INF1009 - Réseaux d'ordinateur 1

Travail pratique

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

Réalisé par:

Abdellah Tchikou
Patrick Duhaime

Professeur: Adam Joly
Session hiver 2018

Table des matières

Avant-Propos

Problèmes ou difficultés rencontrées

Le principal problème rencontré a été la compréhension de l'énoncé pour la réalisation du travail, il a fallu plusieurs soirées pour analyser le problème de long en large avant de pouvoir entreprendre le code à proprement dire. Un autre problème a été de faire fonctionner les Threads, étant à ma première expérience avec ces derniers sous C#, ce fut un bon défi de faire fonctionner le tout.

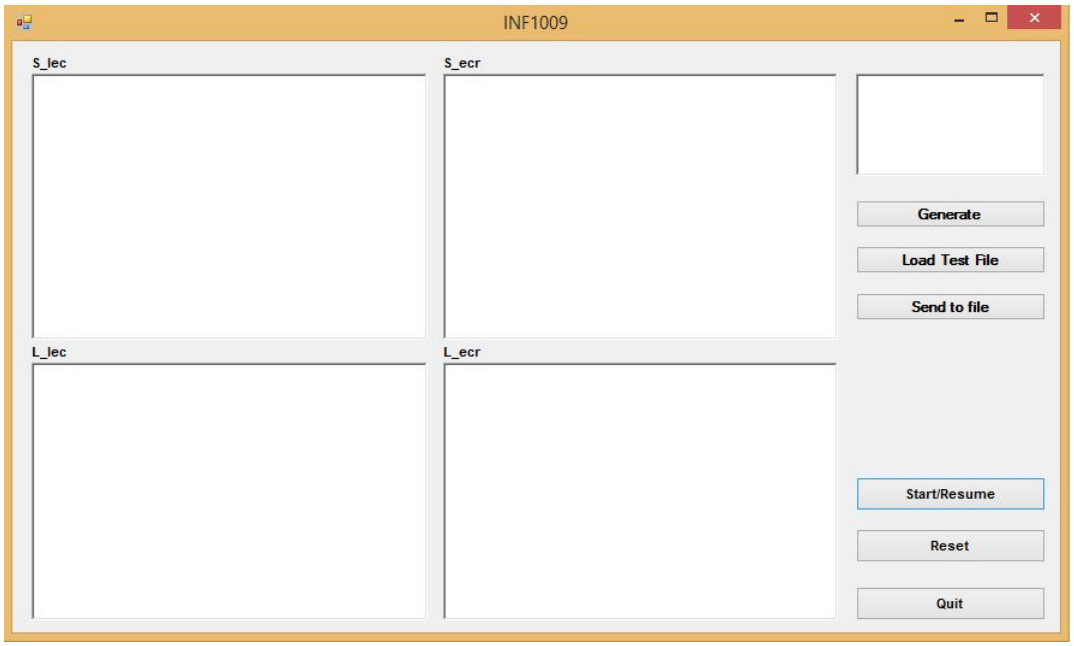
Commentaires sur le travail

J'ai fait ce travail à temps perdu, et avec les cours de la dernière session, je n'ai pas eu beaucoup de temps à perdre disons, dans le cours de conception j'ai fait un travail sur le refactoring ou il était dit que les commentaires sont à éviter, de nommer les variables et les méthodes significativement était beaucoup mieux selon le document utilisé pour faire le travail. J'ai donc décidé d'appliquer ce concept à mon style de programmation ce qui fut une très mauvaise idée au final, j'ai eu de la difficulté à me souvenir des changements que j'ai fait entre chaque fois ou je travaillais sur le programme et j'ai eu de la difficulté à me retrouver, j'ai du relire souvent les méthodes pour retrouver le chemin logique du programme car trop de temps et d'autres projets qui s'entremêlent m'ont fait oublier.

