

A transferência de arquivos usando o protocolo UDP, mesmo sem apresentar erros aparentes, pode falhar ao transferir arquivos grandes por diversos motivos relacionados à natureza do protocolo e à forma como o programa foi implementado.

Os programas udp que usam timeout e tamanho não transferem arquivos muito grandes como deveriam porque o protocolo UDP é um protocolo de transporte leve que não garante confiabilidade, ele não verifica se os pacotes foram entregues, se chegaram na ordem correta ou se foram corrompidos. Então, em aplicações como transferência de arquivos isso pode significar que alguns pacotes podem ser perdidos durante a transmissão/alguns pacotes podem não chegar ao destino. Além disso, Pacotes podem ser duplicados ou chegar fora de ordem. o protocolo UDP não faz controle de congestionamento: arquivos grandes geram muitas transmissões, o que pode saturar a rede e levar à perda de pacotes.

O uso de timeout é uma solução eficaz para lidar com possíveis perdas de pacotes no UDP. No entanto, ele não é eficaz para arquivos grandes por alguns motivos:

1. Dificuldade em configurar o timeout ideal: Um timeout curto pode gerar muitas retransmissões desnecessárias, enquanto um timeout longo pode atrasar a transferência.
2. Ausência de confirmação: Sem confirmações específicas de cada segmento, é difícil garantir que todos os pedaços de um arquivo grande foram entregues corretamente.
3. Tamanho dos pacotes: O UDP impõe um limite ao tamanho dos pacotes (geralmente 512 bytes). Arquivos grandes precisam ser fragmentados em múltiplos pacotes, e o timeout não garante que todos sejam reconstruídos adequadamente.

Enviar o tamanho total do arquivo é uma tentativa de contornar os problemas de fragmentação e controle, mas também apresenta desafios: O tamanho do arquivo precisa ser enviado em formato binário para que o receptor o interprete corretamente. Isso exige o uso de funções como `int.to_bytes` e `int.from_bytes`. Sem essa conversão, a interpretação do tamanho pode ser incorreta, resultando em erros de reconstrução do arquivo.

O servidor precisa informar corretamente o tamanho do arquivo ao cliente. Se essa informação for corrompida ou perdida, o cliente não saberá quando parar de receber dados, o que pode levar a arquivos incompletos ou corrompidos.

O UDP não retransmite pacotes perdidos, causando lacunas nos dados recebidos.