

Redes de Computadores - Relatório de Projeto

Gonçalo Bastos nº2020238997

Maio 2023

1 Introdução

UDP é um protocolo sem conexão e uma estrutura importante na arquitetura TCP/IP. Sockets permitem comunicação entre dois processos diferentes em dispositivos conectados à Internet, seja no mesmo dispositivo ou em dispositivos diferentes. Uma grande vantagem do UDP é sua velocidade, sendo muito mais rápido, simples e eficiente em comparação com o TCP. No entanto, a retransmissão de pacotes de dados perdidos é possível apenas com o TCP.

O objetivo deste projeto é implementar uma aplicação de transferência de arquivos sobre UDP, assumindo que a transferência ocorra em uma rede local e que a confiabilidade não seja uma preocupação.

A aplicação consiste em um servidor que abre um socket e ouve a chegada de dados em uma porta específica, e um cliente que lê um arquivo de texto e transfere o arquivo para o servidor. O tamanho máximo de um datagrama que UDP pode aceitar é de 500 bytes e a porta escolhida foi a 9876 . Quando o servidor recebe os dados do cliente, ele grava os dados em um arquivo. O objetivo deste relatório é detalhar a implementação da aplicação de transferência de arquivos, os desafios enfrentados durante a implementação e os resultados gerais do projeto.

2 Implementação

Nesta secção irei explicar como implementei a aplicação:

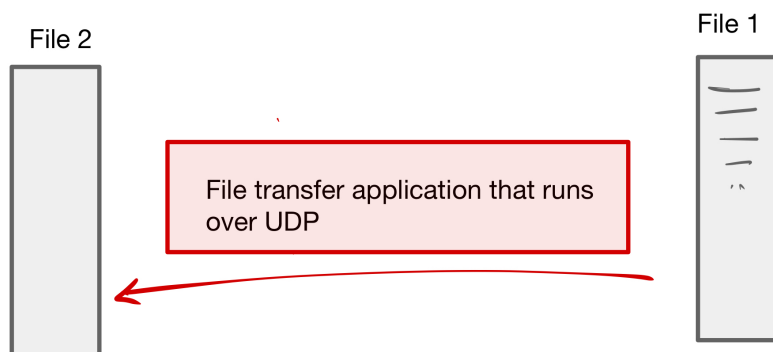


Figure 1: Esquema ilustrativo da aplicação

Vou explicar como implementei a aplicação de transferência de arquivos que funciona sobre o protocolo UDP. O servidor abre um socket e aguarda a chegada de dados em uma porta específica, enquanto o cliente lê um arquivo de texto e transfere o conteúdo para o servidor. Quando o servidor recebe os dados do cliente, ele os escreve em um arquivo.

O código do **cliente** cria primeiro um socket, usando a função `socket()`, para enviar pacotes por UDP. Em seguida, ele abre o arquivo de origem do conteúdo a ser enviado usando a função `fopen()`, lê o arquivo usando `fread()` e envia pacotes de dados para o servidor usando `sendto()`. Cada pacote inclui um número de sequência no início do buffer, que é usado pelo servidor para garantir que os dados sejam recebidos na ordem correta. Após o envio de um pacote, o cliente aguarda um reconhecimento do servidor antes de enviar o próximo pacote. Se um reconhecimento não for recebido em um tempo razoável, o cliente reenvia o pacote. Depois que todos os pacotes forem enviados e reconhecidos, o cliente fecha o arquivo e o socket.

O código do **servidor** cria primeiro um socket para receber pacotes por UDP. Ele abre o arquivo de destino do conteúdo, associa o socket ao endereço de dados e vincula um número de porta 9876. Em seguida, aguarda a chegada de conexões. Quando um pacote é recebido, o servidor verifica o número de sequência e envia um reconhecimento de volta para o cliente com o número de reconhecimento esperado. Se o número de sequência não corresponder ao número de sequência esperado, o servidor ignora o pacote e reenvia o reconhecimento anterior. Se o número de sequência corresponder ao número de sequência esperado, o servidor escreve os dados no arquivo e incrementa o número de

sequência esperado.

O próximo esquema (Fig2) tem como objetivo demonstrar a implementação:

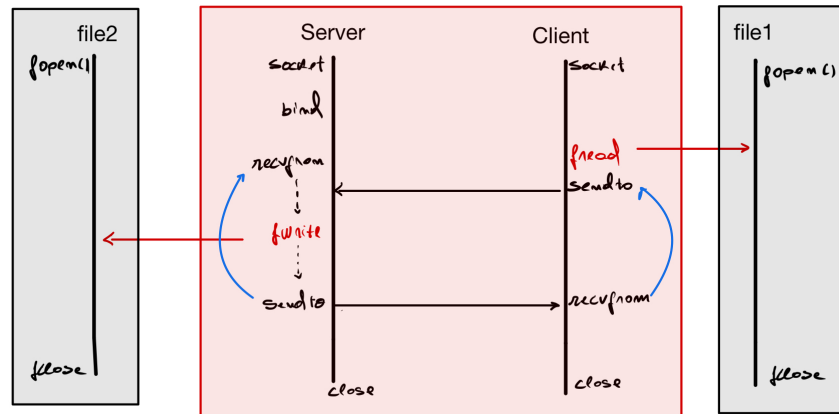


Figure 2: Esquema mais detalhado da aplicação

Para testar esta implementação utilizei o file1.txt onde contem um poema de Shakespeare que será enviado para o file2.txt, de forma a verificar que a mesma esta correta.

3 Conclusão

Em conclusão, o projeto de transferência de arquivo implementado utilizando o protocolo UDP foi bem-sucedido. O cliente conseguiu enviar o arquivo para o servidor por meio de datagramas UDP com tamanho máximo de 500 bytes. O servidor, por sua vez, conseguiu receber esses datagramas, escrevendo-os em um arquivo no destino especificado.