B. Pitticco
J. Oziol
S. Detroz

UE RSX218 2022/2023
S. Secci

Projet 12
NetSquid

Annexe 6 – Mise en place du contexte technologique

Notre projet n'a pas nécessité beaucoup de prérequis technique. En effet, ce sont nos postes personnels qui exécutent les scripts développés. L'installation de NetSquid étant mieux intégrée sur des distributions Linux, nous avons fait le choix d'installer WSL sur nos PC. Le "Windows Subsystem for Linux" est un sous-système de Microsoft qui permet d'exécuter des applications Linux directement sur Windows 10 ou 11, sans créer de machine virtuelle. Ainsi, nous pouvons télécharger la distribution Ubuntu via le Microsoft Store et exécuter toutes les commandes disponibles sur ce système d'exploitation directement sur nos Windows. Nous exécuterons donc nos scripts Python sur ces sous-systèmes Ubuntu.

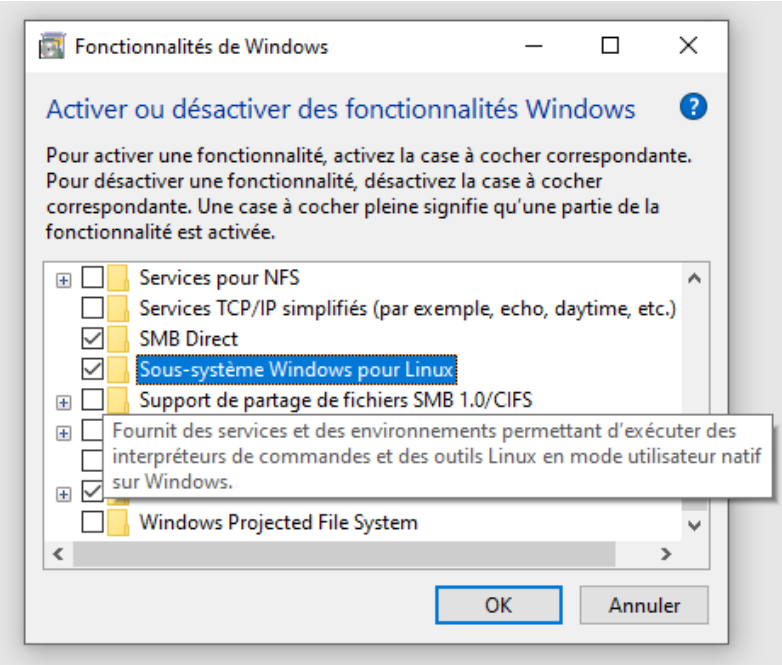

Figure 1 - Activation de WSL

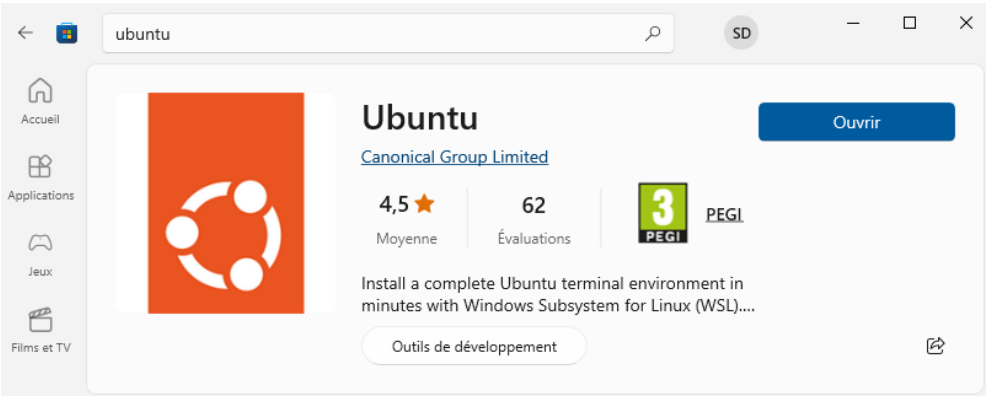

Figure 2 - Téléchargement d'Ubuntu

Notre solution est donc développée en Python. Ce dernier est un langage de programmation orienté objet, conçu pour être facile à lire et écrire. Etant une solution gratuite et open source, les développeurs du monde entier peuvent utiliser et modifier son code source pour leurs projets. Ils peuvent ainsi mettre à disposition des autres utilisateurs une grande variété de bibliothèques de leur création.

C'est ainsi que NetSquid fut créé et rendu disponible au grand public. C'est un outil de simulation de réseau de communication quantique complexes. Il a été développé par l'équipe de recherche QuTech de l'Université de technologie de Delft aux Pays-Bas. En plus de ses propres outils, NetSquid se base également sur d'autres bibliothèques Python, telles que :

- QuTiP, qui permet d'effectuer des calculs de simulation sur des qubits ;
- NumPy, pour réaliser des calculs complexes rapidement, notamment sur des opérations matricielles, qui sont nécessaires à la simulation quantique ;
- PyDynAA, très important pour le protocole E91, qui est un simulateur à évènements discrets.

