est de 3 secondes. Les P-bit ne peuvent être utilisés que du côté de l'émetteur.

L'émetteur envoie des données (extraites d'un fichier texte) sous forme de trames. Les trames échangées sont orientées caractères et elles ont le format suivant :

?

| Flag | Type | Num | Données | CRC | Flag |
|---|---|---|---|---|---|

Où

- **Flag :** ce champ est de la taille de 1 octet (01111110), il délimite une trame.
- **Type :** ce champ est de taille 1 octet; il spécifie le type de la trame de la façon suivante.
  - **I :** trame d'information
  - **C :** demande de connexion (prend Num = 0 pour Go-Back-N)
  - **A :** accusé de réception (RR), le champ Num est utilisé dans ce cas pour le numéro à acquitter**.**
  - **R :** rejet de la trame Num et de toutes celles envoyées après (REJ).
  - **F :** fin de la communication.
  - **P :** trame avec P bit, équivalente à P bit.
- **Num :** ce champ est de taille 1 octet, il spécifie le numéro de la trame envoyée (I) ou de l'accusé de réception (RR, REJ).
- **Données :** ce quatrième champ est de taille variable, il est utilisé dans les trames d'information pour transporter les données. Il est de taille nulle dans le cas des trames d'accusé de réception. La taille peut être calculée à travers la détection des flags et en connaissant la taille des autres champs (qui est toujours fixe).
- **CRC :** contient le checksum calculé en utilisant CRC, la taille du champ est deux octets. Dans ce TP, vous devez calculer le checksum sur les champs Type, Num et Données.

Il faut implanter un émetteur qui permet de:

- Lire des données de fichier (chaque ligne du fichier est utilisée pour former une trame),
- Produire des trames et les envoyer,
- Attendre et traiter les accusés de réception,
- Ré-envoyer les données en cas de perte ou d'erreur.

Il faut implanter un récepteur qui permet de:

- Recevoir des trames
- Vérifier la présence d'erreurs ou non dans les trames reçues
- Envoyer des accusés de réception (RR).
- Envoyer des REJ en cas d'erreur.

Le devoir sera noté à travers les présentations que vous allez faire après la date de la remise. On vous demande de vérifier que le programme marche sur les plateformes du DIRO (sous Linux).

On vous demande aussi d'introduire des procédures de test qui introduisent des pertes et des erreurs de transmission (inverser la valeur d'un bit par exemple). Ces procédures doivent être implémentées dans une classe à part.

Vous devez afficher toutes les trames, côté émetteur et récepteur d'une façon élégante pour contrôler le trafic entre l'émetteur et le récepteur. Le style de la programmation (lisibilité, clarté, commentaires, l'optimisation de votre code…) et la conception orientée objet sont aussi notés.

Un petit rapport sur la conception adoptée est exigé, ce rapport doit contenir:
- Un diagramme de classe.
- Une brève description sur chaque classe et chaque méthode utilisée (c.-à-d. nom de la méthode, ses paramètres, sa fonction, son output).

Le barème de correction de ce devoir est le suivant :

Pour chaque partie :

- **10% :** Style de programmation, clarté du code, commentaires… etc.
- **5% :** Qualité du rapport.
- **10% :** Conception orientée objet (il est nécessaire de fournir un diagramme de classe durant la présentation).
- **5% :** dans la demande de connexion et au choix de Go-Back-N (REJ).
- **10% :** L'introduction du bit stuffing.
- **10% :** Le calcul du CRC.
- **15% :** Une classe à part pour les tests (elle doit permettre un affichage compréhensif).
- **17,5% :** Test de l'émetteur (lors du jour de la validation de votre travail).
- **17,5% :** Test de récepteur (lors du jour de la validation de votre travail).

La commande d'exécution de l'émetteur est :

```
% Sender <Nom_Machine> <Numero_Port> <Nom_fichier> <0>
```

La dernière option fait référence à l'utilisation de Go-Back-N (0).

La commande d'exécution du récepteur est :

```
% Receiver <Numero_Port>
```