



**Institut Supérieur Industriel de Bruxelles**

Rue Royale 150 — 1000 Bruxelles  
Rue des Goujons 28 — 1070 Bruxelles  
[www.isib.be](http://www.isib.be)

---

**Enseignement Supérieur de Type Long et de Niveau Universitaire**

Haute Ecole Paul-Henri Spaak  
Catégorie Technique

## **Rapport d'activités d'immersion en entreprise**

Rapport présenté en vue de l'obtention du grade  
de Bachelier en Sciences de l'Ingénieur Industriel

---

Groupe Génie Electrique  
Année Académique 2014-2015

---

**M. Abderrazzak El Hirach**

Stagiaire chez IRISIB  
150, rue royale  
1000 Bruxelles

Numéro : ISIB-ELIN-RSt-15/03  
Date : 28-05-2015  
Classification : tout public

---

# Résumé

L'Institut de Recherche de l'Institut Supérieur Industriel de Bruxelles (IRISIB) où j'ai eu le plaisir et l'honneur d'effectuer ce stage, contient un département électronique-informatique qui réalise des projets de recherche et développement dans différents domaines scientifiques. De plus, ce département donne des formations dans le domaine informatique pour des étudiants ingénieurs, ainsi que pour différentes entreprises en Belgique.

Au vue de la taille du département, son laboratoire réseau nécessite un grand nombre des routeurs et commutateurs permettant d'interconnecter ses différents locaux. La gestion de tous ces équipements se fait grâce à quelques routeurs Cisco de série 2600, ceux-ci agissent comme étant des serveurs terminal entre l'administrateur et le reste des routeurs et commutateurs. Ces serveurs portent le nom de COMSERVER au sein de laboratoire et ils utilisent chacun des câbles « CAB-OCTAL-ASYNC » connectés du côté des serveurs aux ports asynchrones HWIC16-A, tandis que l'autre extrémité de chaque câble OCTAL fourni 8 sorties console RJ 45 connectés à leur tour aux ports console des routeurs. Avec ces serveurs terminal, l'administrateur n'a besoin que d'un seul terminal ouvert pour atteindre la console de différents routeurs du laboratoire.

Les formateurs qui donnent cours Cisco au laboratoire, désirent améliorer la manière dont les cours pratiques sont dispensés. Par conséquent, ils suggèrent de mettre à la disposition des étudiants un terminal qui reçoit les commandes tapées par le formateur sur sa fenêtre console. Cependant, la communication par serveur terminal comporte quelques inconvénients au sein du laboratoire, parmi lesquelles se trouve principalement l'impossibilité de multiaccès aux équipements, à cela s'ajoute que lorsque l'on veut accéder à un des routeurs à travers les serveurs terminal, que ce soit par Telnet ou SSH, il faut absolument connaître son numéro de ligne que le serveur lui a assigné.

L'objectif de ce stage est de trouver une solution à cette problématique en mettant en place un environnement clients-serveur, qui doit permettre d'établir un canal de communication entre un compte privilégié, c-à-d un administrateur capable de configurer les équipements Cisco (Routeur, commutateurs, ponts...) sur son terminal, et des utilisateurs simples qui se contentent de recevoir des commandes sur leur terminal. Cette tâche consiste donc à développer une application jouant le rôle du serveur terminal en offrant les solutions suivantes :

- Afficher les différents routeurs connectés (par des câbles console EIA/TIA-232-RJ45) à la machine où elle est lancée.
- Rediriger les terminaux TCP des utilisateurs à distance vers les différents routeurs.
- Diffuser la réponse provenant des routeurs vers un canal de communication.

Dans ce stage, nous avons choisi de programmer l'application en langage Python vu qu'il comporte plusieurs avantages, dont celui de contenir un ensemble de modules et de scripts qui supporte la connexion série avec les routeurs.

# Remerciement

Ce stage a été réalisé avec succès grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je voudrais tout d'abord adresser ma gratitude à mon promoteur industrielle Mr. Jacques Tichon pour sa disponibilité et son aide tout au long de la période de stage. Je remercie également toute l'équipe de l'IRISIB pour leur accueil, leur esprit d'équipe et en particulier Mr DE JONGH Steve, qui m'a fourni les outils nécessaires à la réussite de ce stage, et qui m'a aussi aidé à la reconstruction du programme dans la méthode adéquate. Je remercie aussi le préparateur de laboratoire Mr « ARNAUD DELAHAYE » qui n'a jamais hésité de me faire passer le matériel et les équipements requis.

Je tiens à remercier spécialement mon superviseur académique Mr Damien Grobet qui m'a beaucoup aidé dans la recherche de stage, et qui m'a permis de trouver ce stage que j'ai tellement apprécié. Son écoute et ses judicieux conseils ont vraiment contribué à enrichir ma réflexion lors de la période de stage.

Sincèrement, je suis très reconnaissant pour le soutien moral et financier que mes chères parents à l'étranger ainsi que ma chère sœur m'ont fait durant mon cursus scolaire. Je leur remercie également pour la confiance déposée en moi et leur importants mots qui m'ont encouragé.

Les derniers remerciements vont aux chères amis et collègues qui m'ont apporté leur support moral. Un grand merci pour mon co-stagiaire « Van Goettem Maxime » pour son aide concernant les essais au laboratoire.

# Table de matières

1 Introduction.....	1
2 L'Institut de Recherche de l'Institut Supérieur Industriel de Bruxelles (IRISIB).....	2
2.1 Présentation et secteur d'activités.....	2
2.2 Situation Géographique.....	3
2.3 Aperçu historique et réalisations.....	4
2.4 Les départements spécialisés.....	4
2.5 Département électronique - informatique.....	5
3 Présentation du cadre du stage.....	8
3.1 Le contexte du stage.....	8
3.2 Les équipements Cisco et l'interface en ligne de commande (CLI).....	8
3.2.1 L'accès au CLI par un câble console.....	9
3.2.2 L'accès au CLI par Telnet et SSH.....	11
3.2.3 L'accès par un serveur terminal.....	11
3.3 La situation actuelle du réseau d'IRISIB et la problématique posée.....	12
4 Les solutions envisageables et les tâches effectuées.....	13
4.1 Les solutions proposées pour les problématiques posées.....	13
4.2 Notion d'application clients-serveur.....	15
5 Python et les outils de développement.....	15
5.1 Le langage Python.....	15
5.2 La modélisation UML.....	15
5.3 Le choix de l'environnement de développement intégré « Pycharm ».....	16
5.4 Une brève description de l'application.....	17
5.5 les problèmes rencontrés.....	18
5.6 Le teste de l'application.....	19
6 La simulation de la topologie réseau alternative.....	20
6.1 Gns3 et Virtual serial port emulator.....	20
7 Le matériel utilisé et la mise en place réelle.....	21
7.1 Le matériel utilisé.....	21
8 Conclusion générale.....	24

# 1 Introduction

Le stage d'immersion en entreprise organisé en 3<sup>ème</sup> année de Bachelier par l'Institut Supérieur Industriel de Bruxelles, est vraiment une grande opportunité pour parfaire nos connaissances et nous permettre d'obtenir une première expérience dans le milieu professionnel.

En tant qu'étudiant dans cet établissement de la filière « Génie Électrique » et notamment la finalité « Informatique », parvenir à effectuer un stage dans ce domaine est le choix approprié, en particulier si cela me permet de faire une série des tâches qui correspondent parfaitement à mes futures inspirations professionnelles.

Le stage a été effectué du 27 Avril 2015 jusqu'au 29 Mai de même année, à L'Institut de Recherche de l'Institut Supérieur Industriel de Bruxelles (IRISIB). Son but consiste à améliorer l'accès aux équipements réseau et moderniser la manière dont les cours Cisco sont dispensés dans le laboratoire réseau au département électronique- informatique.

Dans ce présent rapport, nous allons dans un premier temps, présenter le secteur d'activités de l'IRISIB, ainsi que ses collaborations et ses participations dans le domaine industriel en Belgique. Ensuite, nous allons focaliser l'attention sur la structure du département électronique-informatique, en donnant quelques projets de recherches déjà réalisées et d'autres actuellement en cours de réalisation.

Dans un deuxième temps, nous allons présenter le contexte du stage en expliquant les problématiques posées et en donnant quelques illustrations sur les termes et les équipements utilisés. Ensuite, nous allons donner une brève description de l'application développée en mettant l'accent sur le langage de programmation Python utilisé.

Enfin, dans un dernier temps, nous citerons les tâches effectuées ainsi que les méthodes adoptées pour mettre en place le système alternatif, et nous exposerons aussi les problèmes rencontrés et les solutions trouvées pour réaliser cette application et mener à bien ce stage.

## 2 L'Institut de Recherche de l'Institut Supérieur Industriel de Bruxelles (IRISIB)

### 2.1 Présentation et secteur d'activités

L'IRISIB est une association sans but lucratif et un centre de recherche interdisciplinaire de l'Institut Supérieur Industriel de Bruxelles (Belgique), il est agréé par le SPF Finances et le SPF Programmation de la Politique Scientifique (AR du 10 décembre 2008, MB du 30 décembre 2008). Son but principale est de développer les activités de la recherche et de formation, en vue d'enrichir les connaissances dans différents domaines scientifiques, de même que contribuer à l'amélioration des pratiques au sein des organisations.

Ses sept départements spécialisés cherchent continuellement à optimiser les activités de recherche et développement dans la mesure où l'on trouve à la tête de chaque département un administrateur responsable et des spécialistes motivés et expérimentés. De plus, ses départements contribuent de façon importante aux travaux des unités de recherche de la Haute Ecole PH Spaak .

