

Implementação de servidores do tipo TCP e UDP

Breno Esteves, Gabriel de Paula, Guilherme Francis

Universidade Federal de São João Del Rei
São João del Rei / MG - Brasil

{brenoesteves243,gabriel.meira.2004,guilherme2036}@aluno.ufsj.edu.br

Resumo. *A comunicação em redes é um componente indispensável de qualquer sistema distribuído e aplicação moderna, se baseando principalmente em dois protocolos de transporte: TCP (Transmission Control Protocol) e UDP (User Datagram Protocol). Ambos diferem em seus aspectos de confiabilidade e desempenho e, portanto, têm um impacto diferente nas aplicações que os utilizam. Enquanto o primeiro é mais reconhecido por sua confiabilidade e capacidade de controlar os erros, o segundo garante alta eficiência ao custo de possíveis perdas de dados. Este trabalho envolve a implementação de servidores e clientes para entender o funcionamento detalhado desses protocolos, associado à análise das estatísticas de taxa de transferência e perda de pacotes.*

1. Introdução

A comunicação eficiente e segura entre sistemas é um dos pilares fundamentais no desenvolvimento de aplicações distribuídas, especialmente em um mundo cada vez mais conectado. Este trabalho apresenta a implementação de um sistema de transferência de arquivos utilizando os protocolos TCP e UDP, que proporcionam uma interface simples e robusta para clientes e servidores.

A solução foi desenvolvida utilizando a linguagem de programação C, priorizando um design modular, reutilizável e de fácil manutenção. O sistema permite a transferência de arquivos tanto em redes locais quanto remotas, atendendo a cenários variados. Além disso, incorpora técnicas avançadas para monitorar e analisar o desempenho da transferência, incluindo métricas como tempo total, velocidade de transmissão e taxa de perda de pacotes. Essas estatísticas proporcionam uma visão detalhada e prática da eficiência do sistema, contribuindo para sua avaliação e aprimoramento.

2. Desenvolvimento

O desenvolvimento do sistema proposto organiza-se em dois componentes principais: cliente e servidor. Cada componente opera com lógica independente, mas mantém uma interface bem definida para comunicação.

A finalidade da implementação é fazer com que o cliente solicite um arquivo que, caso exista no servidor, é enviado em pacotes de 4KB. Para realizar essa tarefa, o sistema teve de ser estruturado separando a lógica que trabalha com a leitura e escrita dos arquivos da que organiza o envio e recebimento dos pacotes. Essa divisão clara não apenas facilita a manutenção e escalabilidade do sistema, mas também promove a reutilização do código e a separação de responsabilidades, contribuindo para um desenvolvimento mais fluido.

2.1. Arquitetura do Sistema

Ao todo foram criados quatro programas principais e independentes entre si, sendo eles o produto cartesiano entre os conjuntos {servidor, cliente} e {TCP, UDP}, cada um com suas necessidades específicas de implementação.

Apesar de possuírem tantas diferenças, muitos aspectos dos dois protocolos se repetem, sendo importante a definição de utilitários para consolidar uma base estável, oferecendo funções auxiliares, como a manipulação de arquivos, o cálculo de estatísticas de desempenho, além de realizar a exibição de mensagens informativas de erro ou sucesso.

2.1.1. Protocolo MONDA

“Manda O Nome Do Arquivo” ou simplesmente “MONDA” foi o apelido dado para o protocolo da camada de aplicação criado para definir como os clientes devem solicitar arquivos aos servidores. Apenas um arquivo deve ser requisitado por comunicação, informando seu nome, por exemplo: “arquivo.txt”. Caso não exista no diretório do servidor, é enviado um pacote apenas com a mensagem “ERROR - FILE NOT FOUND”.

No caso específico do UDP, os 6 primeiros bytes são reservados para identificar o pacote (3 bytes) e identificar o total de pacotes (3 bytes), dessa forma, é possível enviar arquivos em até 2^{24} pacotes.

2.2. TCP

O *Transmission Control Protocol*, abreviado comumente para TCP, é um protocolo orientado à conexão, o que significa que estabelece uma comunicação confiável entre as partes antes de transmitir os dados. Ele assegura a entrega completa e na ordem correta, tornando-se ideal para cenários onde a integridade dos dados é crucial.

