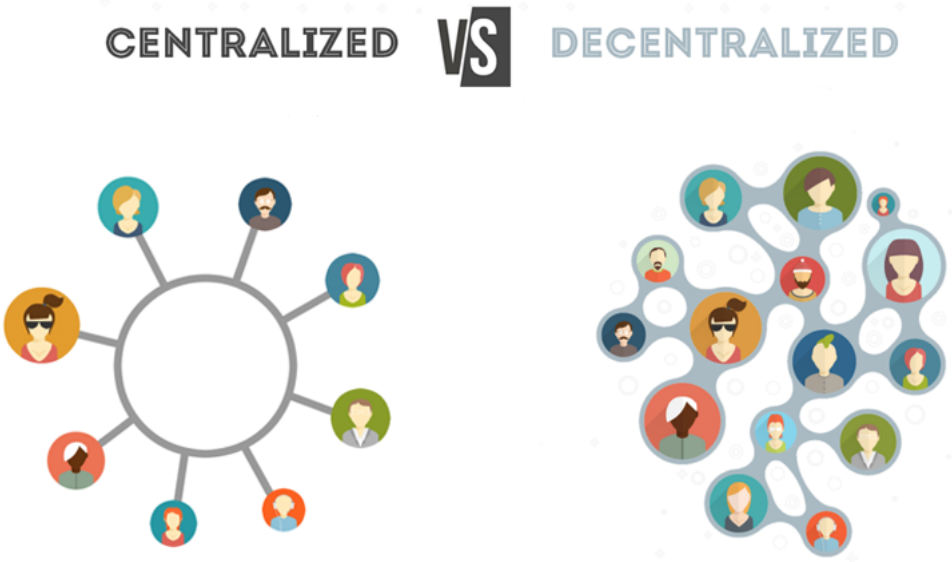
Title

Subtitle



Auteurs

Benjamin NIDDAM

Encadrants

Adrien LUXEY-BITRI

Table des matières

Chapitre 1

Introduction

1.1 Contexte

Dans le cadre du PJI (Projet Individuel) en fin de master 1, j'ai choisi de travailler sous la tutelle de monsieur LUXEY-BITRI sur le sujet des communications réseaux dites pair à pair. Avec peu de connaissances en réseaux, j'y ai vu l'occasion de découvrir un domaine qui m'était inconnu et de me confronter à un sujet de recherche pour l'instant peu exploré mais avec un potentiel d'application très important. En effet, si aujourd'hui les communications réseaux sont majoritairement centralisées, on voit émerger de plus en plus de réseaux dits décentralisés.

Les réseaux décentralisés, également connus sous le nom de réseaux pair-à-pair (P2P), offrent une approche alternative à la communication et à l'échange de données. Contrairement aux réseaux centralisés, où les données transitent par un point central tel qu'un serveur, les réseaux P2P permettent aux utilisateurs de communiquer directement entre eux, en utilisant leurs propres ressources (telles que la puissance de calcul et la bande passante) pour faciliter les échanges.

Ce modèle de communication présente plusieurs avantages, notamment une meilleure résilience face aux pannes, une répartition de la charge plus équilibrée et une réduction de la dépendance à l'égard d'entités centrales. Les réseaux P2P sont largement utilisés dans divers domaines, tels que le partage de fichiers (comme BitTorrent), la messagerie instantanée (comme Skype) et les systèmes de paiement (comme Bitcoin).

J'ai choisi ce sujet car la majorité de mon travail personnel consiste à développer des applications web basées sur des architectures centralisées. Je souhaitais donc découvrir un nouveau domaine et me confronter à de nouvelles problématiques. De plus, les réseaux pair à pair sont un sujet de recherche très actuel et qui a un fort potentiel d'application. En effet, les réseaux pair à pair sont de plus en plus utilisés dans le domaine des communications réseaux. On peut citer par exemple le protocole WebRTC qui permet la communication audio et vidéo entre deux clients sans passer par un serveur.

1.2 Problématique

Quelles sont les solutions possible pour mettre en place une communication textuelle temps réel entre deux clients dans un réseau pair à pair ?

1.3 Objectifs

L'objectif de ce projet est l'étude des mécanismes permettant le "pair à pair" dans le web moderne grace à l'implémentation de prototypes permettant la communication textuelle entre deux clients.