Dans son dernier rapport d'activité en mars 2013, IRISIB comptait 27 membres effectifs et 7 membres adhérents. Son conseil d'administration est constitué de 8 membres dont l'organisation est représentée sur la Figure 1 :

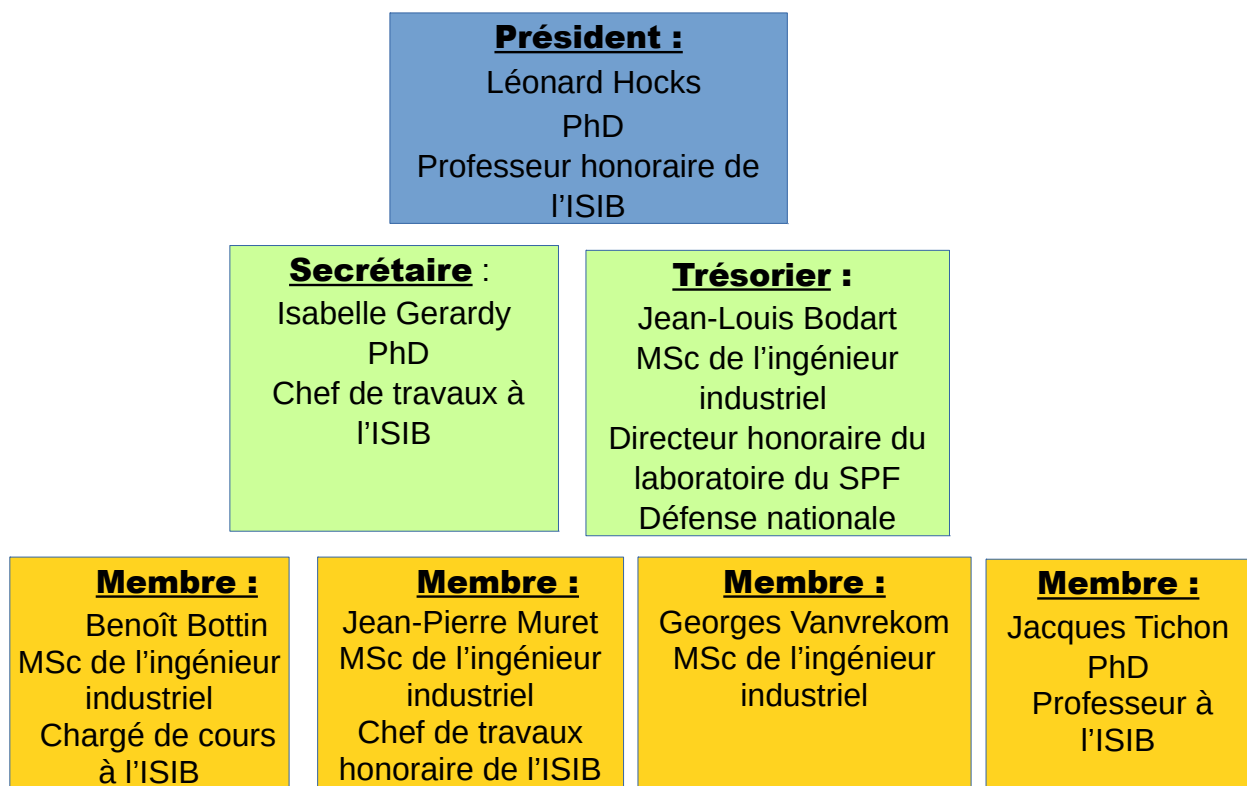


Figure 1 - L'organigramme du conseil d'administration d'IRISIB depuis mars 2013.

Par ses activités, IRISIB est répertorié dans les groupes de compétences Energie, Skywin, Mécatech, Biowin et Wagralim. Il est également repris dans le répertoire général des compétences ainsi que dans le rapport annuel de la cellule ADISIF sur la recherche dans les instituts supérieurs industriels.

IRISIB participe également aux travaux de deux autres asbl : la cellule ADISIF (interface entre la région wallonne et les instituts supérieurs industriels) où Isabelle GERARDY représente IRISIB en son sein, et INDUTEC (interface entre la Région de Bruxelles-Capitale et les instituts supérieurs industriels) dont la mission principale est la valorisation de la recherche dans les instituts supérieurs industriels et les centres de recherches associés.

Le secteur d'activités de l'IRISIB ne se limite pas à faire des projets, mais ils donnent aussi des formations et des contrats de services pour des personnes extérieures de l'IRISIB. Cette année, Jean-Pierre Muret et Olivier Debia de département électricité ont formé des industriels dans le domaine des automates programmables et cela en utilisant les nouveaux programmes fournis par Schneider Electric. Aussi, le département électronique-informatique a donné des formations d'administrateur – certifié CCENT ainsi que des formations techniques pour informaticiens. Notons également que le département mécanique donne des formations CAO/DAO sur le logiciel Solid Works pour le centre CPE-HN. Le département nucléaire a, quant à lui, organisé en collaboration avec l'IRE, la formation des experts en radioprotection (120 h) [8].

## 2.2 Situation Géographique

Le siège sociale de l'IRISIB est situé dans les locaux de l'ISIB, rue Royale 150 à 1000 Bruxelles. Cet endroit est une des principales artères de Bruxelles-ville. Aujourd'hui, cette place est considéré comme un lieu centrale et stratégique pour les instituts supérieurs pédagogiques, surtout en terme de facilité d'accès par les moyens de transport en commun. Effectivement, le siège se trouve juste devant l'arrêt de tram Congrès et quelques dizaines de mètres à côté de deux arrêts de métro Botaniques et Madou. La photo sur la Figure 2 présente le siège central d'IRISIB.



Figure 2 - Le siège social d'IRISIB à Bruxelles [8].

## 2.3 Aperçu historique et réalisations

IRISIB a été constitué le 2 octobre 1970 (MB du 5 novembre 1970). L'appellation IRISIB existe officiellement depuis mars 2000. Auparavant, l'asbl portait le nom ISRAIN (Institut supérieur de Recherche appliquée pour les Industries nucléaires).

Les différents départements de l'IRISIB ont participé aux grandes compétitions internationales reconnues telles que le congrès «Image&3D», «Laval Virtual», «Robotics in Education», la conférence «ENC 2012» à Manchester, le workshop «INSINUME» en 2012 à Bruxelles, «General Meeting» du réseau PRIME et beaucoup d'autres. À cela s'ajoute la collaboration de ces départements avec d'autres instituts et universités supérieurs aux nombreuses événements qui ont lieu partout dans le monde tout au long de l'année. Notons, par exemple, l'aide aux pays en voie de développement qui a eu lieu à L'ESP de Dakar en collaboration avec le wbri (relations internationales wallonie – bruxelles), aussi une autre collaboration s'est amorcée entre l'hôpital Erasme et le département nucléaire concernant des techniques de fluorescence X [8].

Parmi les travaux de recherche et développement et les différents projets remarquables qui ont été réalisés et qui sont en cours de réalisation par les départements de l'IRISIB, on peut souligner les suivantes :

- Les travaux de recherche appliquée effectué par Isabelle Windal de département chimie – gestion de l'environnement, pour l'entreprise Roscic dans le domaine des revêtements de surface de pièces métalliques.
- Le développement de la logique floue avec la nouvelle version du logiciel UNITY V7 par Olivier DEBIA et Nadir BAIBOUN de département électricité.
- Les projets First de département électroniques – informatiques tels que les projets IMAGINE, COLIBRI, et SAMIFIS ( nous allons les expliquer dans le point 2.5).
- Le projet First MICOS (l'algorithme de déconvolution modifié) réalisé en 2011 par le département nucléaire.

## 2.4 Les départements spécialisés

Comme nous l'avons indiqué précédemment, le centre d'IRISIB est structuré de 7 départements, où chaque département est spécialisé dans un domaine scientifique bien particulier et possède d'un administrateur responsable. Sur la Figure 3, nous donnons le nom de chaque département ainsi que son administrateur responsable :

<b>Département électricité</b> Administrateur responsable : Jean-Pierre Muret	<b>Département chimie et gestion de l'environnement</b> Administrateur responsable : Léonard Hocks	<b>Département ISIB</b> Administrateur responsable : Karin Van Loon	<b>Département des services généraux</b> Administrateur responsable : Jean-Louis Bodart
	<b>Département mécanique</b> Administrateur responsable : Benoît Bottin	<b>Département électronique et informatique</b> Administrateur responsable : Jacques Tichon	

Figure 3 - Les spets département spécialisés d'IRISIB.



- **Département des services généraux :** Ce département gère toutes les activités de comptabilité et de trésorerie de l'asbl.
- **Département ISIB :** Ce département finance partiellement des actions publicitaires visant à mieux faire connaître l'ISIB et ses activités d'enseignement ainsi que des journées de rencontre entre les étudiants et les entreprises (journées « entreprises »). Il participe également au financement des bourses Erasmus pour les étudiants de 2e Master.
- **Département chimie et gestion de l'environnement :** Le département a acquis une expertise dans les domaines suivants : abatement de la pollution de l'air et de l'eau, traitement des déchets, analyse du cycle de vie des produits, procédés et services. Il effectue également des prestations pour les entreprises privées dans le domaine de la gestion de l'environnement.
- **Département électricité :** Ce département est un centre agréé par l'entreprise Schneider – Electric. Il effectue principalement des formations pour le personnel d'entreprises privées. Ces formations se font avec du matériel de l'entreprise Schneider-Electric. Les domaines développés sont l'utilisation des automates programmables, la régulation et la logique floue.
- **Département électronique et informatique :** Ce département où nous avons effectué ce stage, développe de nombreux projets dans différents domaines de l'informatique et de l'électronique (le point 2.5 sera consacré pour expliquer les activités de ce département).
- **Département mécanique :** Ce département a acquis une expertise R&D dans les domaines de l'aéronautique, de la conception et de la fabrication assistées par ordinateur (CFAO) ainsi que du dimensionnement, du prototypage et d'essais d'appareillages et de systèmes mécaniques. Ces activités sont financées par les entreprises et, dans le cas des PME, par l'intervention partielle de la Région Wallonne (chèques technologiques).
- **Département nucléaire :** Ce département a acquis une expertise dans le domaine de la protection contre les radiations ionisantes : Technologies de déconvolution spectrale, la caractérisation des spectres de RX provenant de tubes à RX utilisés dans les applications médicales, et beaucoup d'autres expériences dans le domaine physique nucléaire [8].

## 2.5 Département électronique - informatique

Étant donné que ce stage a été effectué dans le département électronique-informatique de l'IRISIB, nous consacrons ce point pour donner plus précisément sa structure organisationnelle ainsi que son équipe fonctionnelle. Ensuite, nous allons citer ses activités de recherche et développement, y compris les projets réalisés et ceux qui sont déjà en cours de réalisation.

L'électronique et l'informatique est l'un des domaines les plus importants dans le développement technologique et industriel en Belgique. Dans le Rapport belge en matière de science de 2010, la Belgique se place parmi les pays les plus importants de l'UE27 en termes d'innovation et de technologie, et cela grâce aux principaux acteurs de la sphère scientifique et de la recherche en Belgique qui sont essentiellement les établissements d'enseignement supérieur (EES), les universités ainsi que les hautes écoles [6].

À cet égard, les activités d'IRISIB sont partiellement financées par la Région Wallonne dans le cadre des programmes First Hautes Ecoles. Celui-ci est l'un des objectifs du gouvernement wallon en matière de recherche et d'innovation technologique, où il permet aux chercheurs de faire l'apprentissage du milieu industriel tout en gardant leur statut universitaire [9].