O fluxo ideal da implementação do protocolo TCP, representado na Figura 1, descreve o funcionamento e as interações entre os componentes cliente e servidor, garantindo uma comunicação confiável e ordenada.

No lado do servidor, o processo inicia com a criação e vinculação do socket a uma porta específica, seguido pela escuta ativa de conexões provenientes de clientes. Quando uma solicitação é recebida, o servidor busca o arquivo solicitado, envia os pacotes correspondentes e, ao final, encerra a conexão de forma segura.

Por sua vez, no lado do cliente, o fluxo começa com a criação do socket de comunicação, seguido pela conexão ao servidor. O cliente solicita o arquivo desejado enviando o nome correspondente e, em seguida, aguarda os pacotes enviados pelo servidor. O cliente reconstrói o arquivo juntando os dados de cada pacote recebido e, ao final da transferência, fecha o socket.

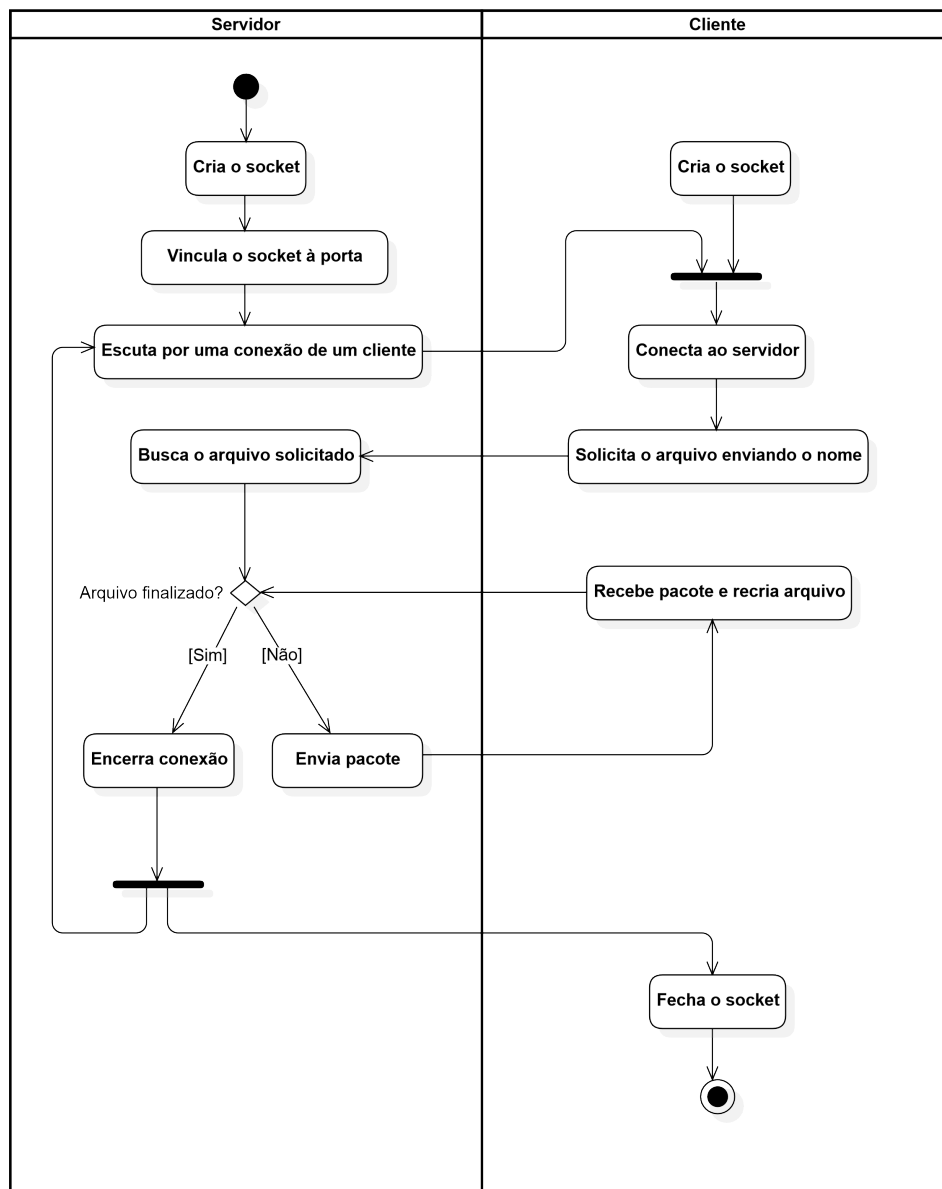


Figura 1. Representação do fluxo ideal da implementação do TCP

2.3. UDP

Em contrapartida, o *User Datagram Protocol* é um protocolo sem conexão, priorizando velocidade e eficiência em detrimento da confiabilidade. Por não realizar controle de fluxo ou retransmissões como o TCP, o UDP é indicado para contextos onde o desempenho é mais importante do que a entrega garantida, como transmissões em tempo real em que pequenas perdas são desprezíveis.

O fluxo ideal da implementação do protocolo UDP, representado na Figura 2, apresenta as etapas de comunicação entre o cliente e o servidor. No lado do servidor, o fluxo inicia com a criação e vinculação do socket a uma porta específica, assim como o TCP. Em seguida, o servidor aguarda uma solicitação de um cliente. Após receber o pedido, o servidor busca o arquivo solicitado e envia os pacotes correspondentes, sem a necessidade de confirmar a entrega ou mesmo se preocupar com a ordem.

No lado do cliente, o fluxo também começa com a criação do socket. Em seguida, envia uma solicitação ao servidor, indicando o arquivo desejado. Em seguida, o cliente recebe os pacotes transmitidos pelo servidor e reconstrói o arquivo localmente. Após um intervalo de 2 segundos sem o recebimento de pacotes adicionais, o cliente considera a operação como concluída e fecha o socket.

