

IFT585 - Télématicque

Travail pratique #1 – Hiver 2020

Date 18 janvier 2020

Enseignant

Bessam Abdulrazak

Statut Version

En vigueur

1.0.0

1 Contexte

La couche liaison de données dispose de série de protocoles permettant la bonne transmission des données. Nous nous intéressons dans ce TP à deux protocoles : Le protocole à fenêtre et au code Hamming

Le protocole à fenêtre d'anticipation est un protocole basé sur les points suivants :

1. Envoi de plusieurs trames¹ avant réception d'un acquittement.
2. Le nombre de trames autorisées à être envoyées est indiqué par la fenêtre de l'émetteur.
3. Le nombre de trames acceptables par le destinataire est indiqué par la fenêtre du récepteur.

Le code de Hamming est un code correcteur qui permet la détection et la correction automatique d'une erreur dans une série de bits

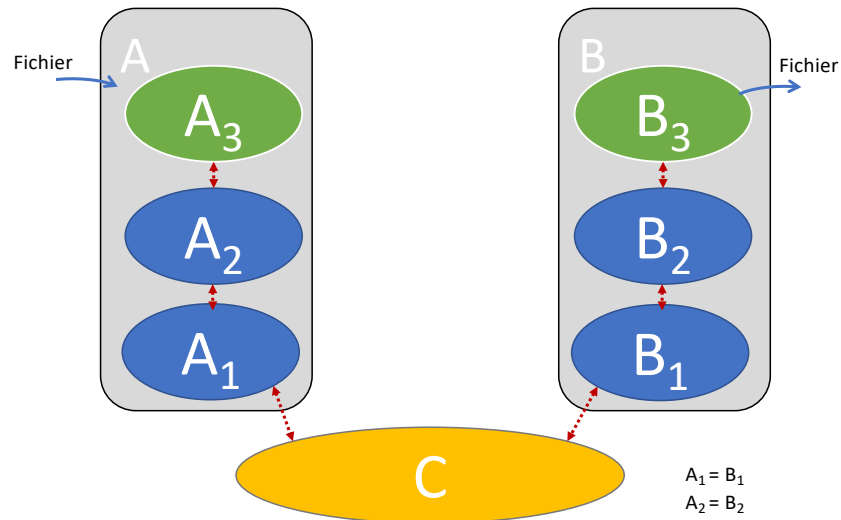
2 Énoncé

Le but de ce TP est de réaliser un programme illustrant le fonctionnement d'une couche LLC avec un protocole bidirectionnel à reprises multiples (sélectives) (voir l'exemple proposé par Tanenbaum / Protocole 6) et le code Hamming.

¹ Trame = format de données de la couche liaison

Le TP1 est à faire en deux étapes

3) Étape 1 : Simulation de l'envoi sur un support de transmission



L'étape 1 consiste à mettre en place un système simple de sept « Thread » qui représenteront :

- La station A : Threads A_3 , A_2 et A_1 ;
- La station B : Threads B_3 , B_2 , B_1 ; et
- Le support de transmission C : Thread C.

Votre programme final devra permettre la copie d'un fichier entre « Thread A_3 » et « Thread B_3 » et l'inverse.

La station A est identique à la station B. Les « Thread $A_{1/2/3}$ » devront être des images de « Thread $B_{1/2/3}$ ».

3.1 Les threads²

Thread A_3 --> Donnée utilisateur

Le « Thread A_3 » lit le fichier d'entrée (donnée utilisateur) et le passe à « Thread A_2 » dans un format de donnée de taille 16 octets (16 octets par 16 octets).

Le « Thread A_3 » doit avoir deux variables booléennes :

- **A3-ReadyToSendDown** (Prêt à transmettre à la couche inférieure) qui indique que la Thread A_3 a des données à transmettre à la couche inférieure.
- **A3-ReadyToReceiveFromDown** (Prêt à recevoir de la couche inférieure) qui indique que la Thread A_3 est prêt à recevoir de la couche inférieure.

La taille maximale du fichier à copier est de 150 M.