Les membres effectifs de département constitue une équipe expérimenté qui développe de nombreux projets de R&D dans différents domaines de l'informatique et de l'électronique, Il se compose des professeurs, chercheurs First, formateurs et assistants ingénieurs. Le Tableau 1 donne le nom, la fonction, et l'expérience de chaque membre de l'équipe :

 <p><b>DEGEEST Alexandra Ir.</b> Maître assistante Machine Learning Automatique &amp; Robotique</p>	 <p><b>DE JONGH Steve</b> Formateur et Administrateur Réseau Certification Cisco • CCNA, CCNP Route</p>	 <p><b>GARCIA Salvador M.Sc.Ing</b> Chef de travaux Électronique générale Électronique num Bases de données</p>	 <p><b>GIOT Rudi Ir.</b> Chargé de cours Traitement de signal Génie logiciel Électronique numérique</p>
 <p><b>GROBET Damien M.Sc.Ing</b> Maître Assistant Robotique &amp; Vision Industrielle Informatique générale</p>	 <p><b>LAFFINEUR Ludovic M.Sc.Ing</b> Chercheur FIRST Monitoring réseau • Programmation C++ Sonification de données</p>	 <p><b>MATTENS Jean- Michel M.Sc.</b> Chef de travaux Mathématiques Java C++</p>	 <p><b>NGUYEN Mai Quang Minh M.Sc.Ing</b> Chercheur FIRST Programmation Multi-tactile Reconnaissance d'objets/personnes</p>
 <p><b>SEVERS Thierry M.Sc.Ing</b> Chef de travaux Génie logiciel Informatique générale Programmation Java</p>	 <p><b>TICHON Jacques Dr.</b> Professeur • Electronique générale Informatique Industrielle Systèmes embarqués</p>	 <p><b>RODRIGUES E SOUSA Abilio Sergio M.Sc.Ing</b>  Chercheur FIRST Traitement d'images Optimisation GPU</p>	

Tableau 1- Les membres d'équipe du département électronique- informatique d'IRISIB [8].

Parmi les projets First du département, on trouve le projet COLIBRI (Conception et développement de bornes interactives multimodales) développé par Minh Nguyen en collaboration avec Rudi Giot. Ce projet consiste à concevoir et développer un ensemble d'outils pour faciliter la création d'applications pour des bornes interactives « multi-touch » capables de reconnaître des objets ou/et des personnes et d'interagir avec ces derniers de plusieurs manières.

Sergio Rodrigues est un autre chercheur FIRST au sein de département. Celui-ci développe dans les laboratoires d'EN/INFO, en collaboration avec Jacques Tichon et Rudi Giot, le projet IMAGINE ( IMplantation d'AlGorithmes de traitement d'Images dans uN système hardware ouvert). IMAGINE a pour objectif d'accélérer les algorithmes de traitements d'images par ordinateur. Pour y parvenir, l'architecture de calcul des GPU NVIDIA (CUDA) est utilisée, offrant ainsi une très grande puissance de calcul.

Jean-Guillaume Louis est le 3e chercheur FIRST du département. Il travaille, en collaboration avec Rudi Giot, au projet SAMIFIS : Système d'analyse multicritère intelligent pour filtrage de spams. Un serveur anti spam est en cours de développement. Le filtrage s'effectue à l'aide d'un système de décision multicritère de type ELECTRE.

Le département a également développé des applications de l'électronique dans le domaine des Arts et des Sciences. Un laboratoire spécialisé dans ce domaine a été créé en partenariat avec des artisans. Ce laboratoire porte le nom de LARAS (Laboratoire des recherches en arts et sciences) et il possède un site web vitrine de ses activités de recherche « [www.laras.be](http://www.laras.be) ».

Le département est doté des équipements sophistiqués ainsi qu'un laboratoire réseau le plus complet en Belgique [8]. Nous citons ici quelques équipements des plus importants :

- Un Humanoid Robot NAO
- Lan Base et un Cisco 2911
- Cameras traitement d'images (Projet FIRST IMAGINE)
- Un « super » PC FIRST COLIBRI
- Un écran Samsung dans le cadre du projet FIRST COLIBRI
- « Super Computer » destiné au projet SAMIFIS (programme FIRST)
- Arduinos et matériel divers
- Matériel audio pour le labo Laras Rme Fireface Ucx
- Matériel robotique « Generation Robots »
- Rayonnages destinés au magasin du labo EN/INFO
- Une imprimante 3D
- Une gravure de PCB

## 3 Présentation du cadre du stage

### 3.1 Le contexte du stage

Dans le cadre de ce stage, il nous a été demandé de développer une application clients-serveur permettant de créer un canal de communication entre des clients TCP sur le réseau et les différents routeurs et commutateurs Cisco installés dans la salle informatique. L'implémentation d'une telle application dans le laboratoire, peut en effet offrir des solutions remarquables en termes d'administration de réseau, ainsi qu'une mise en œuvre plus rigoureuse des équipements en matières d'accessibilité.

Avant de commencer à illustrer en détail la problématique et à exposer les solutions suggérées, il est nécessaire de comprendre quelques notions de bases sur les équipements Cisco, de même que définir le matériel et les nouveaux termes que nous avons utilisé tout au long de la période du stage.

### 3.2 Les équipements Cisco et l'interface en ligne de commande (CLI)

Aujourd'hui, Cisco est l'une des entreprises les plus importantes dans la fourniture des équipements de communication et du matériel réseau professionnel. Parmi les équipements d'un réseau local (LAN) fourni par Cisco, on trouve essentiellement les routeurs et les commutateurs. Les routeurs sont des éléments informatiques qui opèrent au niveau de la couche 3 du modèle OSI<sup>1</sup>, où ils permettent de diriger les paquets entre les réseaux. Tandis que, les commutateurs sont des éléments de la couche 2, ils sont capable de rechercher l'information sur les trames que reçoivent de la part des ordinateurs du réseau. Le but des commutateurs est de concentrer la connectivité des équipements tout en rendant la transmission des données plus efficace.

Le système d'exploitation de la plupart des équipements réseau produit par Cisco est appelé IOS (Internetwork Operating System). Celui-ci est menu d'une interface en ligne de commande (CLI) permettant aux utilisateurs d'utiliser un émulateur terminal qui accepte un texte tapé par l'utilisateur.

Comme montré sur la Figure 4, pour configurer un routeur, nous pouvons accéder au CLI à partir de plusieurs lieux externes, y compris les suivants :

- Une console terminal, en utilisant un ordinateur local connecté au port console.
- Par un modem, en utilisant un port auxiliaire (Auxiliary cable).
- Un terminal virtuel, qui permet d'utiliser Telnet ou SSH après avoir installé un routeur sur le réseau.
- Un serveur TFTP (Trivial File Transfer Protocol) sur le réseau.

Les deux méthodes les plus répandues pour accéder au CLI sont par câble console et par le terminal virtuel en utilisant le Telnet ou SSH [3].

---

<sup>1</sup> OSI : **O**pen **S**ystems **I**nterconnection, c'est un modèle architectural représentant les communications réseau sous forme d'un ensemble de couches superposés les uns aux autres.

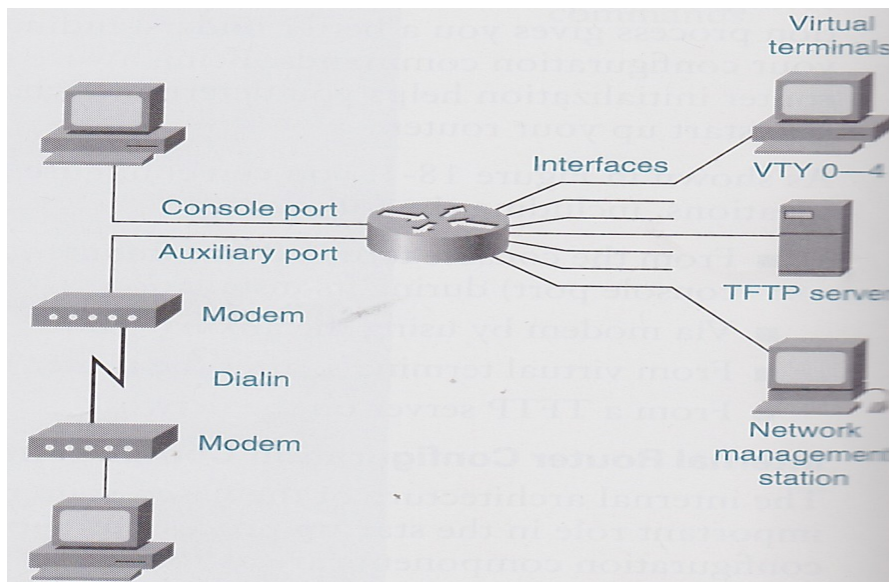


Figure 4 Les différents méthodes pour configurer un routeur Cisco[3].

### 3.2.1 L'accès au CLI par un câble console.

La connexion par le port console ( la Figure 4 ) est une liaison série asynchrone sans contrôle du flux des données, les utilisateurs de cette connexion ont par défaut un accès automatique au (CLI). Ce port fournit un moyen de se connecter à l'équipement même s'il n'est pas encore connecté au réseau.

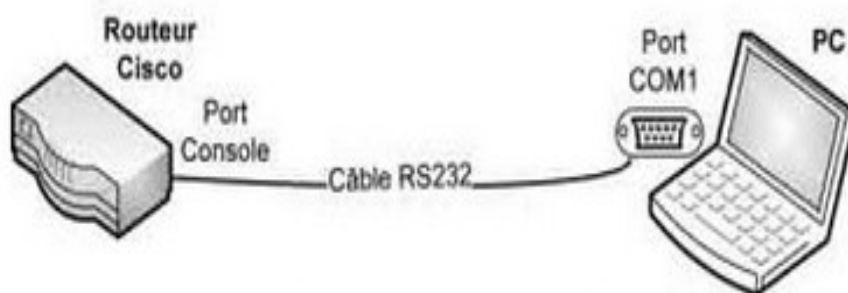


Figure 5 - La connexion console à un équipement Cisco [2].

La transmission série des données est caractérisée par le fait que les bits sont transmis successivement sur un support unique. Nous citons ici quelques exemples des protocoles de communication série : RS-232, RS-485, I2C, USB, FireWire, Fibre Channel, Serial ATA, SPI, UART [2].

Ce type de transmission peut s'opérer suivante deux modes différentes :

- Mode de transmission asynchrone : dans ce mode chaque caractère est séquencé individuellement, un dispositif qui reçoit un caractère asynchrone resynchronise son horloge à l'aide du bit de Start.