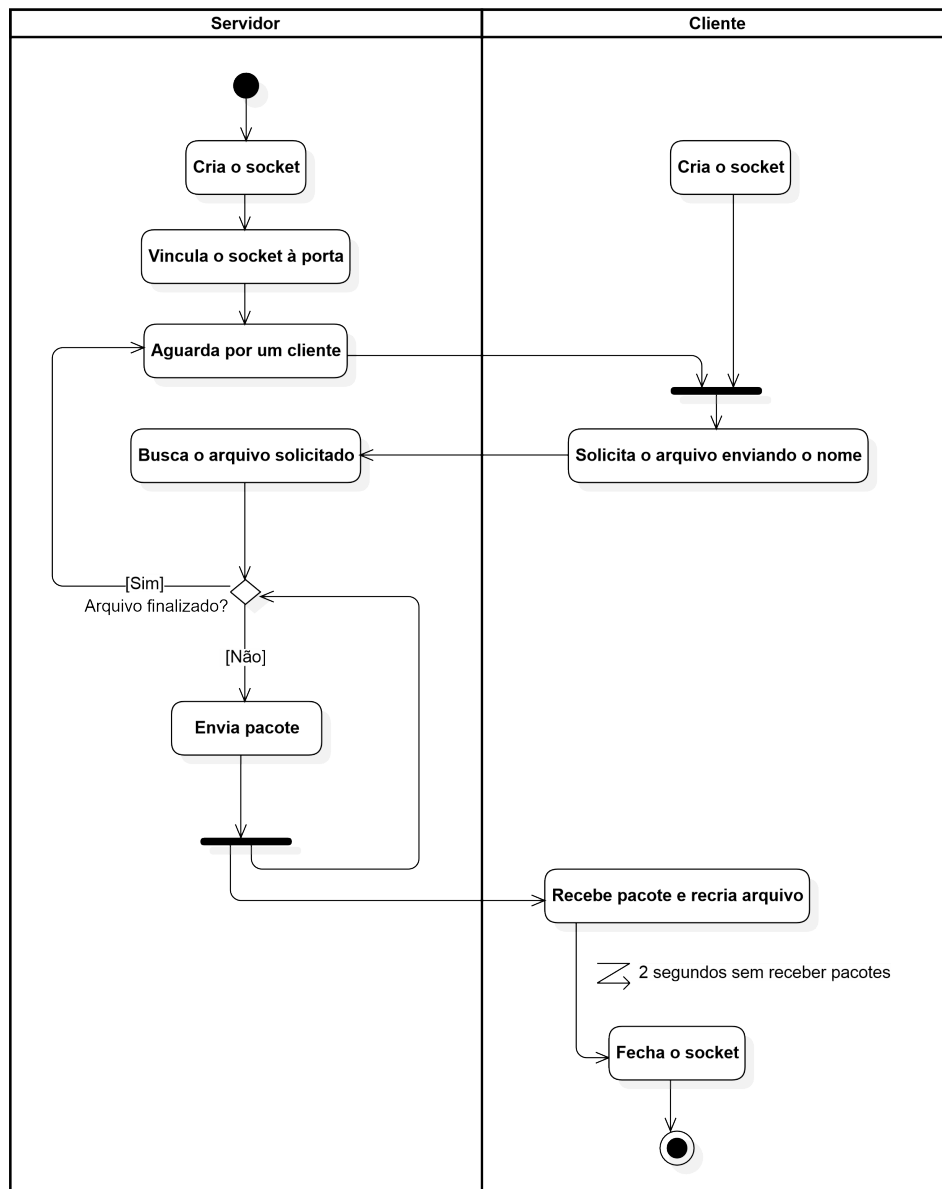


Figura 2. Representação do fluxo ideal da implementação do UDP

3. Resultados

Um aspecto fundamental no estudo de sistemas de comunicação em rede é a avaliação de desempenho. Neste estudo, propõe-se a utilização de métricas como tempo de transferência (segundos), taxa de transferência (bytes por segundo) e taxa de perda de pacotes.

As estatísticas de desempenho do sistema são obtidas por meio de medições precisas realizadas durante o processo de transmissão. O tempo de transferência é calculado como a diferença entre os momentos de início e término da comunicação, fornecendo uma métrica clara da duração total do envio de dados. A taxa de transferência, por sua vez, é derivada dividindo o tamanho total do arquivo transferido pelo tempo de transferência, o que permite avaliar a eficiência do sistema em termos de velocidade de envio de dados. Por fim, a taxa de perda de pacotes é determinada ao comparar o número de pacotes enviados pelo servidor com o número de pacotes efetivamente recebidos pelo cliente.

3.1. Testes reais

Os testes foram realizados de maneira a comunicar duas máquinas distintas, uma com o papel de servidor e outra como cliente. Para isso, foram utilizados computadores do Departamento de Ciência da Computação, pertencentes ao mesmo laboratório de uso geral, conectados na mesma rede em horário de baixo tráfego de dados.

Armazenar e compartilhar arquivos de teste acaba se tornando uma tarefa custosa e inviável ao longo do tempo. Dessa forma, um script foi criado para gerar arquivos com tamanho inteiro de $[1, 10]$ megabytes; a Figura 3 mostra a linearidade sequencial.