Thread A_2 --> Manipulation du format de donnée Trame

Le « Thread A_2 » crée seulement une trame de taille « FrameSize = 64 octets » contenant l'information à transmettre et la passe ensuite au « Thread A_1 ».

² Le descriptif décrit seulement le cas où « A » est la station à l'émission et « B » est la station réceptrice. Le même descriptif s'applique dans le cas inverse.

Le « Thread A₂ » doit avoir quatre variables booléennes :

- **A2-ReadyToRecieveFromUP** qui indique que le Thread A₂ est prêt à recevoir de la couche supérieure.
- **A2-ReadyToRecieveFromDown** qui indique que le Thread A₂ a est prêt à recevoir de la couche inférieure.
- **A2-ReadyToSendUp** qui indique que le Thread A₂ est prêt à transmettre à la couche supérieure.
- **A2-ReadyToSendDown** qui indique que le Thread A₂ est prêt à transmettre à la couche inférieure.

Thread A₁ --> Format de donnée : Trame-support

Le « Thread A₁ » crée seulement une trame-support de taille « **FrameSupportSize** » contenant l'information à transmettre et la passe ensuite au « Thread C ».

Le « Thread A₂ » doit avoir quatre variables booléennes :

- **A1-ReadyToRecieveFromUP** qui indique que le Thread A₂ est prêt à recevoir de la couche supérieure.
- **A1-ReadyToRecieveFromDown** qui indique que le Thread A₂ a est prêt à recevoir de la couche inférieure.
- **A1-ReadyToSendUp** qui indique que le Thread A₂ est prêt à transmettre à la couche supérieure.
- **A1-ReadyToSendDown** qui indique que le Thread A₂ est prêt à transmettre à la couche inférieure.

et une variable numérique

- **FrameSupportSize** = 128 octets dans l'étape 1.

Thread C : Support de transmission

Le « Thread C » permet de transmettre les trames-supports entre les deux machines A et B et simuler les erreurs de transmission.

Le « Thread C » dispose de deux tampons (un pour l'envoi de A vers B et l'autre de B vers A) qui doivent être deux vecteurs (« array ») de taille correspondant à un paramètre entré dans le programme (**BufferThread-C**) multiplié par « **FrameSupportSize** » octets. Ces vecteurs agissent comme une file circulaire (facile à construire avec un modulo %).

Le « Thread C » doit avoir quatre variables booléennes :

- **C-ReadyToRecieveFromA** qui indique que le Thread C est prêt à recevoir de la machine A (Thread A₁).
- **C-ReadyToRecieveFromB** qui indique que le Thread C est prêt à recevoir de la machine B (Thread B₁).
- **C-ReadyToSendToA** qui indique que le Thread C est prêt à transmettre à la machine A (Thread A₁).
- **C-ReadyToSendToB** qui indique que le Thread C est prêt à transmettre à la machine B (Thread B₁).

et deux variables numérique

- **BufferThread-C** qui indique la taille du tampon (en termes de nombre de trame-support)
- **BufferThread-Delay** pour simuler la latence dans « Thread C » = latence entre l'émission et la réception d'une autre machine (en ms).

3.1.1 Entre Threads

A₃ --> A₂

On envoie les données de « Thread A₃ » à « Thread A₂ » dès que « Thread A₃ » est disponible pour l'envoi "**A3-ReadyToSendDown**" et « Thread A₂ » est disponible pour la réception "**A2-ReadyToRecieveFromUp**" (sauf si aucune donnée n'est disponible à envoyer).

A₂ --> A₁

On envoie les données de « Thread A₂ » à « Thread A₁ » dès que « Thread A₂ » est disponible pour l'envoi "**A2-ReadyToSendDown**" et « Thread A₁ » est disponible pour la réception "**A1-ReadyToRecieveFromUp**" (sauf si aucune donnée n'est disponible à envoyer).

A₁ --> C

On envoie les données de « Thread A₁ » à « Thread C » dès que « Thread A₁ » est disponible pour l'envoi "**A1-ReadyToSendDown**" et « Thread C » est disponible pour la réception "**C-ReadyToRecieveFromA**" (sauf si aucune donnée n'est disponible à envoyer).