- Mode de transmission synchrone : dans ce cas les données ne sont pas compréhensibles qu'à l'aide de l'horloge qui a servi à les séquencer. Lors de l'installation de ce type de liaison, il faut choisir à qui confier la fourniture de l'horloge (un sera DTE et l'autre DCE).

Le câble console (rollover cable) est un câble utilisé pour effectuer une transmission série asynchrone avec le routeur. À la première extrémité de ce câble il y a un connecteur RJ45 à connecter au port console du routeur. En revanche, sa seconde extrémité contient soit un connecteur RS-232 de 9 broches, soit un autre câble RJ45 et un adaptateur 9 broches connecté au port COM du PC (La Figure 6) [7].

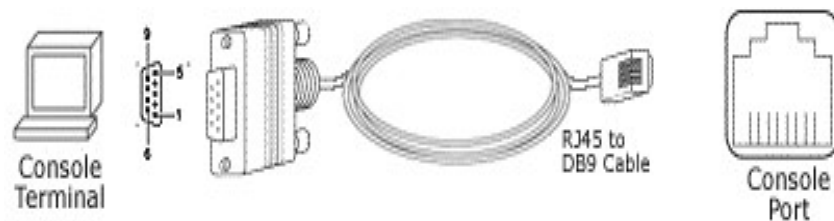


Figure 6 - Un câble console Cisco connecté entre un routeur et un ordinateur [7]

La trame illustrée sur la Figure 7, correspond à une transmission asynchrone irrégulière d'un caractère dont la longueur est 7 bits. La longueur de celui-ci dépend du type de codage utilisé (généralement un codage ASCII) et est limitée à 7 ou 8 bits. Chaque caractère est précédé d'une information indiquant le début de sa transmission (START bit) et terminé par l'envoi d'une information de fin de transmission (bit STOP).

La transmission asynchrone utilise également un bit dit « bit de parité » (le 8ème bit dans le cas du caractère transmis sur la Figure 7), celui-ci contrôle la transmission en calculant à la réception la parité du caractère reçu, ensuite la compare à la valeur de bit transmis par l'émetteur. Elle est paire, si le nombre des bits à l'état logique 1 est pair, et impaire pour un nombre impair des bits à 1[2].

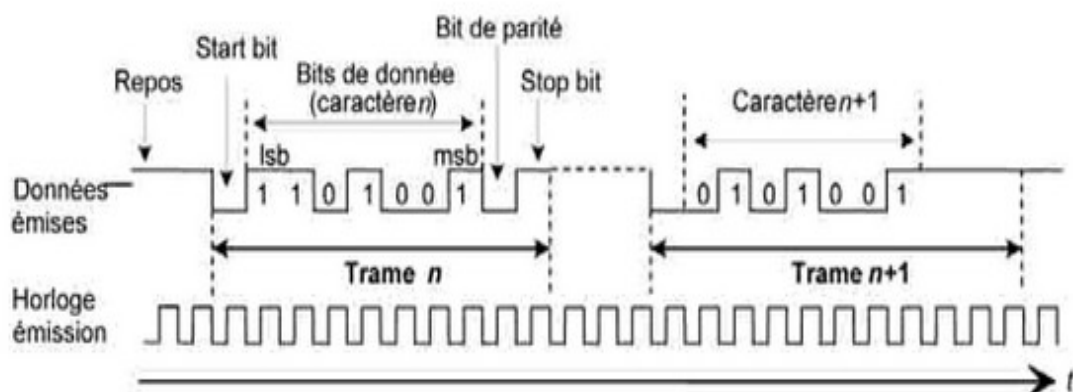


Figure 7 - La transmission d'une trame asynchrone [2].



Pour un PC sur Windows le terminal « HyperTerminal », ou l'application « Putty » qui est couramment utilisée permettent de paramétrer la liaison série et de disposer d'une console pour dialoguer avec le routeur. Les paramètres de cette liaison doivent être les mêmes que ceux du routeur. Nous donnons sur la Figure 8 le paramétrage requis.

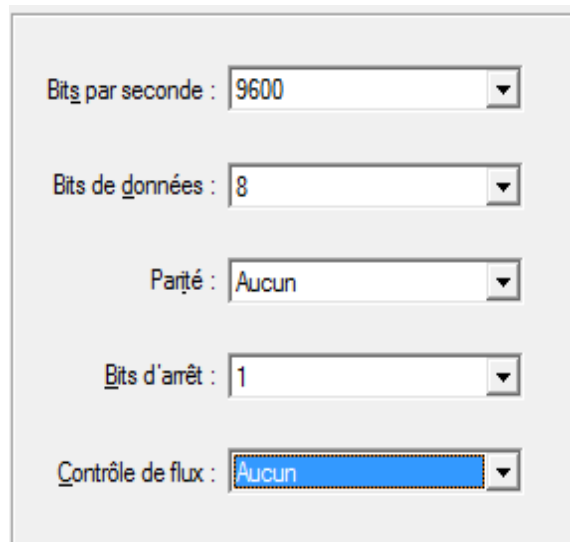
The image shows a configuration window for a serial port connection. It contains five settings, each with a label and a dropdown menu. The settings are: 'Bits par seconde' (Baud rate) set to 9600, 'Bits de données' (Data bits) set to 8, 'Parité' (Parity) set to 'Aucun' (None), 'Bits d'arrêt' (Stop bits) set to 1, and 'Contrôle de flux' (Flow control) set to 'Aucun' (None). The 'Contrôle de flux' dropdown is highlighted with a blue border.

Figure 8- La configuration de l'outil HyperTerminal sous Windows.

### 3.2.2 L'accès au CLI par Telnet et SSH

Le Telnet et le SSH sont des protocoles d'applications qui nécessitent un accès au réseau par un port Ethernet, autrement dit, ils utilisent une adresse IP pour envoyer et recevoir les données. Ces protocoles fournissent un émulateur terminal pour communiquer avec le routeur, et cela en procédant de la même façon que nous l'avons avec le port console. Cependant, dans ce type de communication s'intercalent des séquences de contrôle de flux de données. De plus, ces deux protocoles requièrent une configuration préalable des lignes vty pour pouvoir accéder à distance au routeur. Notons qu'en cas de défaillance d'IOS, il n'est pas possible d'accéder à l'équipement par le port Ethernet. Dans ce cas, l'utilisation du port console est la seule méthode possible pour y accéder [4].

### 3.2.3 L'accès par un serveur terminal

Lorsque la topologie réseau implique un grand nombre des routeurs et commutateurs, la gestion de ces équipements devient de plus en plus difficile. Cela est en grande partie dû au fait que l'administrateur doit ouvrir autant des fenêtres console que le nombre des équipements qu'il voudrait configurer. Heureusement, Cisco fournit la possibilité d'établir un serveur terminal au niveau de ses routeurs en utilisant leur ports asynchrones HWIC16-A (16-Port Async High-Speed WIC). Avec ces serveurs terminal, l'administrateur n'a besoin que d'un seul terminal ouvert pour atteindre la console des différents routeurs de laboratoire.

Un routeur agissant comme étant un serveur terminal utilise des câbles « CAB-OCTAL-ASYNC » connectés à ces ports asynchrones (il peut y avoir un ou plusieurs, cela dépend du série du routeur), l'autre extrémité de chaque câble OCTAL fourni 8 sorties console RJ 45 connectés aux ports consoles de restes de routeurs. La Figure 9 montre comment les ports consoles des routeurs sont connectés au serveur terminal.

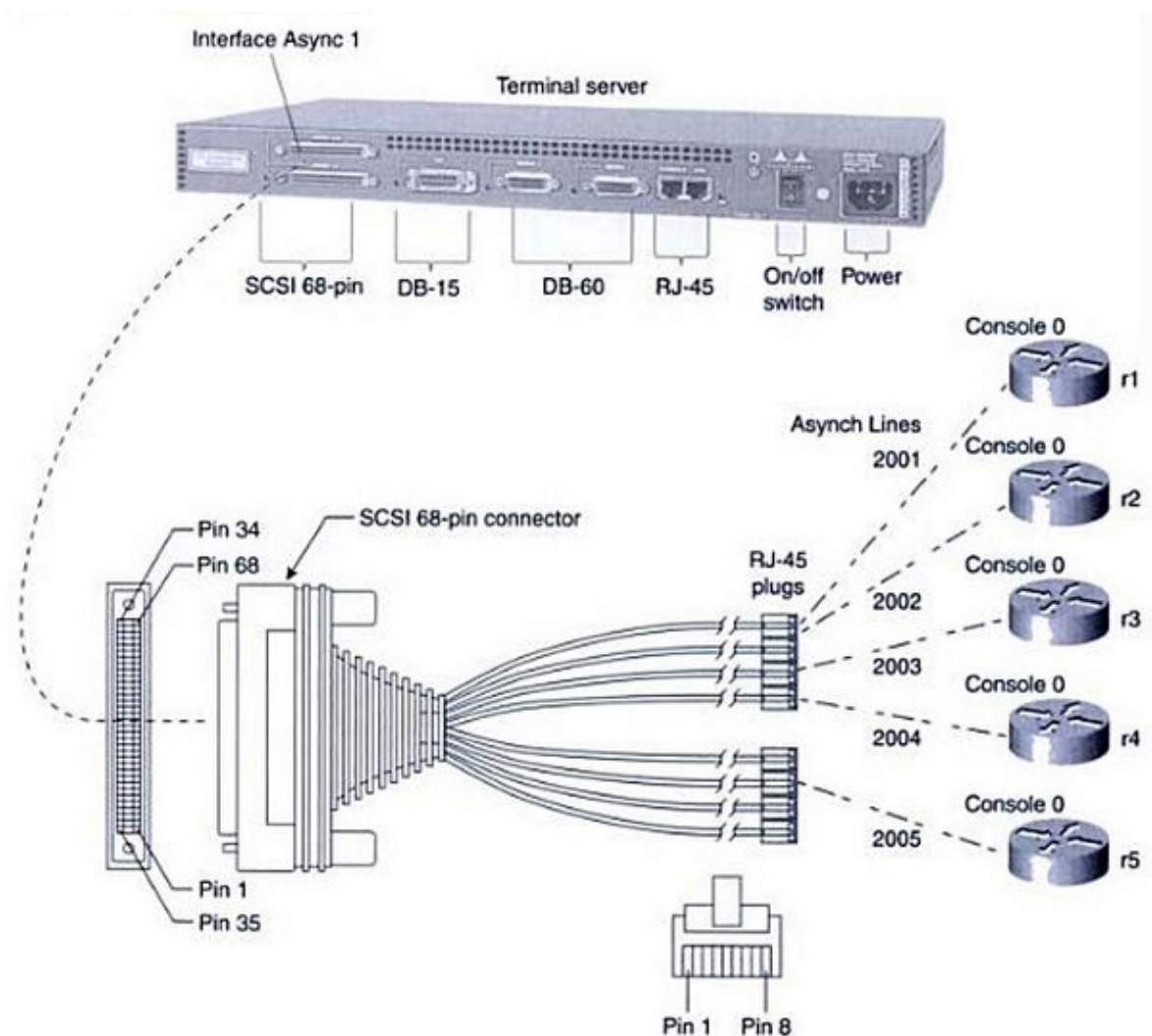


Figure 9- La topologie réseau d'un serveur terminal par un port asynchrone d'un routeur Cisco [1].