La réunion de tout ce code, à travers NetSquid, permet donc de modéliser et de simuler de nombreux systèmes quantiques, tels que des canaux de communication, des ordinateurs, des capteurs et d'autres systèmes de traitement de l'information quantique. Ce simulateur offre également la possibilité de générer du bruit et des erreurs, ce qui nous permettra de tester notre réseau quantique dans des conditions dégradées.

Pour l'installer, il suffit de créer un compte sur le site officiel de NetSquid et de lancer cette simple commande :

pip3 install --extra-index-url https://pypi.netsquid.org netsquid

Cette dernière vous demandera de renseigner votre login et votre mot de passe, et la librairie NetSquid sera présente sur votre machine.

Notre projet se divise donc en deux scripts différents. Le but de chacun d'entre eux et de faire en sorte que Alice envoie un message chiffré à travers un canal de communication classique et que, de l'autre côté, Bob puisse le réceptionner et le déchiffrer. Pour générer une clé de chiffrement, nous avons utilisé les protocoles cryptographiques quantiques BB84 et E91. Pour chaque script, nous avons également eu la possibilité d'ajouter des variations d'aléas, afin de stresser nos simulations. Les résultats des mesures que nous avons obtenues sont disponibles dans le partie « Compte rendu » du rapport final. Enfin, pour chacun des protocoles, nous avons mis en place un troisième hôte. Pour BB84, il s'agit de Eve, qui tente d'exécuter une attaque de type Man in the Middle. Elle souhaite ainsi récupérer le message envoyé par Alice à la place de Bob. Pour E91, l'hôte Charlie intervient pour faire office de tier de confiance entre les deux autres hôtes, et leurs distribuer des qubits intriqués, qu'Alice et Bob pourront utiliser pour générer leurs clés secrètes.

B. Pitticco
J. Oziol
S. Detroz

UE RSX218 2022/2023
S. Secci

Projet 12
NetSquid

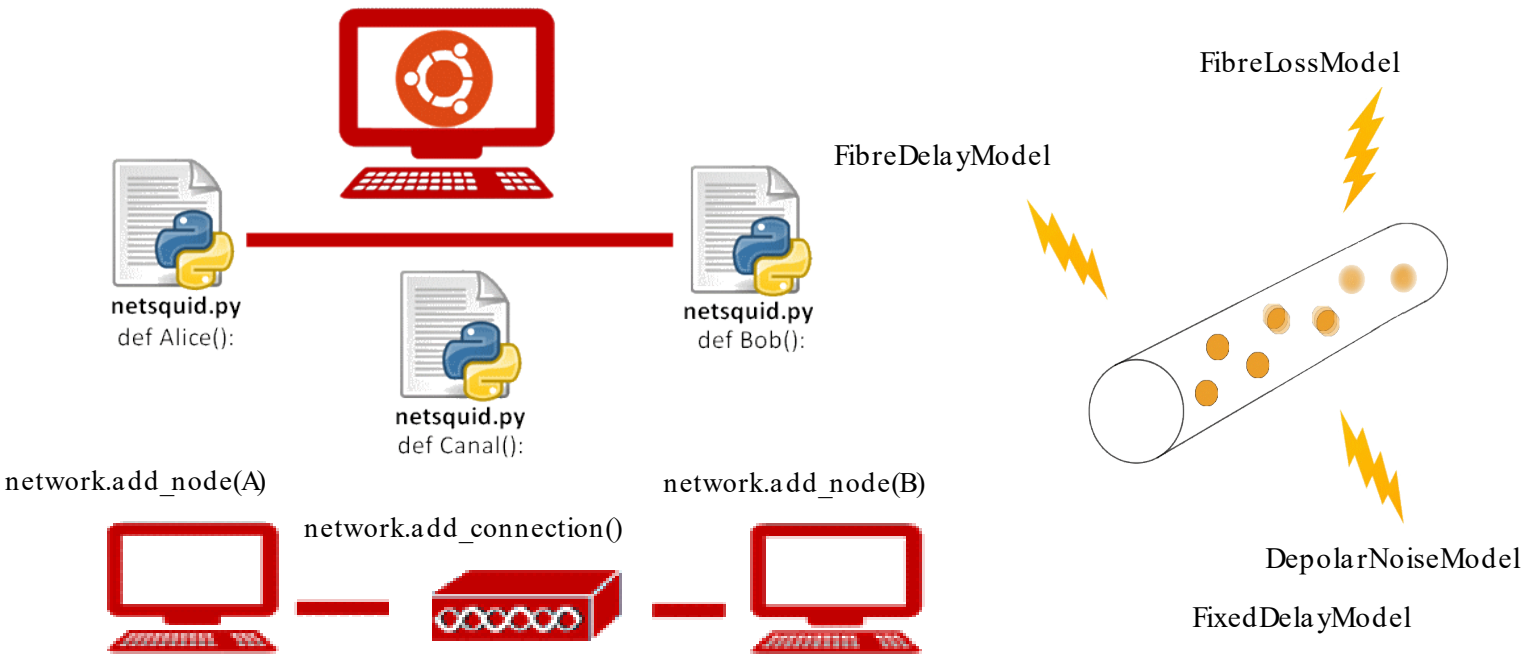

Figure 3 - Illustration de nos maquettes