AMAZOUNE Marc
BENCHRIFA Mohamed Amine
BOUZID Fares
DESTREE Gabriel
KANE Ababakar

Routage des messages entre les nœuds (Blockchain)

SOMMAIRE

DOCUMENTATION

A propos de Kademlia

Les activités de pair à pair de Blockchain sont bien facilitées par le nouveau réseau de recouvrement, Kademlia. Connue pour être la table de hachage distribuée de pair à pair la plus utilisée, Kademlia a d'abord été développée pour aider à trouver et résoudre les nœuds qui contiennent des données spécifiques, mais son protocole contient également une disposition d'interrogation qui permet de garantir une vue complète et claire du réseau avec de nombreux nœuds tout en étant capable de trouver des nœuds spécifiques. Le protocole Kademlia est conçu de telle manière qu'il favorise une communication efficace sur le réseau grâce à l'échange de clés publiques qui crypte la communication. Kademlia utilise des clés opaques et les affecte à chaque réseau participant qui est utilisé pour stocker des informations par paires. De cette façon, les utilisateurs peuvent localiser facilement et rapidement les serveurs disponibles à proximité des clés cibles. Afin d'éviter les problèmes de délai d'attente dus à des nœuds défaillants, Kademlia utilise une fonction de requête parallèle pour vérifier ceci. Avec Kademlia, il y a une réduction des messages de configuration envoyés à travers le fonctionnement des nœuds. Il crée également des chemins à faible latence qui permettent une flexibilité dans le routage des requêtes.

A propos du Routage en Oignon

Le routage en oignon est une technique pour la communication anonyme sur un réseau informatique. Dans un réseau en oignon, les messages sont encapsulés dans des couches de chiffrage, analogues aux couches d'un oignon. Les données cryptées sont transmises par une série de nœuds de réseaux appelés des routeurs d'oignon, dont chacun "enlève" une seule couche, découvrant la destination suivante des données. Quand la couche finale est décryptée, le message arrive à sa destination. L'expéditeur reste anonyme parce que chaque intermédiaire ne connait que l'emplacement immédiatement précédant et suivant des nœuds.

Le réseau de recouvrement Kademlia et la technique de communication grâce au routage en Oignon sont donc l'alliance parfaite pour une transmission de bout en bout anonyme et sécurisée.

Scénario Kademlia

Dans un premier temps, nous supposons que chaque nœud possède un ID, une clé publique chiffrée, une IP et un port. Chaque nœud va stocker une table de nœuds connue et actifs de K nœuds. Nous avons la possibilité de demander à un autre nœud de lui chercher un nœud spécifique, via une requête (ID nœud cherché + ID nœud source).

Le format du tableau sera le suivant : nous aurons K colonnes de manière à ce que la ligne i rassemble tous les nœuds ayant les i premiers bits semblables (pour simplifier l'identification).

1- Nous supposons que le nœud A souhaite contacter le nœud C
2- Le nœud A envoie une requête de type « lookup » au nœud B pour trouver C
3- Le nœud B renvoie au nœud A la liste de tous les nœuds se situant dans la ligne correspondante au nœud C. ;
4- Dans le cas où il ne le trouve pas dans la réponse fournit par le nœud B. → Il va envoyer des requêtes à tous les nœuds de la table de B et ainsi de suite jusqu'à trouver C.

Problématique *: Comment est-ce qu'un nouveau nœud peut identifier son premier nœud et l'ajouter à sa table, et ainsi joindre le réseau ?*

Pour ce faire, nous allons faire en sorte que chaque « peer » ait K « peers » qui fonctionneront tel un serveur DNS. Dès lors, lorsqu'il souhaitera rejoindre le réseau, le nœud pourra demander à ses serveurs « peers » de lui fournir une liste initiale des nœuds connus.

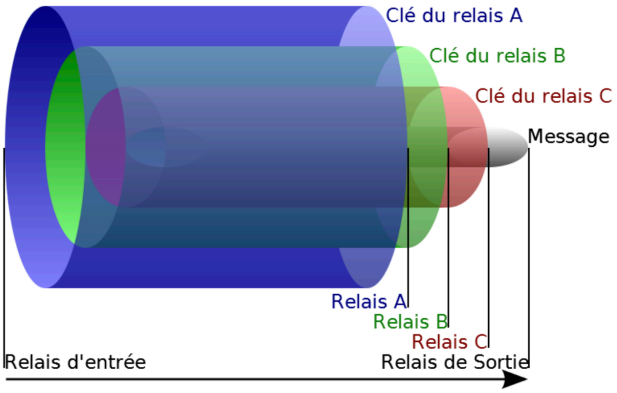
A chaque fois qu'un nœud souhaite rejoindre le réseau on lui ajoutera automatiquement 3 « noeuds » connues qui lui permettront de rejoindre le reste des nœuds existants. Pour des raisons de sécurité, on fera en sorte que ces trois nœuds là seront choisis d'une manière complétement aléatoire.

i = 0	10100	?????	?????	?????	?????
i = 1	00011	00101	00100	00010	00010
i = 2	01100	01101	01110	01111	?????
i = 3	01001	?????	?????	?????	????
i = 4	01011	?????	?????	?????	?????

L'ID du node est 01011, dans notre cas K = 5

Routage en Oignon

Après avoir effectué le routage et choisi le chemin par ou va transiter le paquet (les nœuds choisis par où l'on va passer), le nœud source va demander les clefs publiques de tous les autres nœuds avec le nombre de sauts. Par suite, il va crypter le message avec la clé publique par rapport à la longueur du saut de manière à ce que le premier chiffrement s'effectuera par la clé publique du nœud avec la plus grande longueur de saut. A chaque fois que le message arrivera au nœud suivant, il déchiffrera sa couche avec sa clé privée et ainsi de suite jusqu'à arriver à destination ou après avoir décrypter sa couche il trouvera le message en clair.



Sources

https://en.wikipedia.org/wiki/Onion_routing

https://fr.wikipedia.org/wiki/Kademlia

https://journalducoin.com/altcoins/service-de-livraison-anonyme-base-blockchain-routage-oignon-de-thor/

https://www.techbullion.com/tim-blockchain-gps-based-source-revealed-new-kademlia/

https://medium.com/orbs-network/the-actual-networking-behind-the-ethereum-network-how-it-works-6e147ca36b45

https://openclassrooms.com/fr/courses/2939276-surfez-incognito-sur-internet-avec-le-reseau-tor/2955001-tor-et-le-routage-en-oignon

https://medium.com/@preethikasireddy/how-does-ethereum-work-anyway-22d1df506369

http://www.ijsrp.org/research-paper-0715/ijsrp-p4399.pdf

https://karac.ch/blog/creer-une-blockchain-avec-javascript