### 3.3 La situation actuelle du réseau d'IRISIB et la problématique posée

Au vue de la taille du département électronique-informatique, son laboratoire réseau fait appel à un nombre assez élevé de routeurs et commutateurs. Pour les raisons mentionnées précédemment et afin de mieux gérer le réseau, l'administrateur utilise effectivement des routeurs Cisco de série 2600 comme étant des serveurs terminal. Ces derniers sont identifiés au sein de laboratoire par des étiquettes « COMSERVER » situées sur le côté du boîtier.

Afin de mieux expliquer la topologie réseau actuelle ainsi que la manière dont l'administrateur utilise le réseau pour accéder aux CLI des routeurs, nous avons effectué une simulation de la situation sur le programme « Packet Tracer » comme illustré sur la Figure 10.

D'après la commande utilisée sur le PC0, nous remarquons que pour accéder au CLI d'un des routeurs disponibles via le serveur terminal (en utilisant le protocole SSH), l'administrateur a absolument besoin de connaître son numéro de ligne asynchrone que le serveur lui a assigné (dans



l'exemple du PC0 : on accède au Router0 par la ligne asynchrone 3). De plus, on remarque que le serveur terminal a refusé la connexion du PC1 au même routeur, car il est déjà connecté au PC0. En d'autres termes, l'accès multiple aux équipements n'est pas possible par le serveur terminal.

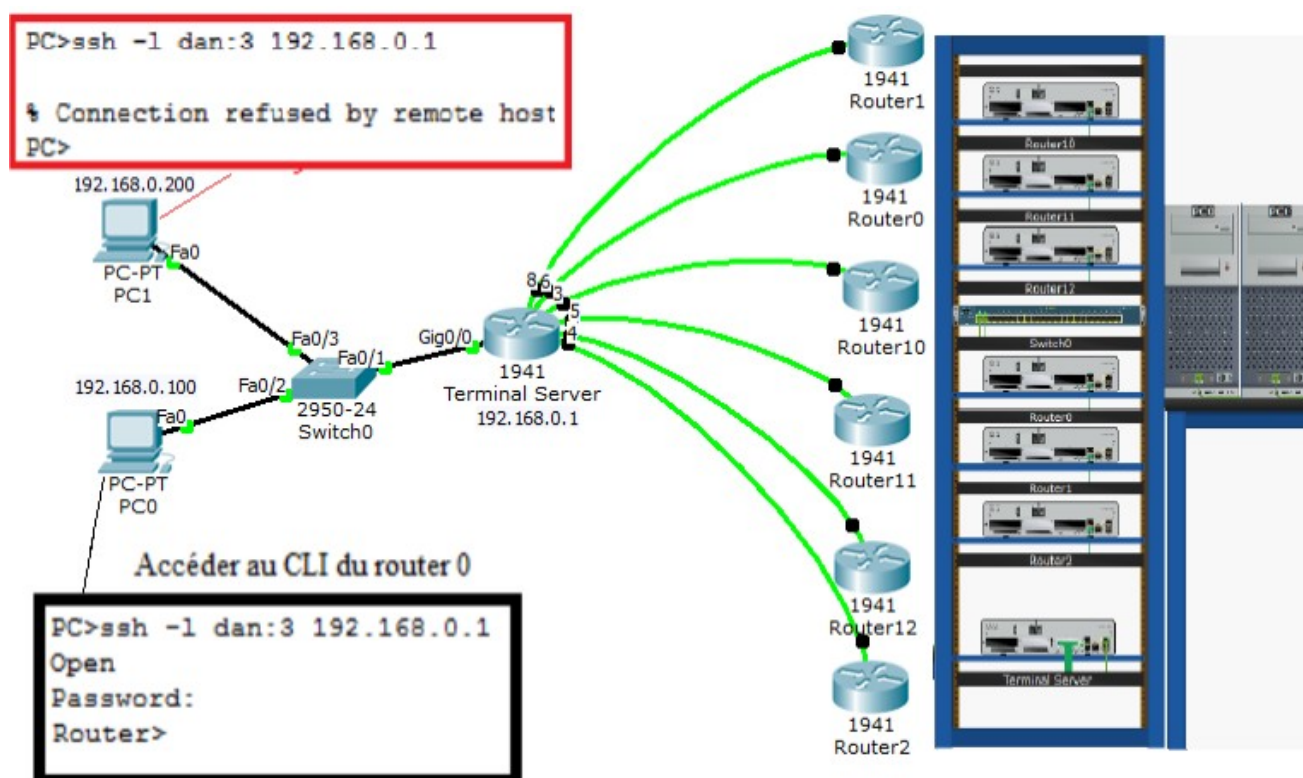


Figure 10 - La simulation sur « Packet Tracer » de la topologie réseau actuelle d'IRISIB.

Il convient de souligner que l'équipe des formateurs réseau au laboratoire désire améliorer la manière dont les cours Cisco sont dispensés. Pour cela, ils suggèrent de rendre l'accès aux équipements réels disponibles aux étudiants, et cela, en mettant à leur disposition un terminal qui reçoit les commandes tapées par le formateur. Cependant, comme nous l'avons vu, le multiaccès aux équipements via le serveur terminal n'est pas possible.

Compte tenu de ces difficultés, l'administrateur du réseau voudrait remplacer les serveurs terminal par une application permettant d'accéder plus facilement à l'ensemble des routeurs. En outre, cette application doit offrir la possibilité d'établir un canal de communication entre l'administrateur qui configure les équipements et des utilisateurs qui se contentent de recevoir des commandes sur leur terminal.

## 4 Les solutions envisageables et les tâches effectuées

### 4.1 Les solutions proposées pour les problématiques posées

Après avoir pris connaissance des problèmes posés par les serveurs terminal et, en particulier, les changements souhaités par les administrateurs du réseau, nous allons donner quelques solutions

techniques envisageables, et bien entendu mettre l'accent sur les tâches effectuées pour élaborer, concevoir et mettre en œuvre un nouveau programme remplaçant plus efficacement les serveurs terminal.

Parmi les solutions envisagées, on retrouve l'idée de développer une application clients-serveur qui permet aux utilisateurs d'accéder aux routeurs à distance via un réseau privé ou internet. Cette application devra fournir les points suivants :

- Un accès sécurisé qui exige des privilèges administrateur. Ceci afin d'assurer que des applications malicieuses ne peuvent pas accéder aux routeurs. Cela sert également à rappeler que nous allons réaliser des actions administratives sur les routeurs qui requièrent notre entière attention.
- Les utilisateurs normaux n'ont pas le droit de configurer les équipements pour des raisons de sécurité.
- Lister les différents routeurs connectés par des câbles console à la machine où elle est lancée.
- Rediriger les terminaux TCP des utilisateurs à distance vers les différents routeurs.
- Diffuser la réponse provenant des routeurs vers un canal de communication.

La Figure 11 montre plus clairement comment l'administrateur du réseau et les utilisateurs simples doivent se connecter aux routeurs via l'application clients-serveur alternatif au serveur terminal.

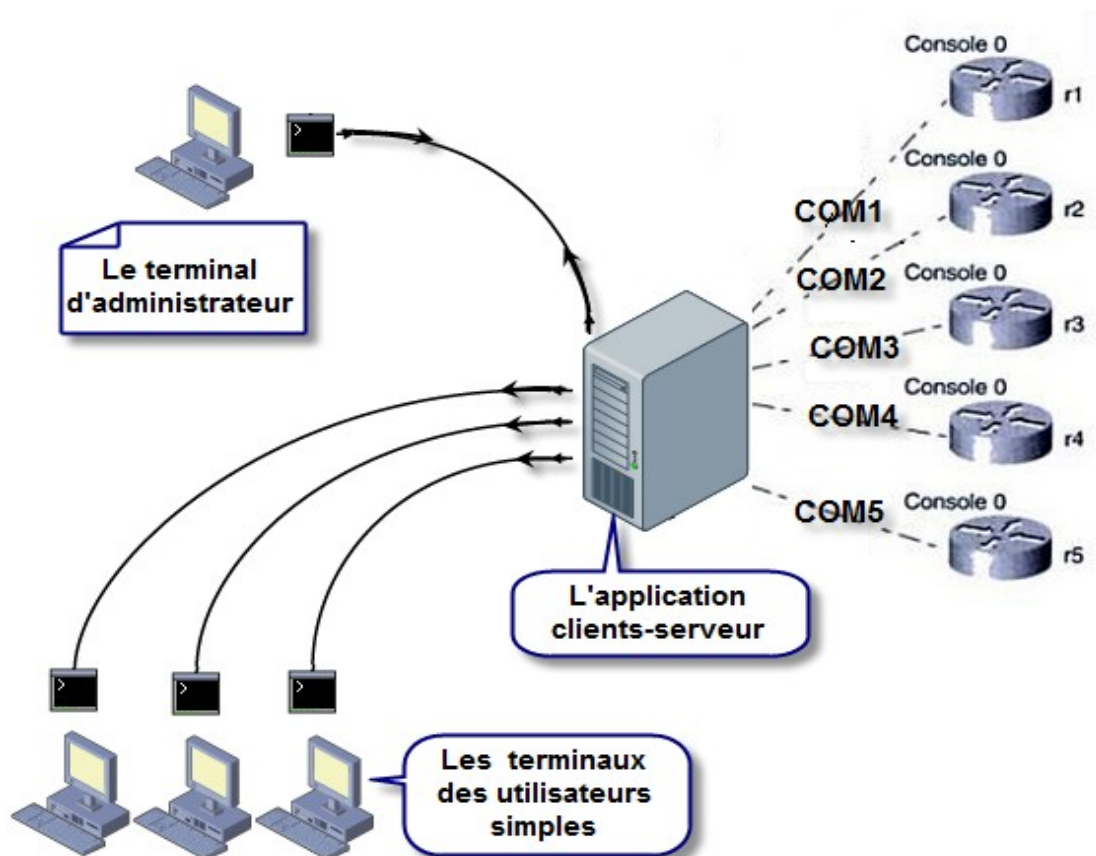


Figure 11 - La topologie réseau alternative aux serveurs terminal.

