

Projet de Session

PRÉSENTÉ À

M BOUCIF AMAR BENSABER

COURS

RÉSEAUX D’ORDINATEURS 1

PAR **GANSONRE ISMAEL** *SEMANOU RICHAR,*

*DIALLO MARIAMA, DIALLO FATIMA,*

*AUDREY BONI*

*ET RAMI BERLAT*

25 AVRIL 2022

Contents

[Description du logiciel 3](#_bookmark0)

[Instructions spéciales 3](#_bookmark1)

[Description des fichiers 3](#_bookmark2)

[S\_lec 3](#_bookmark3)

[S\_ecr 3](#_bookmark4)

[L\_ecr 4](#_bookmark6)

[L\_lec 4](#_bookmark7)

[Description des classes 4](#_bookmark8)

[Les classes de paquet 4](#_bookmark9)

[La classe Constantes 4](#_bookmark10)

[La classe ConnexionTransport 5](#_bookmark11)

[La classe transport 5](#_bookmark12)

[La classe EntitéeReseau 5](#_bookmark13)

# Description du logiciel

Notre logiciel se veut etre une simulation de connexion et de transfert de données vers un ordinateur distant. Le système local et divisé en une couche transport et une couche réseau. La liaison de données et l’ordinateur distant sont simulés complètement par la couche réseau. Ces composantes seront expliquées plus en détail ultérieurement. Chacune de ces couches communique avec la couche qui lui est supérieure et la couche qui lui est inférieure. La couche transport lis une liste d’instruction dans un fichier, les passe ensuite à des threads de la couche réseau qui fabrique des paquets et les envois au distant qui est en fait simulé par réseau. Distant prends les paquets, inscrit la donnée reçue dans un fichier et répond adéquatement à la couche réseau, en plus de garder un log des transactions. Réseau décide ensuite de conserver ou fermer les connexions avec le distant au besoin, et indique à transport lorsqu’une connexion est fermée. Transport note aussi les ouvertures et fermetures de connexion dans un fichier.

# Instructions spéciales

Il y a peu d’instructions spéciales pour exécuter le programme, il suffit de regarder le terminal pour avoir une confirmation de chaque pression sur le bouton et de le lancer a partir de la classe main et de choisir l'option demo file qui vas se charger dans la vue S\_lec puis d'appuyer sur le boutton Commencer pour tout se fait tout seul. On peut ensuite constater ce qui s’est passé à l’aide des différents fichiers et de leur vue dans l'interface.

Il est normal que le programme prenne une dizaine de secondes à s’effectuer, car il doit parfois attendre qu’un thread ne finisse.

Les fichiers textes son situés sous :

src/Couche/Session/S\_lec.txt ;Couche/Session/S\_ecr.txt;Couche/ LiasonDonnee/L\_ecr.txt;Couche/LiasonDonnee/L\_lec.txt

# Description des fichiers.

Les différents fichiers textes que nous utilisons sont les suivants :

S\_lec :

Contient les différentes instructions que nous souhaitons envoyer au distant.

S\_ecr :

Contient les logs de la couche transport. Les trames sont encore une fois séparées en deux, commençant par le nom du programme, suivi du log en tant que tel.

Les deux opérations que note transport sont création d’une connexion, et terminaison ou encore échec d’une connexion.

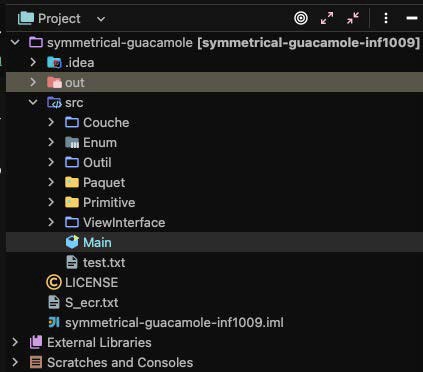
L\_ecr :

Ce fichier correspond à ce que reçoit l’application sur la machine distante, Il n’inscrit que le contenu des paquets de data reçus, précédés du numéro de la connexion qui les a envoyés.

L\_lec :

Ce fichier correspond au fichier log de la liaison de données. Le distant note toutes les transactions dans ce fichier. Il commence encore une fois par le type de paquet suivit par le numéro de la connexion, et de la raison.