Voici un commentaire un peu plus subjective sur le travail, pour évaluer les compétences en réseau, il aurait été beaucoup plus approprié de demander de configurer un cluster DNS ou encore d'établir une communication client/serveur sécurisé utilisant un protocole comme Triple DES ou autre. A mon avis, les connaissances en réseau ont peu voir avec ce travail, les travaux effectués en système d'exploitation (SIF1015) se rapprochent beaucoup plus d'une implémentation réaliste des protocoles de communication client serveur dans une application. Personnellement je dirais que ce travail ne reflète pas du tout mes compétences en réseau.

Court guide utilisateur

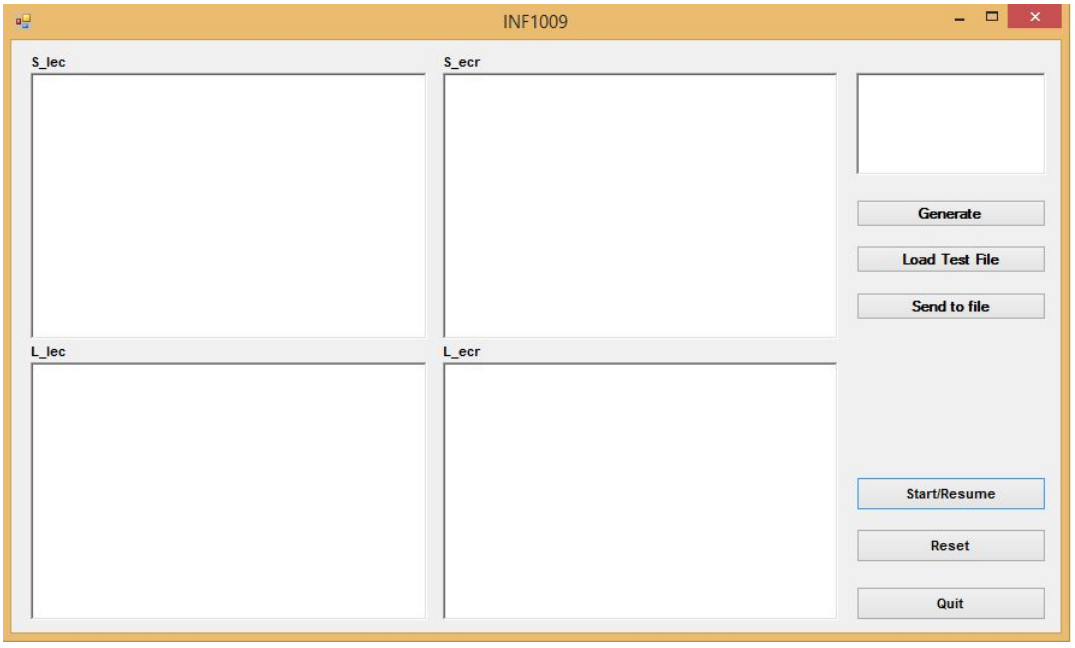
Tests aléatoires



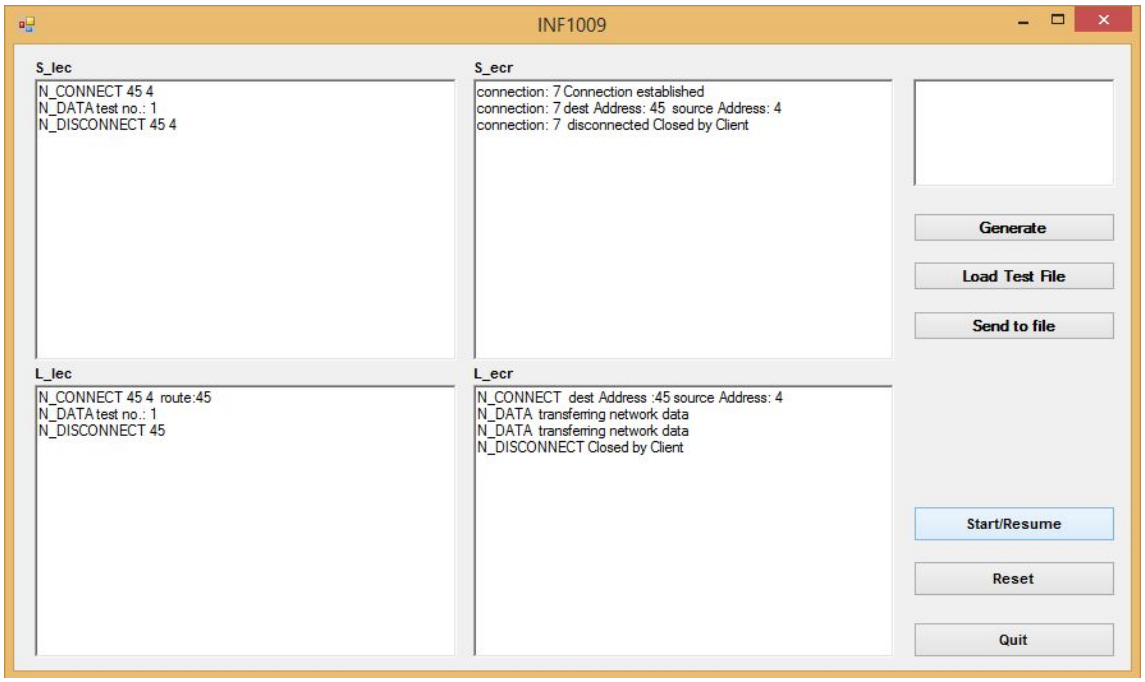
Le programme permet de générer des requêtes aléatoire, il suffit de cliquer sur Generate et une requête sera affiché dans la fenêtre d'affichage, vous pouvez cliquer le bouton Generate autant de fois que vous voulez jusqu'à ce que la requête générée vous convient, lorsque c'est le cas, vous cliquez sur 'Send to file' pour charger la requête sur le fichier S_lec, une fois la requête chargé vous pouvez cliquer sur Start/Resume, cela aura pour effet de lancer la requête sur le réseau simulé.