## 4.2 Notion d'application clients-serveur

La plus part des applications fonctionnant dans un environnement réseau sont classifiés comme applications clients-serveur. Ces applications telles que FTP, Telnet, et e-mail sont composées de deux éléments principaux : les clients et les serveurs. Les clients sont les utilisateurs requérant des services à partir de leur appareil connecté au réseau, tandis que les serveurs sont des ordinateurs distants disponibles en permanence et fournissant des services en réponse aux requêtes des clients.

Une application clients-serveur fonctionne constamment en répétant en boucle la routine suivante : une requête du client, une réponse du serveur et ainsi de suite. Par exemple, un navigateur web accède à une page internet en demandant un URL (Uniform Resource Locator), ensuite le serveur identifié par cet URL répond à la requête [3].

## 5 Python et les outils de développement

### 5.1 Le langage Python

Python est un langage de programmation comportant des caractéristiques attractives qui lui permettent de se démarquer des autres langages :

- Python a été inventé pour la productivité, étant donné qu'il est très facile à apprendre et sa syntaxe est très claire.
- Python est très ouvert sur l'extérieur et avec les autres langages et technologies, de sorte qu'on peut intégrer un code Python avec C, C++, et Java.
- La notion des type de données n'est pas explicite en Python, on est donc pas obligé d'associer à chaque donnée un type.
- Un code écrit en Python n'a pas besoin d'être compilé, les opérations d'analyse et de traductions du code sont réalisées à chaque ligne par un interpréteur.
- Python contient un ensemble de modules et de scripts utiles pour les travaux scientifiques.
- La programmation orientée objet est supportée dans le langage Python.

Le choix de programmer cette application en Python n'était pas seulement en vue de ces avantages mentionnés, mais aussi le fait que la connexion aux routeurs nécessite une communication série avec l'application, ce que j'ai déjà eu l'occasion d'effectuer avec les cartes électroniques Raspberry et Arduino. À plus forte raison encore, la disponibilité d'une très bonne librairie « Serial » au niveau du Python proposant de nombreuses méthodes pour gérer cette connexion.

### 5.2 La modélisation UML

L'UML (Unified Modeling Language ) comme son nom l'indique, est un langage de modélisation qui sert à décrire graphiquement des modèles de programmation informatique. Avec l'UML nous pouvons modéliser toutes les étapes du développement d'une application, on peut schématiser les

classes, modéliser leur méthodes et attributs, ainsi que les liens entre les classes. Donc il est possible d'un simple coup d'œil de représenter le fonctionnement de notre programme.

La Figure 12 correspond au diagramme de classes de l'application, où l'on trouve que chaque entité représente une classe particulière. Cependant, les méthodes et les attributs sont représentés au sein des entités.

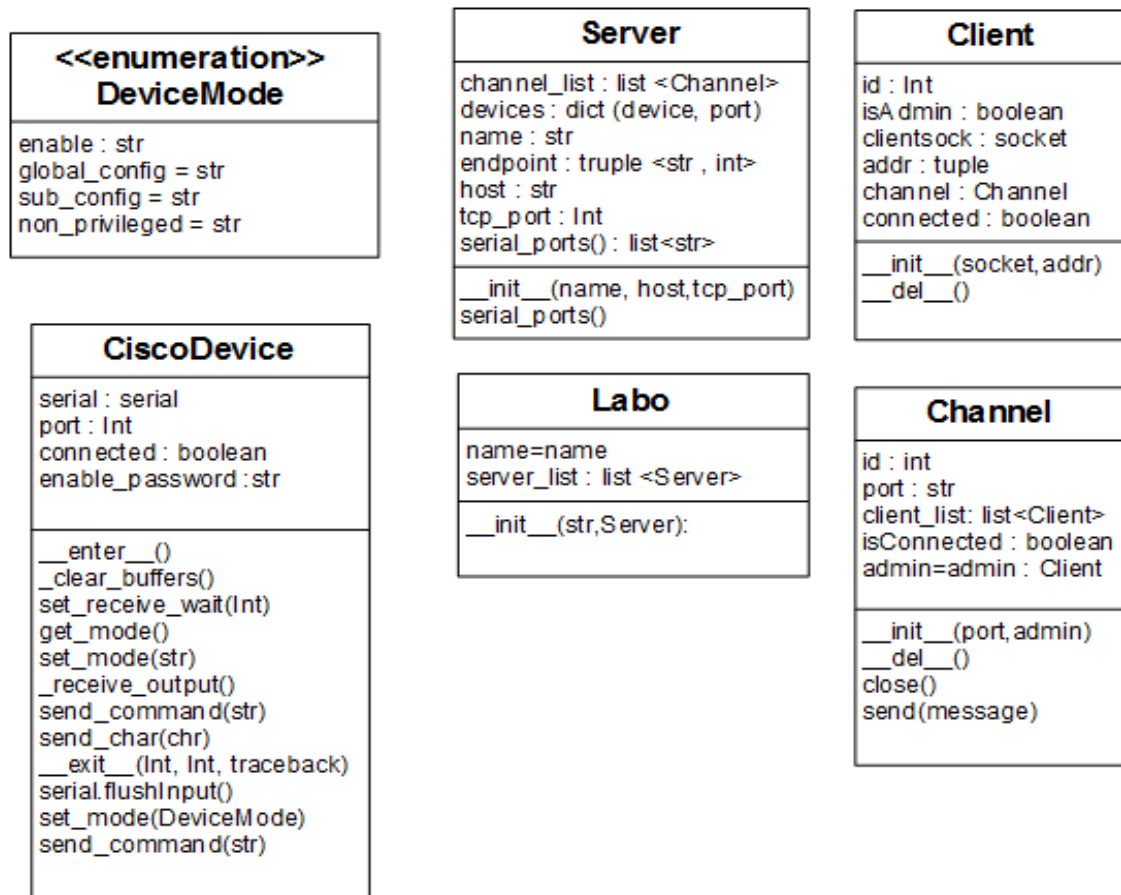


Figure 12 - Le diagramme de classes de l'application développée.

### 5.3 Le choix de l'environnement de développement intégré « Pycharm »

Pour programmer en Python, nous pouvons écrire le code avec un simple éditeur de texte puis l'exécuter avec un interpréteur Python sans aucun problème. Cependant, l'utilisation d'un environnement de développement intégré (IDE) peut présenter quelques avantages par rapport aux éditeurs de texte ordinaires de manière à faciliter la réalisation du projet.

À titre d'exemple, Pycharm est un des IDE libres (open source) qui offre un éditeur de texte intelligent avec coloration syntaxique. Ce programme est très utile pour les programmeurs en Python, JavaScript, CSS et beaucoup d'autres. Il fournit également une détection des erreurs automatiques, une mise à jour continue des libraires Python, ainsi qu'un téléchargement intégré et facile des dernières versions des modules. La Figure 13 est une fenêtre de Pycharm montrant la structure du code, des classes et des projets dans ce programme.

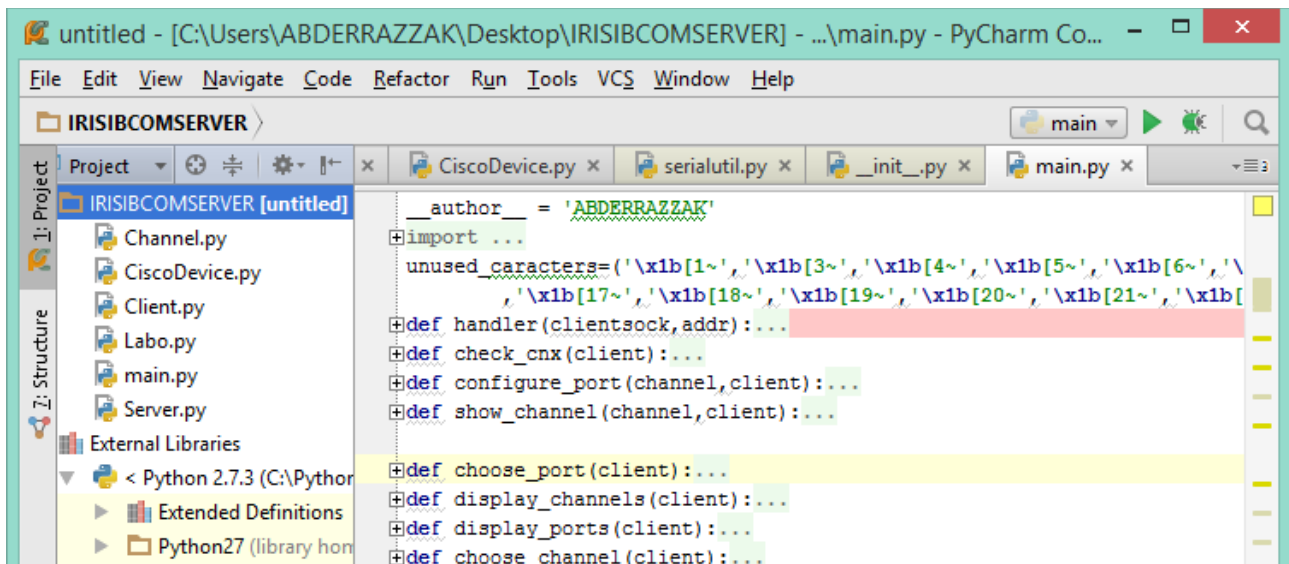


Figure 13 - L'environnement de développement intégré « Pycharm ».

## 5.4 Une brève description de l'application

La mise en place d'une connexion sécurisée avec un serveur Telnet ou SSH était vraiment une tâche difficile, surtout que le fonctionnement de ces deux protocoles reposent sur le concept des terminaux NVT (Network Virtual Terminal) qui utilisent des séquences de contrôle et des négociations des options entre le client et le serveur.

Afin de réaliser un programme fonctionnel, nous avons choisi d'implémenter un serveur avec un socket<sup>2</sup> TCP brut (Raw TCP socket), celui-ci permet d'envoyer et de recevoir des paquets sur un canal de communication sans spécifier le protocole de transport utilisé (UDP ou TCP). Cela signifie que les données transmises par un socket brut ne sont pas encapsulées par la couche transport du modèle OSI (La Figure 14). Donc cette connexion nous permet de s'affranchir de la contrainte des séquences de contrôle utilisées par le Telnet et le SSH lors de la programmation de l'application [5].