# Vue d'ensemble du projet



# Description des classes

Les classes de paquet

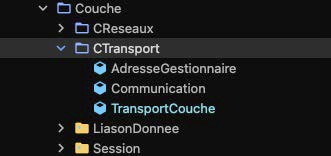
Tous les types de paquet sont représentés par une classe distincte qui hérite de la classe générale Paquet se se trouvent dans le dossier Paquet



La classe des Primitives

La classe Primitives contient toutes les constantes nécessaires au bon fonctionnement du programme et se trouvent dans le dossier Primitives

Le dossier CTransport

Il contient les classes necessaires pour le transport

1. class AdresseGestionnaire permet de gestion des adresse class Communication
2. *class Communication*

qui presenter une communication entre deux extremite,

les deux extremite peuvent etre deja connecte ou non connecte.

La classe main

Le programme commence par L'ouverture de l'interface utilisateur



Le dossier CReseau

Graphical user interface

Description automatically generated

1. *class Connexion*

*un classe qui presenter un connexion entre deux extremite*

*p(s) et p(r) sont les numéros de séquence des paquets représentant respectivement le numéro  
// du paquet envoyé et le numéro du prochain paquet attendu en réception.  
// ici, je delcare les numero de séquence en type demical,  
// Et je vais les convertir en binaire quand je construire les paquets*

1. classe Reseau

*une classe qui presenter la couche Reseaux*

*// Une List de controle de couche Reseaux  
private final ArrayList*<*Connexion*> TCR;  
  
*// couche Superieure -> couche Transport  
private TransportCouche* transport;  
  
 *un counter ,qui va presenter le numero de connexion (auto-increment)*  
*son couche inferieure -> couche LiaisonDonnee  
private final LiaisonDonnee* liaisonDonnee;

*la methode qui permet de lire les primitive envoye par la couche transport par la methode*lire\_de\_transport *si c'est un primitive NConnectReq  
  
si c'est un primitive NDataReq*   
  
 *lancer le processus de transfert des donnees  
si c'est un primitive NDisconnectReq*   
  
 *liberer les resource* TCR.remove(connexion);  
  
 *envoyer le paquet Indication Liberation avec le message suivant :*"ET terminer la liaison de donne  
   
  
   
  
 *// Une methode qui evoyer les primitive et id vers la couche transport par* ecrire\_vers\_transport

*le Fournisseur reseau peut refuser ou accepter le demande de connexion de la couche transport  
// ce processus est aleatoire par la methode :  
 private void* decisionFournisseurAleatoire(  
*// si l’adresse de la station source est un multiple de 27, le fournisseur refuse la connexion.  
 sion accepter , et envoyer le paquet appel  
 //creer un nouveau object Connexion,et l'ajouter dasn la table de controle*  
 *//envoyer le paquet appel, en utilissant le service liason donnee(ecriture dans le fihcier L\_ecr)*

*// une methode qui gerer les transfert des donnees  
 public void* gestionTransfertDonnee  
  
  
 *//une methode qui envoyer un paquet en utlisant le service de liason de donne  
 private void* envoyerPaquet  
 *//une methode qui permet de recevoir le paquet de dianstant(en simulant par une processus alelatoire)  
 public void* recevoirPaquetDistant  
  
  
 *//une methode qui permet de trouver un Connexion par numero de connexion  
 public Connexion* findConnexionByNumeroConnexion( *//une methode qui permet de trouver un Connexion par l'identifiant d'extremite  
 public Connexion* findConnexionByid(

Le dossier Liaison de donnnees

Graphical user interface, text, application

Description automatically generated

*class LiaisonDonnee classe qui presenter la couche de liason de donnee*

*et les fichiers L\_ecr et L\_lec*

Le dossier Liaison de Session

Graphical user interface, text, application

Description automatically generated

Le dossier Enum



Le dossier Interface

Text

Description automatically generated

Comment utiliser :



Appuyer sur demo file qui vas charger dans la vue le fichier de demo

Graphical user interface, application, Word

Description automatically generated

Et ensuite appuyer sur commencer pour voir la suite :

Text

Description automatically generated with low confidence

Le boutton reset permet d’effacer toute les vues et eventuellement les fichiers

Le boutton Regenerer permet de Generer de nouvelles donnees S\_lec

Graphical user interface, Word

Description automatically generated Graphical user interface

Description automatically generated with low confidence

Resultat de la transmission :

Graphical user interface, table

Description automatically generated with medium confidence

Cette section permet d’ouvrir dans l’éditeur de texte et d’afficher les traces

Graphical user interface, text, application

Description automatically generated