1.4 Plan

1. Dans un premier temps, afin de mieux comprendre le fonctionnement des réseaux pair à pair, nous allons étudier et présenter brièvement les différents les solutions existantes
2. Dans un second temps, nous étudierons les différentes façons de mettre en place une communication textuelle temps réel entre deux clients dans un réseau pair à pair. Nous verrons qu'il existe plusieurs solutions possibles telles que l'utilisation de sockets, l'utilisation de WebRTC ou encore l'utilisation de WebSockets.
3. Dans un troisième temps, présenterons la l'alternative avec WebRTC que nous avons implémenter. Nous expliquerons les raisons de ce choix et nous détaillerons l'implémentation de notre solution. Avant de tenter la réalisation de cette même solution via des sockets. Nous verrons dans cette partie les difficultés rencontrées, les solutions apportées et les limites de notre implémentation.
4. Enfin, nous conclurons sur les résultats obtenus et nous évoquerons les perspectives d'amélioration de notre solution.

Chapitre 2

Contexte et Motivation

Dans le domaine des communications réseaux pair à pair, il est souvent nécessaire d'établir une communication directe entre deux clients. Cependant, certains défis se posent lors de la mise en place d'une telle communication, notamment en présence de dispositifs de traduction d'adresses réseau (NAT) qui limitent la connectivité directe entre les clients.

2.1 Problèmes liés aux dispositifs de traduction d'adresses réseau (le NAT)

Les dispositifs de traduction d'adresses réseau (NAT) sont couramment utilisés pour partager une seule adresse IP publique entre plusieurs appareils d'un réseau local. Les NAT peuvent être trouvés dans de nombreux environnements, tels que les réseaux domestiques, les réseaux d'entreprises et même les réseaux mobiles.

Lorsqu'un client se trouve derrière un NAT, il reçoit une adresse IP privée non routable, qui n'est pas accessible depuis Internet. Cela rend difficile l'établissement d'une communication directe avec d'autres clients qui se trouvent également derrière des NAT.

2.2 La nécessité d'un tiers pour faciliter la communication

Dans le contexte des communications pair à pair, il devient nécessaire d'avoir un tiers (un serveur intermédiaire) pour faciliter la communication entre les clients. Ce tiers agit comme un relai en permettant aux clients de s'échanger des données même s'ils ne peuvent pas établir une connexion directe en raison des NAT.

Le relai joue un rôle crucial en tant qu'entremetteur dans la communication entre les clients. Il permet aux clients de s'enregistrer et de découvrir les autres clients disponibles. Lorsqu'un client souhaite établir une communication avec un autre client, le relai facilite l'établissement de la connexion en relayant les messages entre les clients.

2.3 Solutions existantes utilisant WebSockets et WebRTC

Dans ce contexte, différentes solutions ont été développées pour permettre une communication efficace entre les clients. Deux technologies largement utilisées sont les WebSockets et WebRTC.

Les WebSockets offrent une communication bidirectionnelle et persistante entre un navigateur Web et un serveur WebSocket. Ils permettent une communication en temps réel, avec une latence réduite et une meilleure efficacité par rapport aux requêtes HTTP traditionnelles.

WebRTC, quant à lui, permet une communication peer-to-peer directe entre les navigateurs Web. Il prend en charge la communication audio, vidéo et de données en temps réel. WebRTC permet aux clients d'établir des connexions directes, contournant ainsi les NAT et réduisant la latence.

Ces technologies offrent des solutions puissantes pour la communication en temps réel dans un réseau pair à pair. Cependant, leur utilisation nécessite la mise en place de mécanismes de signalisation et de relais pour l'établissement des connexions et la transmission des données entre les clients.

2.4 Architecture générale d'un système réseau avec NAT

Dans un système réseau avec dispositifs de traduction d'adresses réseau (NAT), l'architecture générale implique plusieurs composants qui interagissent pour permettre la communication entre les clients. Voici les principaux composants et leur rôle :

Clients : Les clients sont les dispositifs ou les applications qui souhaitent communiquer entre eux. Ils peuvent être des ordinateurs, des téléphones, des tablettes ou tout autre appareil connecté au réseau. Les clients peuvent être situés derrière des NAT, ce qui limite leur capacité à établir une communication directe avec d'autres clients.

Serveur relai : Le serveur relai agit en tant qu'entremetteur dans la communication entre les clients. Lorsqu'un client souhaite communiquer avec un autre client, il s'enregistre auprès du serveur relai et communique ses informations de connexion. Le serveur relai joue un rôle clé dans la découverte des clients disponibles et facilite l'établissement de la communication en relayant les messages entre les clients.

Mécanisme de signalisation : Le mécanisme de signalisation est utilisé pour l'échange d'informations entre les clients et le serveur relai. Il permet aux clients de s'enregistrer auprès du serveur relai, de découvrir les autres clients disponibles et d'échanger les informations nécessaires pour établir une connexion. La signalisation peut se faire via des protocoles tels que le protocole HTTP, les WebSockets ou d'autres protocoles personnalisés.