C --> A₁

On envoie les données de « Thread C » à « Thread A₁ » dès que « Thread C » est disponible pour l'envoi "**C-ReadyToSendToA**" et « Thread A₁ » est disponible pour la réception "**A1-ReadyToReceiveFromDown**" (sauf si aucune donnée n'est disponible à envoyer).

A₁ --> A₂

On envoie les données de « Thread A₁ » à « Thread A₂ » dès que « Thread A₁ » est disponible pour l'envoi "**A2-ReadyToSendUp**" et « Thread A₂ » est disponible pour la réception "**A2-ReadyToReceiveFromDown**" (sauf si aucune donnée n'est disponible à envoyer).

A₂ --> A₃

On envoie les données de « Thread A₂ » à « Thread A₃ » dès que « Thread A₂ » est disponible pour l'envoi "**A2-ReadyToSendUp**" et « Thread A₃ » est disponible pour la réception "**A3-ReadyToReceiveFromDown**" (sauf si aucune donnée n'est disponible à envoyer).

3.1.2 Erreurs de transferts à simuler dans le « Thread C »

Des erreurs de transferts³ devront pouvoir être introduites dans le « Thread C ». Il est nécessaire d'afficher à l'écran les erreurs avec les informations d'émission et de réception.

Les types d'erreurs à introduire dans le « Thread C » sont :

1. Changement de la valeur des bits (ex. : 0,1,4, 50, 190) dans des trames (ex. : numéro 234,445,290)
2. Changement de la valeur des bits dans toutes les trames
3. Changement de la valeur des bits dans des trames choisies de façons aléatoire.

Le support de transmission peut induire des erreurs dans les deux sens, c-à-d. lorsque A envoie à B et lorsque B répond à A.

3.2 Paramètres de configuration (voir section 5)

Dans l'étape 1 du TP1, votre programme prendra les paramètres suivants ⁴:

1. L'emplacement du fichier source à copier et celui de destination pour la copie du fichier.
2. L'emplacement des erreurs.
3. **BufferThread-C**: la taille du tampon du « Thread C »
4. **BufferThread-Delay** : La latence dans le « Thread C »

3.3 Livrables de la première étape

- Un rapport collectif de travail à soumettre via Moodle :
 - Le rapport devra contenir une explication du code, la procédure d'exécution, une méthode pas à pas pour compiler et exécuter les programmes, un guide d'utilisateur expliquant le fonctionnement de chacun des programmes, une légère analyse expliquant comment vous avez effectué la conception de ces programmes ainsi qu'une indication concernant ce qui fonctionne bien, ce qui fonctionne moins bien et ce qui ne fonctionne pas dans ces programmes (la rédaction de votre travail ne doit pas excéder 4 pages, plus une page de garde précisant le nom et le matricule des auteurs).

³ Erreurs de transfert = altération de bits

⁴ Vous devez utiliser le fichier de commande pour configurer les paramètres. L'utilisateur pourra donc choisir l'emplacement du fichier d'entrée, l'emplacement du fichier de sortie, introduire les différents paramètres et simulant le support de transmission.

- La plateforme Git-Lab du cours doit être utilisée pour la gestion du code. Soumettez une première ébauche de code et graduellement chaque fois que vous apporté une modification mettez votre code à jour sur Git-lab.
- La dernière version à remettre pour la première étape doit être soumise également via TurnIn pour une correction automatique. Vous devez également soumettre un fichier « Langage.txt » avec une seule entre qui indique le langage utilisé (C++, C#, Java, ou Python).

3.4 Gabarit de correction

Les points suivants sont octroyés par l'accomplissement des fonctionnalités requises.

Aspect	Points
Rapport explicatif détaillé sur le fonctionnement du programme ⁵	5
Implémentation du système avec une démonstration que le programme fonctionne correctement --> Réussir le passage au correcteur automatique de TurnIn ⁶ .	25
Total	33

⁵ Le correcteur ou la correctrice peut soustraire jusqu'à 5% de chaque évaluation pour la qualité du français. Des consignes supplémentaires ou des modifications pourront être communiquées au cours du trimestre.