Si vous générez un test et cliquez sur Send to file et que par la suite vous générez un autre test et cliquez sur Send to file à nouveau, autrement dit si vous faites les étapes 1 et 2 à répétition sans faire l'étape 3, cela aura pour effet d'écraser la première requête et vous ne verrez que la dernière requête générée se lancer sur le programme, vous devez donc faire les 3 étapes en ordre. pour effacer l'affichage de toutes les fenêtres et remettre tous les paramètres du programme à zéro cliquez sur Reset.

Le fichier test



Le fichier test est coder directement dans le programme, vous pouvez le charger en cliquant ‘Load test file’, il faut ensuite écrire le fichier S_lec en cliquant sur ‘Send to file. A ce stade le test est prêt à être lancé, vous pouvez cliquer sur Start/Resume pour lancer la première requête de connection sur le réseau simulé, la lecture du fichier s'arrêta après la lecture d’un premier ‘disconnect’.



Pour poursuivre vous cliquez sur Start/Resume à nouveau, la lecture du fichier reprendra à la prochaine requête.

Fichier test généré par le programme:

N_CONNECT 1 11
N_DATA Start testing INF1009
N_DISCONNECT 1 11
N_CONNECT 47 15
N_DATA negative Acknoledgment
N_DISCONNECT 47 15
N_CONNECT 200 27
N_DATA declined by Network
N_DISCONNECT 200 27
N_CONNECT 200 500
N_DATA declined by Network - no route
N_DISCONNECT 200 500
N_CONNECT 9 19
N_DATA no awnser
N_DISCONNECT 9 19
N_CONNECT 12 13
N_DATA multiple of 13 - connection declined by destination
N_DISCONNECT 12 13
N_CONNECT 58 217
N_DATA This is the last test, this program was written for the recognition of the achievements of Patrick Duhaime for the course INF1009 ...
N_DISCONNECT 58 217

