UNIVERSIDADE FEDERAL DE LAVRAS
Departamento de Ciência da Computação

TRABALHO PRÁTICO
try-p2p

Bruno Queiroz Santos - 201621166

GCC129 - Sistemas Distribuídos
Lavras, 10 de novembro de 2018

1. Introdução

O objetivo desse trabalho é criar uma rede P2P (Peer-to-peer) que distribua informações escritas em arquivos texto. A aplicação não será p2p puro, ela terá um servidor de controle parecido com a arquitetura do Napster, o objetivo desse servidor será guardar os ips e as portas dos nós presentes na rede que estão servindo algum arquivo texto para rede.

A principal motivação deste trabalho é compreender o quão é difícil projetar uma aplicação p2p, visto que existem vários desafios que precisam ser resolvidos.

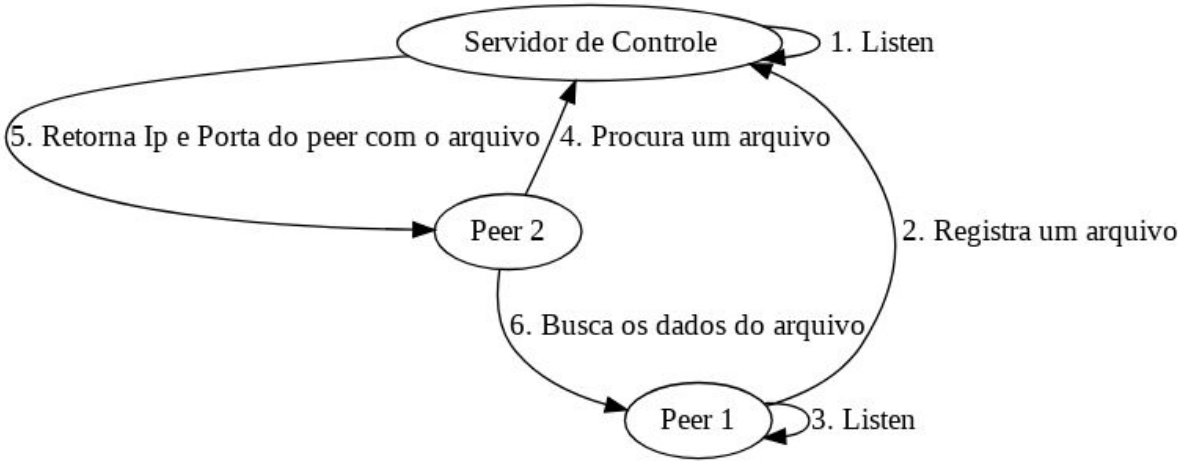
2. Descrição geral do sistema

Como dito anteriormente a aplicação será composta por um 1 servidor de controle e vários peers que se conectarem a rede. Para rodar a aplicação será necessário ligar o servidor de controle para depois ligar os peers, assim o usuário que estiver com o peer ligado poderá escolher 2 opções: Fazer download de um arquivo; ou servir seu arquivo para que algum outro peer possa baixar.

Quando o usuário escolher a opção de download, ele irá buscar esse arquivo no servidor de controle, caso esse arquivo exista o servidor de controle retorna o ip e a porta do peer que está com o arquivo, só então ele conseguirá baixar o arquivo.

Quando o usuário escolher a opção de servir o arquivo, ele irá enviar o nome do arquivo, seu ip e sua porta para o servidor de controle.

3. Diagrama de fluxo da aplicação



4. Requisitos Funcionais

- RF1: Servidor de controle rodando
- RF2: Registrar peer no servidor de controle
- RF3: Procurar peer no servidor de controle
- RF4: Download do arquivo que um peer estiver servindo
- RF5: Servidor procurar dado na sua estrutura de dados
- RF6: Servidor registrar dados na sua estrutura de dados

4.1. Requisitos Não-funcionais

- RNF1: Facilidade em executar o código para entender conceitos básicos sobre p2p.
- RNF2: Criptografar a mensagem transmitida entre peers e servidor de controle.