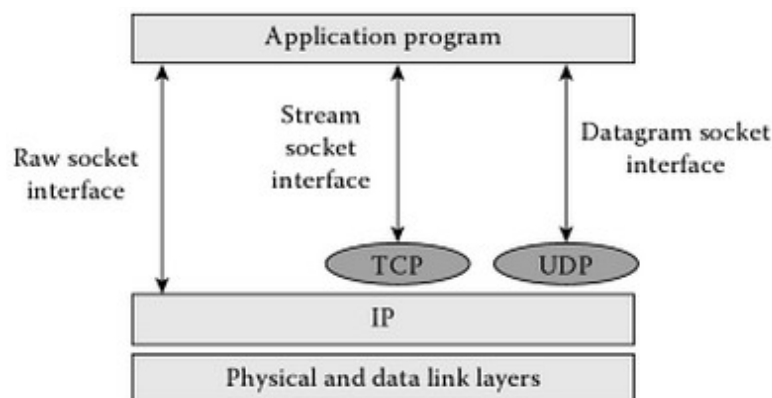


Figure 14- L'encapsulation des données avec les différents types de sockets [5].

<sup>2</sup> Un socket est un interface de connexion permettant d'établir un canal communication logique entre deux machines.



Bien que l'on ait utilisé une connexion non sécurisée avec des sockets TCP bruts, l'application réalisée pendant la courte période du stage, a pu répondre aux exigences de performance attendue. Effectivement, nous avons réussi à gérer la connexion de plusieurs clients, en utilisant ce que l'on appelle le « Multithreading », celui-ci permet d'exécuter plusieurs « threads<sup>3</sup> » simultanément de telle manière que chaque socket est assigné à un client. Donc le traitement de chaque client est lancé et géré dans un processus à part.

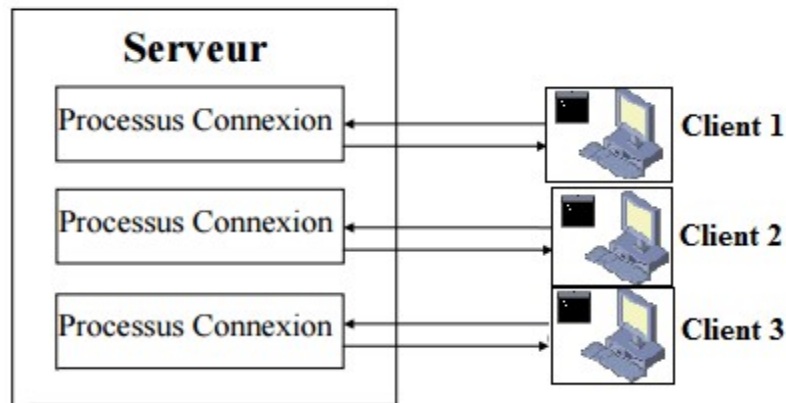


Figure 15 - Le Multithreading des clients TCP au sein du serveur.

Un des avantages remarquables du Python réside dans le fait qu'il est multiplate-forme, c.-à-d il fonctionne sur plusieurs plates-formes informatiques tel que Windows, Linux, Android, et aussi iOS Mac. Compte tenu de cela, nous avons écrit un code permettant de lancer l'application sous Windows et les différentes distributions de Linux. La Figure 16 montre comment l'application identifie le système d'exploitation, ensuite elle détecte les ports série selon le système.

```
if sys.platform.startswith('win'):
    ports = ['COM' + str(i + 1) for i in range(256)]
elif sys.platform.startswith('linux') \
    or sys.platform.startswith('cygwin'):
    ports = glob.glob('/dev/tty[A-Za-z]*')
elif sys.platform.startswith('darwin'):
    ports = glob.glob('/dev/tty.*')
else:
    raise EnvironmentError('Unsupported platform')
result = []
for port in ports:
    try:
        s = serial.Serial(port)
        s.close()
        result.append(port)
    except (OSError, serial.SerialException):
        pass
return result
```

Figure 16 - Détection des ports série par l'application sous Windows et Linux.

## 5.5 les problèmes rencontrés

Parmi les difficultés rencontrées, il convient de mentionner celle de gestion des caractères envoyés par les utilisateurs à distance au serveur. En effet, nous avons pu remarquer que les routeurs Cisco ne comprennent pas certains caractères ASCII, et par conséquent, cela conduit à la rupture de la connexion quand un texte contient un de ces caractères. Pour éviter ce problème, nous avons testé toutes les touches du clavier posant ce problème, ensuite, nous les avons converti au format adéquat.

<sup>3</sup> Un thread ou un fil d'instruction, est un processus qui s'exécute en parallèle de l'application et qui peut exécuter au cours de sa vie différentes procédures.

Un autre problème important est celui de la persistance de la connexion même après la fermeture du terminal par l'utilisateur. Cette étape était très importante car les équipements restaient occupés et non accessibles par les prochains utilisateurs. Ce problème a été résolu en effectuant un test sur la mémoire tampon (Buffer) qui reçoit les données du terminal. Comme montré sur la Figure 17, lorsque les données reçues sont « None »<sup>4</sup>, c'est à dire l'utilisateur ne renvoie plus de données pour une raison ou une autre, on ferme la connexion au routeur puis on sort de la boucle « while ».

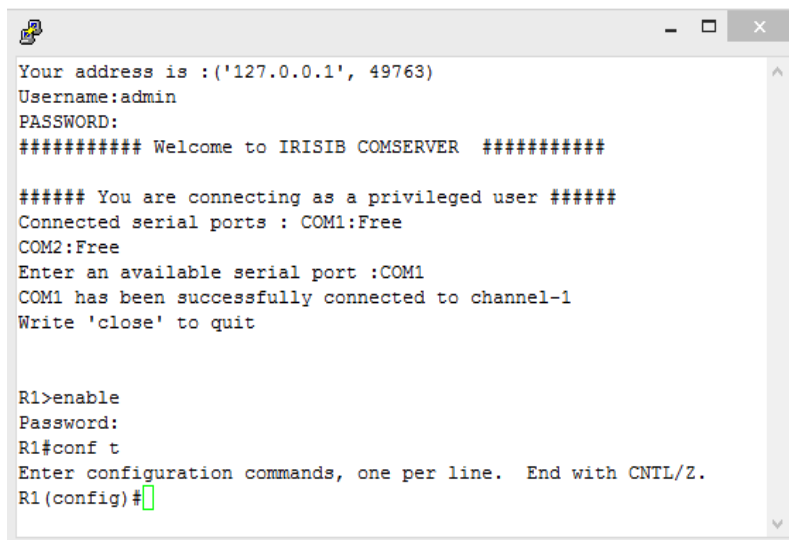
```
while 1:
    drdata=(device._receive_output())
    print(drdata.decode())
    channel.send(drdata)
    prdata = client.clientsock.recv(BUFF)
    if prdata == '\xc3\xa9': prdata='i'
    elif prdata == '\xc2\xa7': prdata=' '
    elif prdata == '\xc3\xa8': prdata='h'
    elif prdata == '\xc3\xa7': prdata='g'
    elif prdata == '\xc3\xa0': prdata='`'
    elif prdata == '\xc2\xb2': prdata='2'
    if not prdata:
        device.connected=False
        return -1
```

Figure 17 - Un morceau du code de l'application montrant comment le programme ferme la connexion au routeur.

## 5.6 Le teste de l'application

En fait, le teste d'une application n'est pas juste une action à la fin de sa réalisation, mais plutôt une routine pour un programmeur tout au long du développement du projet. En effet, la gestion des erreurs et les testes effectués à chaque modification du code sont des tâches lourdes et permanentes, car elles prennent la plus grande part du temps et des efforts consacrés au développement d'une application informatique. Dans cette partie, nous allons exposer seulement les résultats finaux de l'application « TCP-COM serveur », en vous expliquant essentiellement comment cette application redirige les utilisateurs vers les routeurs.

Dans un premier temps, l'administrateur accède au serveur avec son compte privilégié. Celui-ci lui envoie par la suite les ports série y connectés et disponibles en mode configuration (la Figure 18). Une fois cela fait, l'administrateur choisit un port pour créer un canal de communication de sorte qu'il puisse configurer le routeur associé. Comme illustré sur la Figure 19, les utilisateurs simples peuvent s'associer à ce canal de telle manière qu'ils reçoivent seulement les lignes de commande rentrées par l'administrateur,



```
Your address is : ('127.0.0.1', 49763)
Username:admin
PASSWORD:
##### Welcome to IRISIB COMSERVER #####

##### You are connecting as a privileged user #####
Connected serial ports : COM1:Free
COM2:Free
Enter an available serial port :COM1
COM1 has been successfully connected to channel-1
Write 'close' to quit

R1>enable
Password:
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#
```

Figure 18 - Création d'un canal de communication par un administrateur.

<sup>4</sup> None en Python signifié « Vide » ou « l'objet ne possède pas de valeur ».

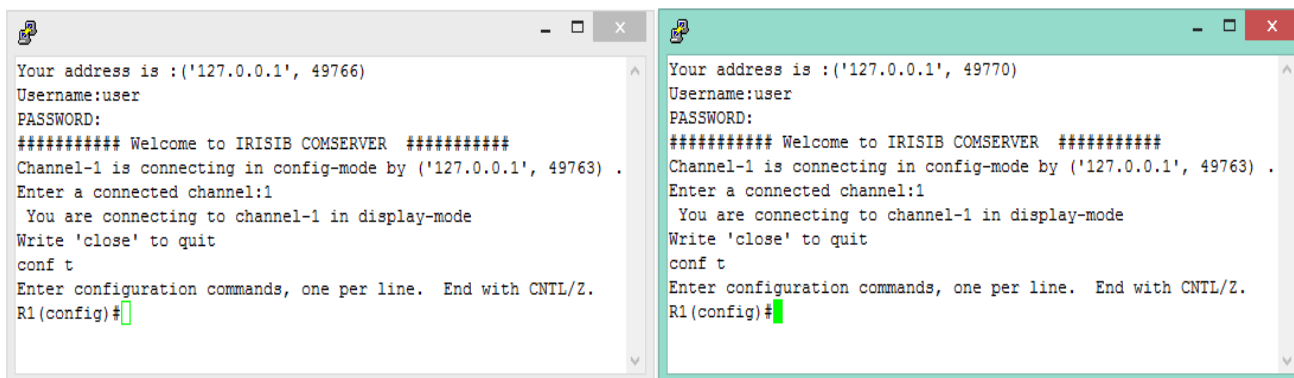


Figure 19 - Les utilisateurs simples s'associent à un canal puis ils reçoivent la configuration sur leur terminal.