Mécanisme de traversée NAT : Étant donné que les clients peuvent être situés derrière des NAT, un mécanisme de traversée NAT est nécessaire pour permettre l'établissement de la communication directe entre les clients. Ce mécanisme utilise des techniques telles que le protocole de traversée NAT (NAT traversal protocol) ou les techniques de trous de poinçonnage (hole punching) pour contourner les limitations des NAT et permettre aux clients de se connecter directement.

Protocoles de communication : Une fois que la communication directe est établie entre les clients, les protocoles de communication tels que les WebSockets ou WebRTC peuvent être utilisés pour la transmission des données. Ces protocoles permettent une communication bidirectionnelle en temps réel, avec une latence réduite et une efficacité améliorée par rapport aux protocoles traditionnels.

En combinant ces composants et mécanismes, l'architecture générale d'un système réseau avec NAT permet aux clients de surmonter les limitations de connectivité causées par les dispositifs de traduction d'adresses réseau. Cela permet aux clients de communiquer efficacement même s'ils sont situés derrière des NAT, en utilisant des serveurs relais et des mécanismes de signalisation et de traversée NAT

appropriés.

Chapitre 3

Comment mettre en place une connexion pair à pair

3.1 Avec WebRTC

L'utilisation de WebRTC est l'une des solutions possibles pour mettre en place une communication textuelle en temps réel entre deux clients dans un réseau pair à pair. Pour établir cette connexion, les étapes suivantes peuvent être suivies :

Initialisation d'une pair et récupération de son ID Chaque client doit être initialisé en tant que paire et se voit attribuer un ID unique. Cet ID est utilisé pour identifier le pair et permettre à d'autres pairs de s'y'connecter.

Partage des ID entre les deux pairs Une fois que chaque pair a obtenu son ID, ces identifiants doivent être partagés entre les deux pairs afin qu'ils puissent s'identifier mutuellement lors de l'établissement de la connexion. Cela peut être réalisé par l'intermédiaire d'un serveur de signalisation ou d'un canal de communication préalablement établi, tel qu'un serveur centralisé ou un autre moyen de communication sécurisé.

Création d'une connexion entre les deux pairs Une fois que chaque pair dispose de l'ID de l'autre, ils peuvent utiliser ce dernier pour établir une connexion directe entre eux en utilisant le protocole WebRTC.

Echange de données entre les deux pairs Une fois la connexion établie, les deux pairs peuvent s'échanger des données. Ces données peuvent être des messages textuels, des fichiers, des flux audio et vidéo, etc. (Dans notre cas, nous nous intéressons uniquement aux messages textuels.)

