

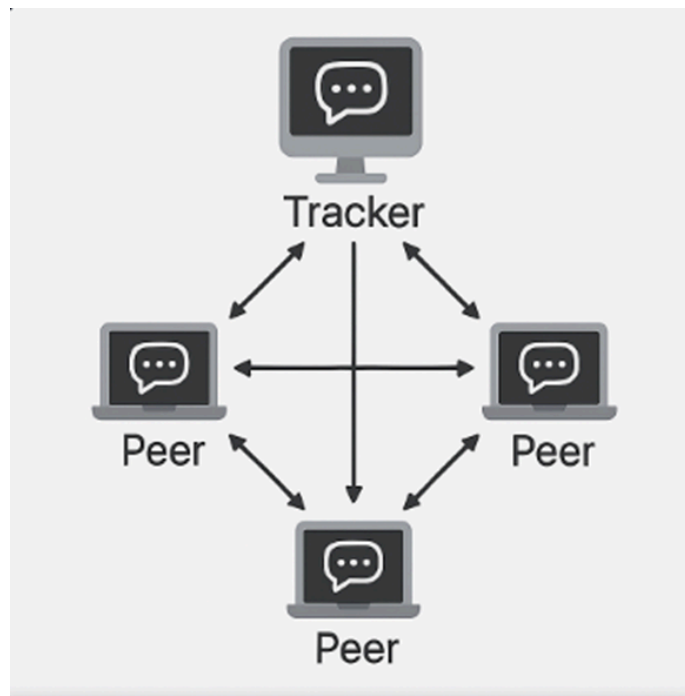
Relatório Técnico – MiniBit

Grupo: Hudson Cunha, João Pedro, Lucas França e Renato Neves

1. Visão Geral da Arquitetura

O sistema adota uma arquitetura distribuída com um tracker centralizado e múltiplos peers que se conectam inicialmente com o tracker e após, com os demais peers diretamente para compartilhar blocos de arquivos.

2. Diagrama da Arquitetura



- O tracker é responsável por registrar e informar quais peers estão ativos e distribuir os blocos iniciais.
- Os peers trocam blocos entre si diretamente, sem mediação do tracker.
- Toda a lógica de troca de blocos, rarest-first e tit-for-tat ocorre entre os peers.

3. Funções do Tracker

a. Registro de Peers:

- i. Cada peer, ao iniciar, envia ao tracker seu IP e porta.
- ii. O tracker armazena essas informações em memória.

b. Descoberta de Peers:

- i. **Quando um peer consulta o tracker, ele recebe de volta:**
 1. Uma lista aleatória de até 4 peers ativos ou todos, se menos houverem menos de 5 peers.(exceto ele mesmo)
 2. Os seus blocos iniciais.
- ii. Essa resposta inclui também os blocos disponíveis de cada peer retornado.

4. Estratégia: Rarest First

A estratégia Rarest First é implementada por cada peer ao decidir quais blocos baixar primeiro.

- Cada peer mantém uma lista de blocos disponíveis entre os peers que conhece.
- Ele calcula a frequência de cada bloco na rede.
- Blocos menos comuns (mais raros) recebem maior prioridade de download.
- Isso garante melhor distribuição dos blocos, evitando que um bloco fique escasso ou perdido.

Objetivo: Evitar gargalos e garantir que todos os blocos estejam amplamente distribuídos.

5. Estratégia: Tit-for-Tat (Olho por Olho)

Essa estratégia é usada para controlar quem pode trocar blocos com quem, e tenta equilibrar cooperação e justiça:

- A cada 10 segundos, o peer:
 - Desbloqueia (unchoke) até 4 peers com os quais teve boa troca (com blocos raros).
 - Escolhe 1 peer aleatório (optimistic unchoke) para testar novas conexões.
- Se esse novo peer provar ter blocos raros, pode ser promovido a “prioritário”.

6. Resultado de testes:

7. Reflexões individuais e dificuldades encontradas:

Hudson:

A maior dificuldade foi garantir que todos os blocos da imagem fossem trocados entre os peers e reconstituídos na ordem certa. Ao rodar o código alguns peers ficavam travados com blocos faltando recebendo repetidamente logs como “peer D: Progresso: 3/8 blocos (37.5%)”, impedindo a conclusão do download, mas após ajustes no código podemos resolver esse problema.

João Pedro:

Como trabalho com DevOps, já fazia um tempo que eu não desenvolvia algo diretamente. Entender como eu iria construir a minha parte no projeto, especialmente a reconstrução do arquivo e a garantia da integridade dos dados, foi um desafio que exigiu um certo tempo.

Lucas:

Foi certamente um grande desafio pensar na elaboração de cada funcionalidade, de como priorizar cada bloco e como escrever o algoritmo, mesmo já sendo acostumado a desenvolver software diariamente. No entanto, estou satisfeito com o resultado final, acredito que tenha ficado satisfatório apesar das dificuldades encontradas.

Renato:

Pessoalmente não tenho experiência e atuação profissional com projetos de programação, trabalho na área de redes e infraestrutura. Por essa razão, a implementação do código, no geral, foi desafiadora; e por motivo semelhante; o entendimento da comunicação e os protocolos se mostraram mais familiares com as minhas experiências cotidianas. O projeto em si, além de relevante por simular o funcionamento de uma ferramenta conhecida e muito usada no dia a dia, permitiu exercitar de forma prática diversos conteúdos mostrados em sala.