**Rapport de Projet : Implémentation d'une DHT avec SimPy**

**Introduction**

L'objectif de ce projet est de concevoir et implémenter une table de hachage distribuée (DHT) en utilisant SimPy. Une DHT permet un stockage et une récupération efficace des données dans un réseau pair-à-pair. Le projet a été divisé en plusieurs étapes :

1. Gestion de l'ajout et du retrait des nœuds (join/leave)
2. Routage des messages (Send/Deliver)
3. Stockage et récupération des données (Put/Get)

**1. Gestion des nœuds : Join/Leave**

Dans cette première étape, chaque nœud a un identifiant unique et est organisé dans un anneau basé sur cet identifiant. Chaque nœud ne connaît que ses voisins immédiats.

**Ajout d'un nœud**

Lorsqu'un nouveau nœud rejoint la DHT, il contacte un nœud existant. Il trouve sa position dans l'anneau en itérant sur les nœuds voisins jusqu'à ce qu'il détermine son emplacement optimal. Une fois inséré, il informe ses nouveaux voisins.

**Suppression d'un nœud**

Lorsqu'un nœud quitte la DHT, il informe ses voisins gauche et droit, qui se reconnectent directement entre eux pour maintenir la structure de l'anneau.

**2. Routage des Messages : Send/Deliver**

Une fois que l'anneau est fonctionnel, les nœuds peuvent s'envoyer des messages. Chaque message contient un expéditeur, un destinataire et un contenu.

**Fonctionnement**

Lorsqu'un nœud souhaite envoyer un message à un autre, il vérifie si le destinataire est l'un de ses voisins. Si ce n'est pas le cas, il transmet le message à son voisin le plus proche du destinataire, optimisant ainsi le routage.

**3. Stockage et Récupération des Données : Put/Get**

La dernière étape consiste à permettre aux nœuds de stocker des données.

**Attribution des données**

Chaque donnée a un identifiant unique, et le nœud ayant l'identifiant le plus proche devient responsable de cette donnée. Pour garantir la redondance, chaque donnée est répliquée sur les deux voisins du nœud responsable.

**Récupération des données**

Lorsqu'un nœud cherche une donnée, il envoie une requête qui est routée à travers la DHT jusqu'au nœud responsable. Celui-ci renvoie alors la donnée.

**Conclusion**

Le projet a permis de mettre en place une DHT fonctionnelle avec les mécanismes essentiels : gestion des nœuds, routage des messages et stockage des données. Des améliorations peuvent être envisagées, comme une gestion plus robuste des départs de nœuds ou l'introduction d'une tolérance aux pannes.

**Perspectives et Améliorations**

* **Optimisation du routage** : Implémenter un routage plus efficace en utilisant des structures de sauts longs (ex: Chord).
* **Tolérance aux pannes** : Ajouter des mécanismes pour détecter et remplacer dynamiquement les nœuds défaillants.