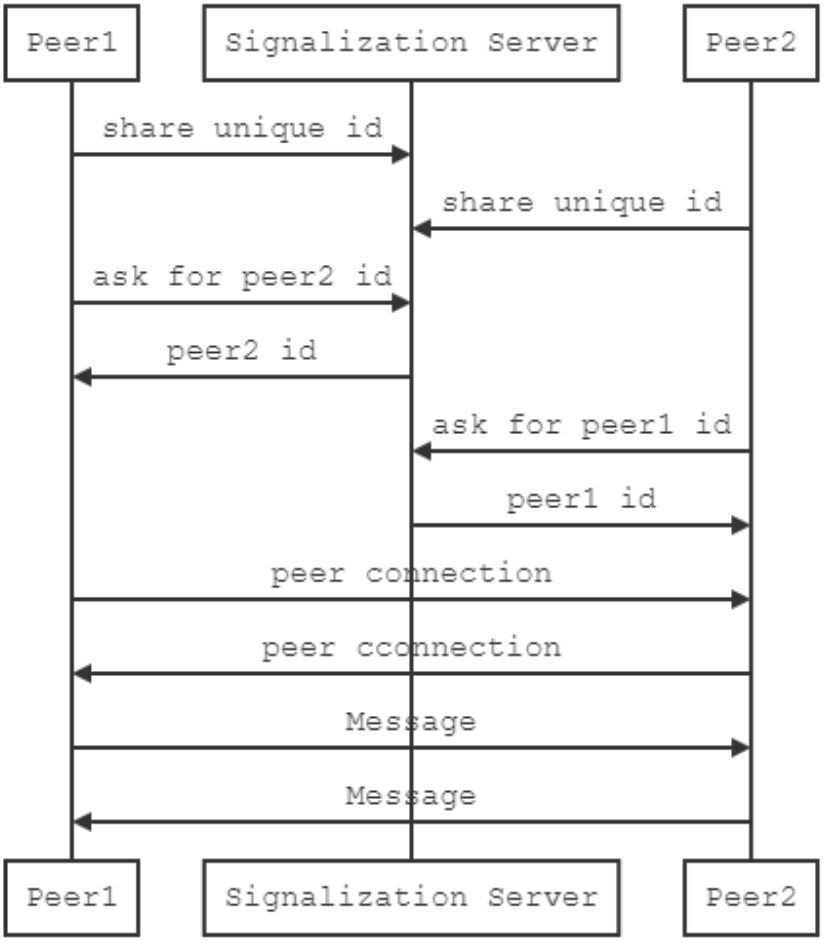


FIGURE 3.1 – Schéma de fonctionnement de WebRTC

3.2 Avec WebSockets

Une autre solution pour mettre en place une communication textuelle en temps réel entre deux clients dans un réseau pair à pair est d'utiliser les WebSockets. La principale difference avec WebRTC, c'est que les Sockets nécessitent la connaissance des adresses IP des deux pairs pour pouvoir établir une connexion entre eux. Pour établir cette connexion, les étapes suivantes peuvent être suivies :

Récupération de l'adresse IP publique du client Pour pouvoir établir une connexion entre deux pairs, il faut que chaque pair connaisse l'adresse IP publique ainsi que le port utilisé par l'autre pair. Mais avant cela, il faut que chaque pair

récupère son adresse IP publique afin de la partager avec l'autre pair. Pour cela, il existe une solution (non unique) qui consiste à utiliser un service STUN qui permet de récupérer l'adresse IP publique du client ainsi qu'un port ouvert sur le pare-feu du client.

Partage des adresses IP entre les deux pairs Une fois ces informations récupérées, chaque pair doit partager son adresse IP publique ainsi que le port ouvert sur son pare-feu avec l'autre pair. Cela peut être réalisé par l'intermédiaire d'un serveur de signalisation ou d'un canal de communication préalablement établi, tel qu'un serveur centralisé ou un autre moyen de communication sécurisé.

Création d'une connexion entre les deux pairs Une fois que chaque pair dispose de l'adresse IP publique et du port ouvert sur le pare-feu de l'autre pair, ils peuvent utiliser ces informations pour établir une connexion directe entre eux.

Echange de données entre les deux pairs Une fois la connexion établie, les deux pairs peuvent s'échanger des données. Ces données peuvent être des messages textuels, des fichiers, des flux audio et vidéo, etc. (Dans notre cas, nous nous intéressons uniquement aux messages textuels.)