Analyse des résultats

1ere requête

Fenêtre S_lec
N_CONNECT 1 11
N_DATA Start testing INF1009
N_DISCONNECT 1 11

Fenêtre S_ecr
connection: 1 Connection established
connection: 1 dest Address: 1 source Address: 11
connection: 1 disconnected Closed by Client

Fenêtre L_lec
N_CONNECT 1 11 route:1
N_DATA Start testing INF1009
N_DISCONNECT 1

Fenêtre L_ecr
N_CONNECT dest Address :1 source Address: 11
N_DATA transferring network data
N_DISCONNECT Closed by Client

La première requête demande une connection à l'adresse de destination numéro 1 à partir de l'adresse source numéro 11. On peut observer la route choisi par la table de routage définie dans l'énoncé de travail. La connection se déroule normalement.

2ieme requête

Fenêtre S_lec
N_CONNECT 47 15
N_DATA negative acknowledgement
N_DISCONNECT 47 15

Fenêtre S_ecr
connection: 4 Connection established
connection: 4 dest Address: 47 source Address: 15
connection: 4 disconnected Closed by Client

Fenêtre L_lec
N_CONNECT 47 15 route:47
N_DATA negative acknowledgement
N_DISCONNECT 47

Fenêtre L_ecr
N_CONNECT dest Address :47 source Address: 15
N_DISCONNECT Closed by Client

Cette requête ne reçoit pas de réponse du destinataire, le traitement retourne un paquet de non acquittement.

```
if (found)
{
    if (packet.sourceAddr % 15 != 0)...
}
else
{
    returnPacket = Packet.encapsulateAcknowledge(packetFromNetwork[0], Packet.ConvertToByte(pr), false);
    packet2Network = Packet.encapsulateBytes(returnPacket, "NACK");
    packetProcessing2Network.Enqueue(packet2Network);
}
```

Si un paquet a été trouvé et que ce dernier n'est pas originaire d'une adresse qui est un multiple de 15 on procède, sinon on retourne un paquet de non acquittement.

3ieme requête

Fenêtre S_lec
N_CONNECT 200 27
N_DATA declined by Network
N_DISCONNECT 200 27

Fenêtre S_ecr
connection: declined by Network!

Fenêtre L_lec
N_CONNECT 200 27 route:254
N_CONNECT 200 500 route: Error, not found !

Fenêtre L_ecr
N_DISCONNECT Closed by Client

Puisque l'adresse source est un multiple de 27, le réseau refuse et retourne un disconnect immédiat.

```
if (sourceAddr[0] % 27 == 0 || int.Parse(transportNpdu.sourceAddr) > 249 || int.Parse(transportNpdu.destAddr) > 249)
{
    disconnected = true;
    connected = false;

    npdu2Transport = new Npdu();
    npdu2Transport.type = "N_DISCONNECT.ind";
    npdu2Transport.target = "00000010";
    npdu2Transport.connection = "255";
    network2Transport.Enqueue(npdu2Transport);
}
```

Si l'adresse source est un multiple de 27 ou supérieur à 249 ou que l'adresse de destination est supérieur à 249, le réseau refuse et retourne un disconnect au requérant.

4ieme requête

Fenêtre S_lec
N_CONNECT 200 500
N_DATA declined by Network - no route
N_DISCONNECT 200 500

Fenêtre S_ecr
connection: declined by Network!

Fenêtre L_lec
N_CONNECT 200 500 route: Error, not found !