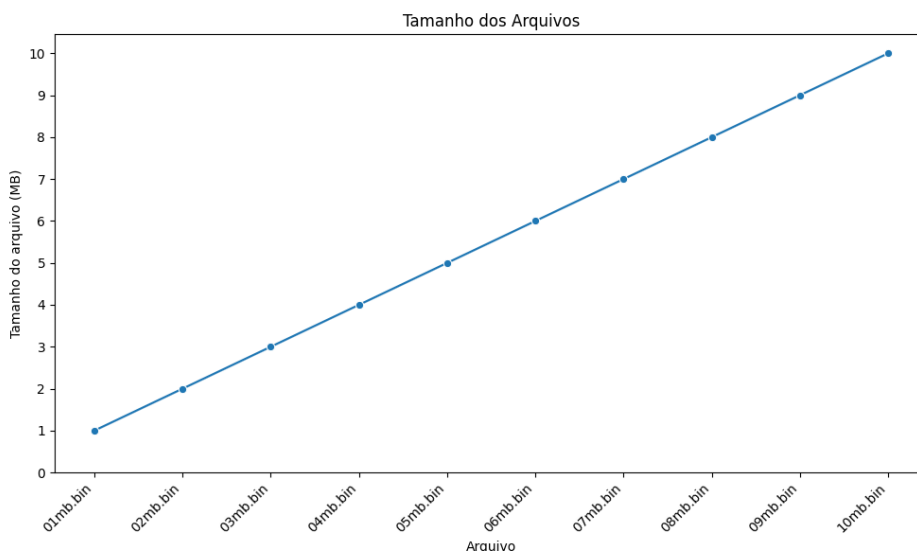


Figura 3. Gráfico do tamanho dos arquivos de teste

Os resultados apresentados foram obtidos a partir de uma média de 30 execuções realizadas para cada arquivo de tamanho constante, com o objetivo de assegurar a consistência e confiabilidade nos dados coletados. Durante os testes, foi analisada a relação entre as variáveis da transferência e o total de bytes.

3.2. Resultados dos Testes com TCP

Os resultados dos testes realizados com o protocolo TCP medindo a média do tempo de transferência e velocidade alcançada para arquivos de 1 a 10 megabytes estão sumarizados na Tabela 1.

Tabela 1. Resultados de desempenho para TCP

Arquivo	Tempo (s)	Velocidade (bytes/s)
1MB	0.013791	105,033,355
2MB	0.018916	109,866,568
3MB	0.028300	111,156,466
4MB	0.037305	111,432,757
5MB	0.046406	112,978,494
6MB	0.055591	113,246,705
7MB	0.064756	111,349,217
8MB	0.073905	111,535,460
9MB	0.083063	112,907,935
10MB	0.092305	113,470,324

Com base nos resultados obtidos, podemos traçar um gráfico, exibido na Figura 4. Este gráfico evidencia uma relação direta e linear entre o tamanho do arquivo e o tempo decorrido, com variações muito baixas.

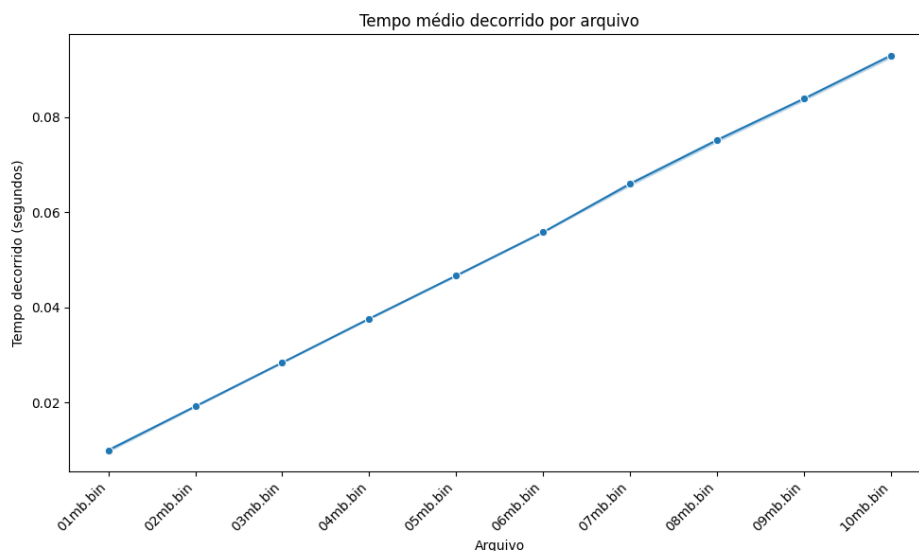


Figura 4. Gráfico de tempo decorrido - TCP

Por outro lado, o gráfico da Figura 5, retratando a velocidade, apresenta mais variação dos resultados, principalmente para arquivos menores.