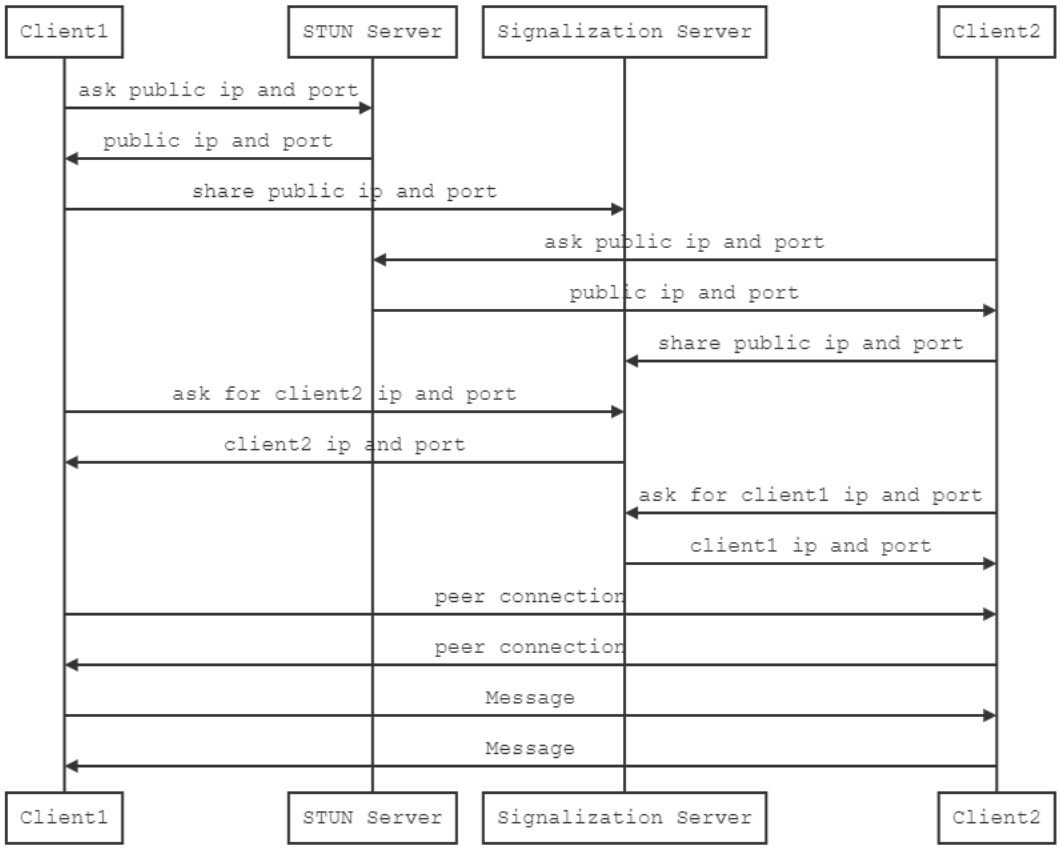


FIGURE 3.2 – Schéma de fonctionnement de WebSockets

3.3 Les outils tiers nécessaires pour la communication pair à pair

3.3.1 Les serveurs de signalisation

Pour établir une communication pair à pair entre deux clients, il est souvent nécessaire de mettre en place des serveurs de signalisation. Les serveurs de signalisation agissent comme des intermédiaires pour faciliter l'établissement de la connexion initiale entre les clients.

Lorsqu'un client souhaite établir une communication avec un autre client, il doit d'abord échanger des informations telles que les adresses IP et les ports. Les serveurs de signalisation permettent aux clients de partager ces informations entre eux. Ils peuvent également aider à coordonner les étapes de négociation et d'établissement de la connexion.

Les serveurs de signalisation peuvent utiliser différents protocoles et techniques pour échanger les informations nécessaires. Par exemple, ils peuvent utiliser des protocoles basés sur le web tels que HTTP pour envoyer des messages de signalisation entre les clients. Ils peuvent également utiliser des protocoles de signalisation spécialisés tels que SIP (Session Initiation Protocol) ou XMPP (Extensible Messaging and Presence Protocol).

Il existe différentes options pour mettre en place des serveurs de signalisation, allant des solutions prêtes à l'emploi aux serveurs personnalisés développés en interne. Certains services cloud offrent également des fonctionnalités de signalisation pour faciliter l'établissement de connexions pair à pair.

Il est important de noter que les serveurs de signalisation ne sont généralement pas impliqués dans le transfert direct des données entre les clients. Ils jouent simplement un rôle de coordination pour permettre l'établissement de la connexion initiale.

3.3.2 Les protocoles TURN et STUN

Pour faciliter les communications pair à pair, il est souvent nécessaire de prendre en compte les contraintes de réseau telles que les pare-feu et les NAT (Network Address Translation). Deux protocoles couramment utilisés pour résoudre ces problèmes sont TURN (Traversal Using Relays around NAT) et STUN (Session Traversal Utilities for NAT).

Le protocole STUN permet à un client d'obtenir des informations sur son adresse IP publique et le type de NAT qu'il traverse. Cela permet au client de comprendre comment il est connecté au réseau et d'adapter ses stratégies de connexion en conséquence. STUN est généralement utilisé pour faciliter l'établissement de connexions pair à pair lorsque les clients sont derrière des NAT symétriques ou des pare-feu restrictifs.

Le protocole TURN est utilisé lorsque les connexions pair à pair directes ne sont pas possibles en raison de pare-feu ou de NAT restrictifs. TURN utilise un serveur intermédiaire (relais) auquel les clients se connectent pour acheminer leurs données. Ce serveur intermédiaire agit comme un proxy et transfère les données entre les clients. Cela permet aux clients de communiquer même s'ils ne peuvent pas établir une connexion directe.

Lorsqu'un client rencontre des difficultés pour établir une connexion pair à pair directe en raison des restrictions du réseau, il peut utiliser STUN pour obtenir son adresse IP publique, puis essayer d'établir une connexion directe. Si cela échoue, le client peut passer à l'utilisation de TURN pour acheminer les données via un serveur relais.

Il est important de noter que l'utilisation de TURN peut entraîner une latence supplémentaire car les données doivent être acheminées via le serveur relais. De plus, l'utilisation d'un serveur relais peut entraîner des coûts supplémentaires, notamment si vous utilisez un service cloud ou un fournisseur tiers pour héberger le serveur.

En résumé, les protocoles TURN et STUN sont des outils importants pour faciliter les communications pair à pair lorsque les clients sont derrière des NAT restrictifs ou des pare-feu. Ils permettent de contourner les contraintes réseau et de faciliter l'établissement de connexions directes ou le transfert des données via un serveur relais lorsque cela est nécessaire.