Fenêtre L_ecr

Ici l'adresse source 500 ne fait pas parti du réseau, il est impossible d'établir une route puisque cette dernière est défini selon des adresses destination et source valides.

```
if (sourceAddr[0] % 27 == 0 || int.Parse(transportNpdu.sourceAddr) > 249 || int.Parse(transportNpdu.destAddr) > 249)
{
    disconnected = true;
    connected = false;

    npdu2Transport = new Npdu();
    npdu2Transport.type = "N_DISCONNECT.ind";
    npdu2Transport.target = "00000010";
    npdu2Transport.connection = "255";
    network2Transport.Enqueue(npdu2Transport);
}
```

Si l'adresse source est un multiple de 27 ou supérieur à 249 ou que l'adresse de destination est supérieur à 249, le réseau refuse et retourne un disconnect au requérant

5ieme requête

Fenêtre S_lec
N_CONNECT 9 19
N_DATA no awnser
N_DISCONNECT 9 19

Fenêtre S_ecr
connection: 6 Connection established
connection: 6 disconnected Closed by Client

Fenêtre L_lec
N_CONNECT 9 19 route:9

Fenêtre L_ecr
N_DISCONNECT Closed by Client

L'adresse source étant un multiple de 19, le traitement ne retourne un paquet de release qui termine la connexion.

```
if (packetFromNetwork[2] % 19 != 0)...
else
{
returnPacket = Packet.encapsulateRelease(packetFromNetwork[0], packetFromNetwork[2], packetFromNetwork[3], true);
packet2Network = Packet.encapsulateBytes(returnPacket, "release");

packetProcessing2Network.Enqueue(packet2Network);
}
```

Si le paquet ne provient pas d'une adresse source qui est un multiple de 19, on procède, sinon, retourne un paquet release.

6ieme requête

Fenêtre S_lec
N_CONNECT 12 13
N_DATA multiple of 13 - connection declined by destination
N_DISCONNECT 12 13

Fenêtre S_ecr
connection: 4 Connection established
connection: 4 disconnected Closed by Client

Fenêtre L_lec
N_CONNECT 12 13 route:12

Fenêtre L_ecr
N_DISCONNECT Closed by Client

L'adresse source étant un multiple de 13, le traitement ne retourne un paquet de release qui termine la connexion.

```
if (packetFromNetwork[2] % 13 == 0)
{
    returnPacket = Packet.encapsulateRelease(packetFromNetwork[0], packetFromNetwork[2], packetFromNetwork[3], true);
    packet2Network = Packet.encapsulateBytes(returnPacket, "release");
}
else
{
    returnPacket = Packet.encapsulateConnectionEstablished(packetFromNetwork[0], packetFromNetwork[2], packetFromNetwork[3]);
    packet2Network = Packet.encapsulateBytes(returnPacket, "established");
    packets.Add(returnPacket);
}

packetProcessing2Network.Enqueue(packet2Network);
```

Si le paquet provient d'une adresse source qui est un multiple de 13, on retourne un paquet release, sinon, on retourne un paquet established.

7ieme requête

Fenêtre S_lec
N_CONNECT 58 217
N_DATA This is the last test, this program was written for the recognition of the achievements of Patrick Duhaime for the course INF1009 ...
N_DISCONNECT 58 217

Fenêtre S_ecr
connection: 5 Connection established
connection: 5 dest Address: 58 source Address: 217
connection: 5 disconnected Closed by Client

Fenêtre L_lec
N_CONNECT 58 217 route:251
N_DATA This is the last test, this program was written for the recognition of the achievements of Patrick Duhaime for the course INF1009 ...
N_DISCONNECT 58

Fenêtre L_ecr
N_CONNECT dest Address :58 source Address: 217
N_DATA transferring network data
NACK negative Acknowledgment :255
N_DATA transferring network data
N_DATA transferring network data
N_DATA transferring network data
N_DATA transferring network data
N_DISCONNECT Closed by Client

La requête se déroule normalement mais le message est long ce qui force le découpage en plusieurs paquets.

Il est à noter que dans plusieurs tests un disconnect est envoyer du traitement ou du réseau mais que puisque le fichier test contient lui aussi un disconnect, vous verrez des N_DISCONNECT Closed by Client **aux fin des commandes, la fermeture à déjà eu lieu par l'autre extrémité.**

Il se peut que l'affichage ne soit pas exactement comme dans le test, le programme semble ne pas être stable à 100%, il suffit de faire un reset et de relancer le test si vous rencontrez ce genre de problèmes