

## 1 Description générale

Le projet PPII a pour objectif de vous faire appliquer au sein d'un même projet les différentes connaissances acquises en STIC durant le S6 de la formation, à savoir en langage C, en bases de données et en réseau. Le projet consiste cette année à implanter les fonctionnalités principales d'un système de fichier décentralisé et basé sur des communications de type pair à pair (P2P) appelé Telecom Nancy File System qui reprend les grands principes d'Inter-Planetary File System<sup>1</sup>.

## 2 Partie 1 : Étude d'IPFS

La première partie du projet PPII (janvier-février) consiste en une étude bibliographique, d'une part des fonctionnalités proposées par IPFS, et d'autre part des concepts et mécanismes sous-jacents, ce qui inclut notamment les aspects hashing / content addressing, Content Identifiers (CIDs), Merkle DAGs, Mutable File System (MFS), Kademlia, les aspects protocolaires, etc. Un premier rendu est attendu le 07/03 pour rendre compte sous la forme d'un rapport de votre étude de IPFS.

IPFS repose notamment sur une table de Hachage Distribuée (DHT). Vous devez comprendre le fonctionnement de Kademlia<sup>2 3</sup> qui décrit comment sont réalisés l'indexation et le routage au sein du réseau en garantissant que l'ensemble du système passe à l'échelle (de fait, des implantations de Kademlia supportent des millions de pairs simultanés). Kademlia utilise la métrique XOR pour calculer les distances dans l'espace d'adressage virtuel de la DHT et distribue ensuite les contacts de manière à avoir K-contacts pour représenter des zones dont la taille double à chaque niveau. Ceci est illustré par les figures 1 et 2 (la première présente les adresses réelles, la seconde les distances XOR).

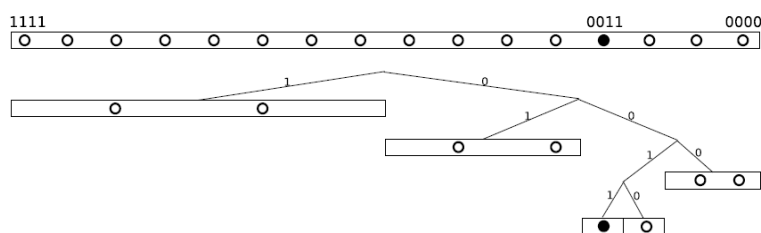


FIGURE 1 – Illustration de K-buckets de Kademlia

## 3 Partie 2 : Réalisation

Le langage de développement est le C. Aucune interface graphique n'est demandée. Dans un premier temps, il vous faut développer les fonctionnalités du système de fichier, puis les aspects réseaux dans un second temps

1. <https://ipfs.io/>
2. <https://en.wikipedia.org/wiki/Kademlia>
3. <https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf>

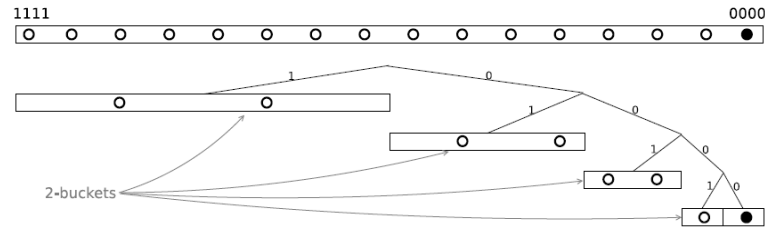


FIGURE 2 – Illustration de K-buckets de Kademlia selon leur distance à la source

car la programmation réseau en C ne vous sera enseignée qu’au mois d’avril. Vous avez le droit d’utiliser des bibliothèques cryptographiques tierces implantant des fonctions de hachage ou des bases de données (par exemple Redis dont la bibliothèque C officielle est hiredis <sup>4</sup>).

### Principales fonctionnalités attendues

- Étant donné un répertoire, créer les identifiants et structure de données nécessaires à TNFS ;
- Implanter la DHT (recherche/publications de pairs/contenus) ;
- Permettre le téléchargement d’un fichier étant donné son identifiant (mono-source) ;
- Bonus : répliquer automatiquement un fichier (fonction Pin).

## 4 Rendu

Le projet se fait en trinôme (groupes libres) et sera évalué sur la base d’un rapport (à rendre pour le 23/05), d’une soutenance (le 01/06) et du code produit et versionné sur le Gitlab de l’école. Vous devez m’ajouter au dépôt git de votre groupe en tant que "Développeur" pour que je puisse accéder aux sources.

Vous réaliserez un Makefile permettant la génération de l’exécutable et un fichier README donnant les indications pour l’exécution.

Le rapport devra notamment contenir :

- une étude bibliographique du fonctionnement de IPFS (rapport intermédiaire du 07/03) ;
- la description de votre implantation des structures de données utilisées pour système de fichier ;
- la spécification du protocole applicatif (type et format des messages) entre pairs ;
- la description de l’algorithme général d’un pair ;
- le protocole de test effectué pour valider votre développement et en garantir la qualité ;
- la répartition du travail et les éléments de gestion de projet pertinents.

4. Disponible dans les dépôts de paquets de la plupart des distributions GNU/Linux : [Debian](#), [Ubuntu](#), [Fedora](#), [Archlinux](#), ...