⁶ Langages acceptés : C++, C#, Java, Python 3.

4) Étape 2 : Simulation de la couche LLC

La deuxième étape consiste à implémenter le protocole à fenêtre dans A_2 (et B_2) et le code Hamming dans A_1 (et B_1). Les « Thread $A_{1/2/3}$ » devront être des images de « Thread $B_{1/2/3}$ », mais ayant des paramètres de configuration différentes.

4.1 Scenario où A est la machine d'émission et B est la machine de réception⁷⁻⁸

Thread A2

Dans cette étape 2, « Thread A_2 » crée une trame de taille « **FrameSize** = 64 octets » contenant l'information à transmettre (similaire à l'étape 1) et ajoute le numéro de la trame et les messages de contrôle (ACK, NAK, etc.) du protocole à fenêtre avant de passer la trame au « Thread A_1 ».

- Vous pouvez décider du format de la trame, en utilisant une structure de données (STRUCT) et la traduire en octets.
- Lorsqu'il reste de l'espace non utilisé dans la trame, on remplit ces espaces avec du bourrage.
- La taille des fenêtres d'anticipation (**WinSize**) est variable selon le choix de l'utilisateur.
- Le délai de temporisation (« time-out ») utilisé pour le protocole à fenêtre doit être configurable (**TimeOut**).

Thread A1

Dans cette étape 2, « Thread A_1 » reçoit la trame de A_2 , applique le code Hamming, ajoute l'information adéquate dans un nouveau format de donnée du support de transmission (« trame-support »).

- La taille « **FrameSupportSize** » de la trame-support est à calculer en fonction du code Hamming (dans cette étape 2).
- Lorsqu'il reste de l'espace non utilisé dans la trame-support, on remplit ces espaces avec du bourrage (à ne pas oublier que Hamming agisse sur des bits).

Thread B1

Le « Thread B_1 » attend qu'une trame-support soit disponible sur le support de transmission (**C-ReadyToSendToB**) et la lit lorsqu'il est prêt à la recevoir (**B1-ReadyToReceiveFromDown**).

Lorsqu'une trame est reçue, « Thread B_1 » vérifie la trame-support.

- Dans le cas de code détecteur, le « Thread B_1 » s'assurer que la trame est correcte
 - Si ce n'est pas le cas, le « Thread B_1 » rejette la trame.
 - Si elle est bonne, le « Thread B_1 » passe la trame au « Thread B_2 ».

Lorsqu'un octet est reçu par le « Thread B_3 », le programme l'écrit dans le fichier de destination.

« Thread B_2 » envoie également les messages de contrôle (ex. : « ACK ») à « Thread A_2 » via « Thread B_1 ».

4.1.1 Paramètres de configuration (voir section 5)^{9 10}

En plus des paramètres de l'étape 1,

⁷ Le descriptif décrit seulement le cas où « A » est la station d'émission et « B » est la station de réception. Le même descriptif s'applique dans le cas inverse.

⁸ Nous parlons ici de protocoles bidirectionnels. Les threads « Thread $A_{1/2/3}$ » et « Thread $B_{1/2/3}$ » émetteur et récepteur au même temps.

⁹ Le programme devra afficher à l'écran les données envoyées/reçues par l'émetteur/le récepteur. L'utilisateur doit pouvoir suivre facilement le transfert du fichier (L'affichage peut se faire en mode console ou en utilisant une interface graphique).

¹⁰ Le programme devra afficher à l'écran les données envoyées/reçues par l'émetteur/le récepteur. L'utilisateur doit

5. « **WinSize** » pour A et pour B : La taille du tampon utilisée pour le protocole à fenêtre pour la machine A et B (WinSize pour A peut être différent du celui du B).
6. « **TimeOut** » : Le délai de temporisation (« time-out ») pour la machine A et B (TimeOut pour A peut être différent du celui du B).
7. Choix de l'utilisation du code Hamming --> détecteur ou correcteur