## 6 La simulation de la topologie réseau alternative

### 6.1 Gns3 et Virtual serial port emulator

Afin de vérifier le bon déroulement du projet, chaque étape lors de la programmation de l'application nécessite un essai réel sur les équipements. Bien évidemment, cela sera important aussi bien pour tester les erreurs et les exceptions de programmation que pour comprendre le fonctionnement des équipements. Cependant, l'absence d'un des éléments nécessaires pour faire le montage réel entrave l'avancement du projet.

Heureusement, il existe des programmes professionnels permettant de simuler virtuellement le comportement des systèmes réels. Parmi ceux-ci, on trouve Gns3 (Graphical Network simulator), qui est un programme libre permettant de faire la simulation des topologies réseaux informatiques. Gns3 peut établir des liaisons réelles entre les routeurs simulés et la machine en utilisant la carte réseau réelle, ou aussi par des cartes de bouclage Microsoft (LoopBack Network Adapters). De plus, Gns3 intègre des images IOS réelles des routeurs Cisco. Par conséquent, la mise en œuvre d'un tel simulateur permet de s'affranchir des contraintes liées à rester au laboratoire durant tout le temps des essais.

Malheureusement, « Gns3 » ne permet pas de simuler les ports console des routeurs Cisco. Donc nous ne pouvons pas faire une connexion console sur les routeurs. Un autre problème est lié au nombre limité des ports COM (RS-232) disponibles sur une machine. Pour remédier à ces problèmes nous utilisons un autre programme très réussi dans la famille des programmes de simulation appelé « Virtual serial port emulator ». Celui-ci permet d'ajouter autant des ports COM virtuels que vous le désirez, mieux encore, ce programme permet d'établir un pont de flux des données (Stream Bridge ) entre les connexions TCP et les ports COM virtuels. La Figure 20 illustre plus clairement la simulation effectuée.



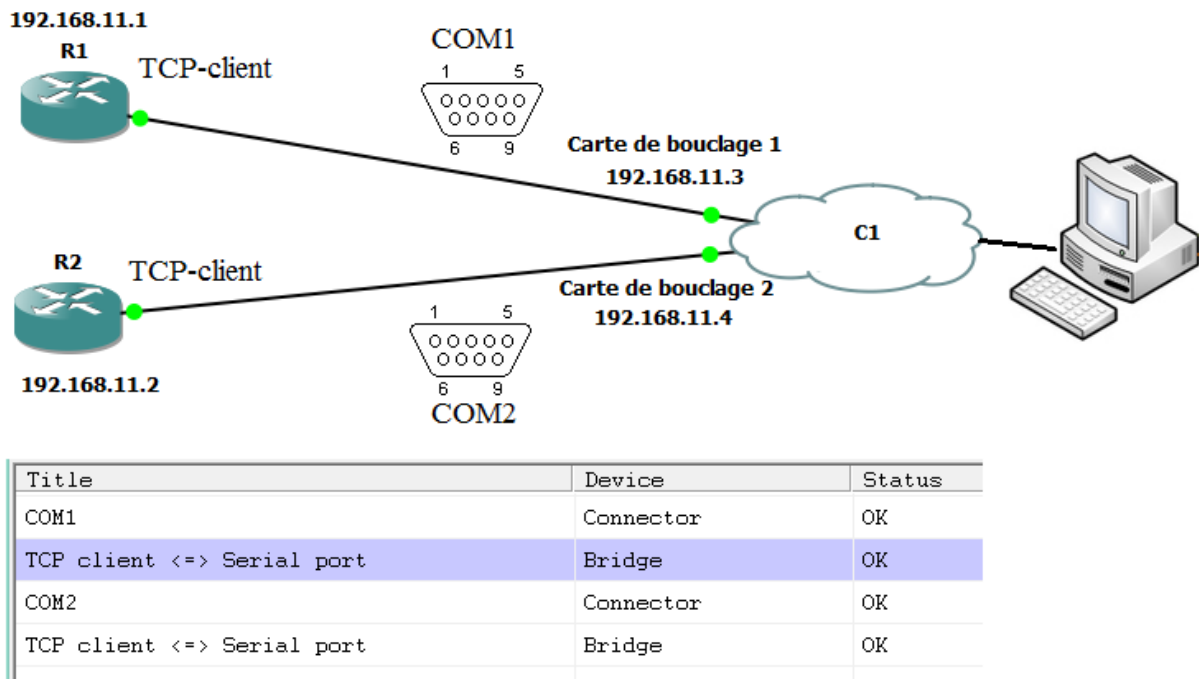


Figure 20 - La simulation de la topologie réseau sur Gns3 et Virtual serial port emulator.

## 7 Le matériel utilisé et la mise en place réelle

### 7.1 Le matériel utilisé

Malgré que les applications de simulation nous a simplifié grandement la vie durant le stage, le fonctionnement du système avec le matériel réel pourrait être différent de ce qui est en simulation. Donc, il était nécessaire d'effectuer quelques testes sur les équipements réels avant d'installer définitivement le système dans le laboratoire.

Comme nous l'avons déjà mentionné, parmi les fonctionnalités disponibles en simulation et qui ne sont pas possibles en pratique, le nombre limité des ports séries (COM) et les sorties USB sur une machine. Heureusement pour nous, il existe des concentrateurs USB et des convertisseurs USB-COM qui permettent de multiplier ses ports et ainsi pouvoir installer un grand nombre des équipements nécessaires.

Une autre remarque très importante porte sur l'utilisation de la carte électronique « Raspberry Pi » au lieu d'un ordinateur, cette mini-carte électronique est en soi un mini-ordinateur supportant plusieurs langages de programmation. Le plus souvent on trouve le langage Python installé par défaut sur la distribution linux « RASPBIAN ». Le Raspberry comporte également d'autres caractéristiques remarquables, parmi lesquelles figure :

- La faible consommation d'électricité.
- L'absence de pièces en mouvement, c-à-d il ne contient pas de disque dure, il suffit de brancher une carte mini-SD qui contient le système d'exploitation.

- Sa taille est très petite par rapport à un ordinateur.
- Elle possède une énorme documentation ainsi qu'un grand nombre des projets et des forums de support techniques.

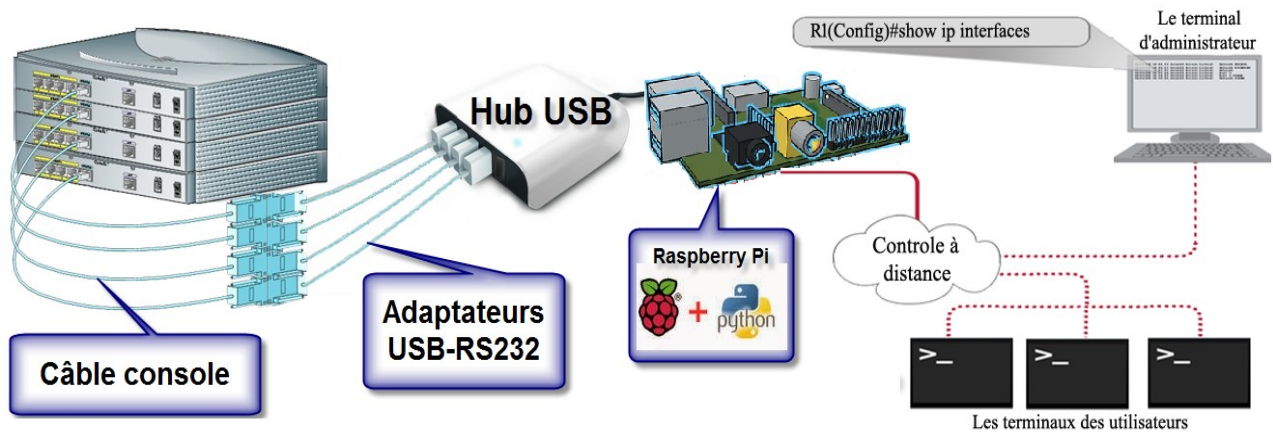


Figure 21 - Une vue schématique représentant simplement le montage effectué.

Le Tableau 2 donne la listes du matériel utilisé lors des essais pratiques :

Câbles console (RJ45-RS232)	
Concentrateurs USB 2.0 7 Ports D-Link	
Routeurs Cisco série 2500	



<p>Convertisseurs USB-RS232</p>	
<p>Raspberry Pi B+</p>	

Tableau 2- La liste du matériel utilisé.

## 8 Conclusion générale

À la fin d'une mission réussie, nous avons bien répondu à la problématique de départ consistant à remplacer les serveurs terminal par un nouveau système reposant sur une application logicielle installée sur un ordinateur. En effet, l'application développée a été installée avec succès et elle est prête à être utilisée au sein du laboratoire.

Sur base des tâches effectuées lors de ce stage, j'ai eu l'occasion de développer mes compétences en programmation orientée objet, et en particulier, l'utilisation d'un langage très intéressant tel que Python, nous a ouvert l'esprit sur d'autres projets intéressants .

Par ailleurs, la manipulation des équipements Cisco dans un laboratoire réseau comme celui de l'IRISIB, qui dispose d'une grande variété de matériel réseau et un environnement agréable, m'a permis de développer mes connaissances dans ce domaine. Surtout que je suis très intéressé par les examens de Cisco CCNA et CCNP.

Je pense que cette expérience m'a donné une bonne préparation pour l'insertion professionnelle, car elle était très enrichissante et complète. De plus, les activités effectuées m'ont beaucoup motivé et conforté dans mon choix de m'orienter vers mon futur métier dans le domaine de l'informatique.

## Bibliographie

- [1] DONNA, H-L. CCNP Practical Studies: Troubleshooting. *Indianapolis, Cisco Press, 2003.*
- [2] LOHIER, S. & PRESENT, D. Transmissions et réseaux Cours et exercices corrigés, 5ème édition. *Paris, DUNOD, 2010.*
- [3] LINDHOLM, C et al. Cisco Networking Academy Program: First-Year Companion Guide, 2ème édition. *Indianapolis, Cisco Press, 2001.*
- [4] ODEM , W. CCENT/CCNA ICND1 640-822 Official Cert Guide, 3ème édition. *Indianapolis, Cisco Press, 2011.*
- [5] SATHYAN, J. Fundamentals of EMS, NMS and OSS/BSS. *Boca Raton, Auerback, 2010.*
- [6] ZIARKO, W et al. Rapport Belge en matière de science, technologie et innovation 2010. *Belgique, Belspo, 2010.*
- [7] <http://www.cisco.com> consulté le 18 Mai 2015.
- [8] <http://www.irisib.be> consulté le 18 Mai 2015.
- [9] <http://www.recherche-technologie.wallonie.be/>, Les programmes FIRST. *consulté le 18 Mai 2015.*