

## Documentação do Protocolo IDP2P

Versão 1.0

Data: 2023-12-12

### O que é?

IDP2P é um protocolo simples para compartilhar arquivos. Clientes se conectam a um servidor e podem trocar mensagens entre si para baixar e enviar arquivos.

### Papéis na Rede:

- **Servidor:** Gerencia conexões de clientes e registra arquivos compartilhados. Sempre online.
- **Cliente:** Conecta ao servidor e a outros clientes, baixa e envia arquivos.

### Mensagens:

- **Client para Servidor:**
  - **REG:** Registra com senha, porta e arquivos compartilhados.
  - **UPD:** Atualiza arquivos compartilhados.
  - **LST:** Lista todos os arquivos compartilhados e suas localizações.
  - **END:** Desconecta do servidor.
- **Servidor para Client:**
  - **OK/ERR:** Confirmação/erro para ações do cliente.
  - **Dados do Arquivo:** Conteúdo do arquivo solicitado após receber GET válido.

### Compartilhamento de Arquivos:

- Clientes identificam arquivos pelo hash MD5.
- O servidor rastreia quais clientes possuem cada arquivo.
- Clientes baixam diretamente de outros clientes usando suas portas listadas.

### Pontos Principais:

- Clientes se conectam diretamente para transferência de arquivos.
- Hash MD5 garante integridade do arquivo, além de diferenciar arquivos com conteúdos diferentes e o mesmo nome.

### Implementação:

#### Servidor:

- **Configurações do servidor:**
  - O servidor está configurado para escutar conexões na porta 54494.
  - O servidor usa o protocolo UDP para comunicação com os clientes.
- **Funções de processamento de mensagens:**
  - A função `registrar_cliente()` registra um novo cliente no servidor, e é chamada quando um cliente se conecta ao servidor através da mensagem REG <Senha> <Porta> <Arquivos> .
  - A função `atualizar_cliente()` atualiza as informações de um cliente já registrado, e é chamada quando recebe um UPD <Senha> <Porta> <Arquivos>.
  - A função `listar_arquivos()` lista todos os arquivos compartilhados pelos clientes registrados, entra em ação quando o cliente manda a mensagem LST.
  - A função `desconectar_cliente()` desconecta um cliente do servidor, quando o servidor recebe a mensagem END <Senha> <Porta>.

Cada função de processamento de mensagens verifica o formato da mensagem recebida do cliente. Se o formato estiver correto, a função executa a ação apropriada.

- **Recursos do servidor:**

- Usa o protocolo UDP para comunicação.
- Verifica os erros de entrada.
- Utiliza uma senha para autenticar os clientes.

**Aqui está um exemplo de fluxo de trabalho do servidor:**

O servidor inicia ouvindo conexões na porta 54494. Quando um cliente se conecta, o servidor recebe uma mensagem do cliente. O servidor então processa a mensagem e envia uma mensagem de resposta ao cliente. O servidor continua a receber mensagens de clientes e processá-las até que o servidor seja interrompido.

**Cliente:**

**Funcionalidades básicas:**

- O cliente se conecta a um servidor de controle por meio do protocolo UDP.
- O cliente se conecta a outros clientes por meio do protocolo TCP para compartilhar seus arquivos.
- O cliente fornece um menu para que o usuário possa atualizar a lista de arquivos disponíveis, baixar outros arquivos e se desconectar.

**Implementação:**

- O cliente usa a biblioteca padrão do Python para sockets.
- O cliente usa a função `md5()` para gerar hashes de arquivos.
- O cliente usa threads para executar atividades paralelamente, como se comunicar com o servidor e se conectar a outros clientes para receber arquivos, assim como abrir uma conexão para receber requisições de arquivos de outros clientes.

**Exemplo de uso:**

```
# Executar o cliente com o IP e o diretório de arquivos
python cliente.py 127.0.0.1 /home/usuario/arquivos
# Selecionar a opção "Atualizar arquivos disponíveis" (Input: 1)
# Selecionar a opção "Baixar um arquivo" (Input: 2)
# Selecionar um arquivo da lista (Input: 1)
# Selecionar um host para baixar o arquivo (Input: 2)
# O arquivo será baixado para o diretório /home/usuario/arquivos
```

**Sugestões de melhorias:**

- Implementar segurança, como autenticação e criptografia.
- Implementar recursos adicionais, como pesquisa ou grupos de usuário

Além deste documento, o código possui comentários para auxiliar sua leitura e a compreensão das funcionalidades auxiliares implementadas para o funcionamento do protocolo.