4.2 Livrables de L'étape 2

- Rapport individuel d'activité à soumettre via Moodle :
Un rapport personnel d'une page maximum, décrivant la participation du reste de l'équipe, et non pas de son propre travail¹¹⁻¹².
- Un rapport collectif de travail à soumettre via Moodle :
Le rapport devra contenir un imprimé décrivant le code, la procédure d'exécution, une méthode pas à pas pour compiler et exécuter les programmes, un guide d'utilisateur expliquant le fonctionnement de chacun des programmes, une légère analyse expliquant comment vous avez effectué la conception de ces programmes ainsi qu'une indication concernant ce qui fonctionne bien, ce qui fonctionne moins bien et ce qui ne fonctionne pas dans ces programmes (la rédaction de votre travail ne doit pas excéder 8 pages, plus une page de garde précisant le nom et le matricule des auteurs).
- La plateforme Git-Lab du cours doit être utilisée pour la gestion du code¹³. Soumettez une première ébauche de code et graduellement chaque fois que vous apporté une modification mettez votre code à jour sur Git-lab.

4.3 Gabarit de correction

Les points suivants sont octroyés par l'accomplissement des fonctionnalités requises.

Aspect	Points
Rapport explicatif détaillé sur le fonctionnement du programme ¹⁴	10
Implémentation de l'algorithme Rejet sélectif (avec une démonstration que le programme fonctionne correctement ¹⁵)	25
Afficher à l'écran les données	7
Implémentation d'un algorithme Hamming (avec une démonstration que le programme fonctionne correctement ¹⁵)	25
Total	100

pouvoir suivre facilement le transfert du fichier (L'affichage peut se faire en mode console ou en utilisant une interface graphique.

¹¹ Ne pas soumettre le rapport individuel implique un -30 points automatique.

¹² Le correcteur peut appliquer une pénalité (ou une note 0) aux membres du groupe qui n'ont pas participé à la réalisation du TP d'une façon adéquate. Limiter l'implication à la rédaction ou à la correction du document n'est pas acceptée.

¹³ Le code doit être exécutable sur les machines des laboratoires du département d'informatique.

¹⁴ Le correcteur ou la correctrice peut soustraire jusqu'à 5% de chaque évaluation pour la qualité du français. Des consignes supplémentaires ou des modifications pourront être communiquées au cours du trimestre.

¹⁵ Le correcteur ou la correctrice peut appliquer une note 0 pour un programme qui ne fonctionne pas correctement.

5 Paramètres de configuration

Un fichier avec le nom « parametres.txt » (voir l'exemple de fichier joint)

Paramètres généraux

- Ligne 1: L'emplacement du fichier source à copier
Ligne 2: L'emplacement du fichier de destination pour la copie du fichier

Paramètres des Erreurs

- Ligne 3: Changement de la valeur des bits (ex. : 0,1,4, 50, 190)
Ligne 4: Choix du changement
 - A pour toutes les trames
 - B pour des trames choisies de façons aléatoire
 - C pour des trames choisit dans la ligne suivante.
Ligne 5: Les trames à changer (ex. : 234,445,290)

Paramètres de l'étape 1

- Ligne 6: BufferThread-C (ex. 100)
Ligne 7: BufferThread-Delay (ex. 10 pour 10 millisecondes)

Paramètres de l'étape 2

- Ligne 1: WinSize pour la machine A (ex. 85)
Ligne 2: WinSize pour la machine B (ex. 36)
Ligne 3: TimeOut pour la machine A (ex. 1000 pout 1 secondes)
Ligne 4: TimeOut pour la machine B (ex. 1500 pout 1.5 secondes)
Ligne 5: Choix de l'utilisation du code Hamming --> détecteur ou correcteur
 - D : Détecteur
 - R : Correcteur

6 Nom des fichiers

Il est demandé de soumettre les fichiers / rapports sous le format Word (.doc / .docx) et d'utiliser le format suivant pour le nommage.

Rapport collectif

Groupe-Numéro du groupe__TP6__Rapport-Collectif.doc

Exemple : Groupe-F__TP2__Rapport-Collectif. Doc

Rapport Individuel

Groupe-Numéro du groupe__TP6__Rapport-Individuel__Nom-Prenom.doc

Exemple : Groupe-F__TP2__Rapport-Individuel__Charles-Roy.doc