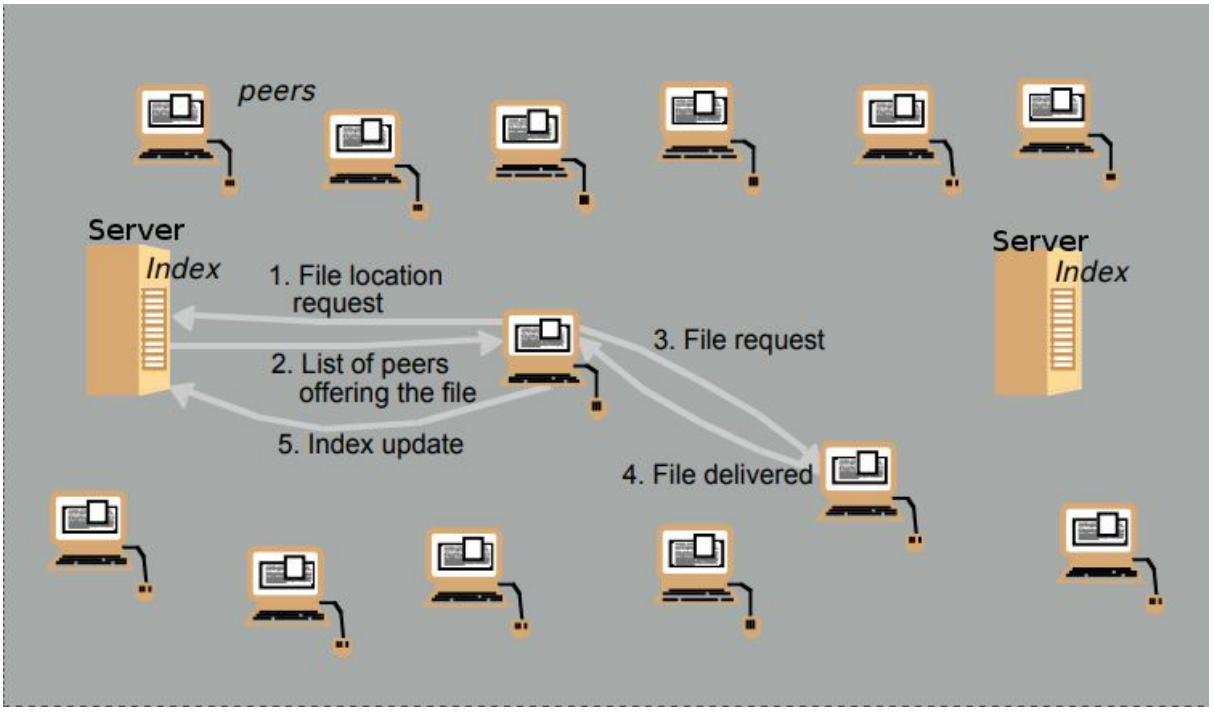
5. Desafios

Um dos principais desafios desse trabalho será fazer essa aplicação funcionar em redes que utilizam NAT. Outro desafio seria transportar os dados criptografados utilizando sockets.

6. Projeto e implementação

O sistema em si foi feito utilizando a técnica orientação a objetos, sua estrutura foi dividida em 2 classes: Peer e Server, sendo que a classe Peer efetua as seguinte ações: upload de arquivos para o server, e download de arquivos, já a classe Server armazena todos os dados da rede, esses dados são os ips de cada Peer que está compartilhando algum arquivo.

O sistema tem um módulo utils, o qual é responsável por criptografar as mensagens enviadas entre Peer <> Peer e Server <> Peer, também é responsável por ler os dados dos arquivos a serem enviados em uma pasta específica (upload_files), e também escrever os dados baixados em um novo arquivo armazenado em outra pasta específica (download_files). Segue abaixo um esboço da arquitetura do projeto:



O sistema foi desenvolvido utilizando a linguagem Python 3 junto com tecnologia de sockets e o protocolo de transporte TCP. O servidor de controle utiliza uma tabela hash (Dicionário em python), o algoritmo utilizado para hash é o MD5, sua estrutura base é:

{‘Hash md5 do nome do arquivo’: [[‘Ip do peer a onde o arquivo está guardado’, ‘Tipo de operação que peer irá efetuar (Registro ou Pesquisa) ’, ‘Hash md5 do nome do arquivo’, ‘Nome do arquivo’, ‘porta que o peer está escutando’]] }

Seguindo a lógica das estruturas de dados do python a estrutura é um dicionário com uma lista de lista em cada índice do dicionário, a lista de lista é útil para quando ocorrer colisão de hash (ocorre quando o nome do arquivo servido pelos peers tem o mesmo nome).

A criptografia utilizada é a AES assimétrica, as chaves pública e privada já são conhecidas pelos Peers e Servidor de controle, então não é necessário o transporte dessas chaves junto com a mensagem criptografada.

A tecnologia de Threads também foi utilizada para que vários Peers possa se conectar de forma paralela com o Servidor de Controle.