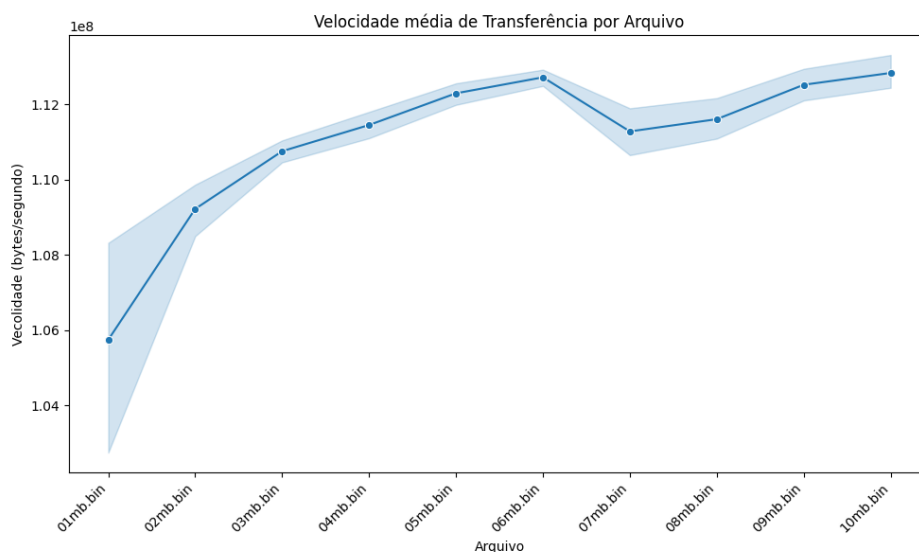


Figura 5. Gráfico de velocidade registrada - TCP

Os resultados mostram que, à medida que o tamanho do arquivo aumenta, a velocidade de transferência permanece consistente. Isso demonstra a eficiência do protocolo TCP para transferência de arquivos grandes em redes controladas. Pequenas variações na velocidade observadas estão dentro da margem esperada para o ambiente de teste.

Embora os tempos de transferência aumentem proporcionalmente com o tamanho do arquivo, as variações na velocidade entre esses testes são mínimas. Essas pequenas variações podem ser atribuídas a fatores como gerenciamento de buffers, latência da rede ou características específicas do hardware utilizado.

3.2.1. Verificação de integridade dos arquivos

Apesar do próprio protocolo TCP garantir a transferência confiável dos dados, diversos problemas podem ocorrer no ato de envio e recebimento dos pacotes para a remontagem do arquivo original no cliente, sendo assim, é necessário ter certeza de que esse processo esteja ocorrendo sem falhas.

Um método muito comum para isso é a utilização de funções de dispersão criptográfica, popularmente conhecidas como funções de hash. A característica desse método é a geração de um código específico para o conteúdo a ser lido. Existem vários padrões de geração disponíveis, sendo escolhido o MD5 para este trabalho.

Os arquivos de hash (.md5) são gerados pelo servidor no ato de criação dos arquivos de dados, basta que o cliente os solicite para realizar a checagem por conta própria pelo terminal. Esse processo foi muito benéfico durante a etapa de desenvolvimento, evitando verificações manuais.

3.3. Resultados dos Testes com UDP

Os testes realizados com o protocolo UDP mediram o tempo de transferência, a velocidade alcançada e a taxa de perda de pacotes. Os resultados para arquivos de 1 a 10 megabytes estão sumarizados na Tabela 2.

Tabela 2. Resultados de desempenho para UDP

Arquivo	Tempo (s)	Velocidade (bytes/s)	Perda de Pacotes (%)
1MB	0.010416	105,668,901	0.00
2MB	0.017521	117,240,552	0.90
3MB	0.025765	118,081,253	2.60
4MB	0.034122	118,940,471	3.40
5MB	0.042563	117,923,668	3.00
6MB	0.051012	117,612,584	3.20
7MB	0.059472	117,492,140	3.10
8MB	0.067952	116,083,906	1.70
9MB	0.076446	117,463,342	2.50
10MB	0.084961	117,466,715	1.75

Com base no gráfico da Figura 6 apresentado, que ilustra o tempo médio decorrido para transferências utilizando o protocolo UDP, podemos observar uma relação linear e crescente entre o tamanho do arquivo e o tempo de transferência, com pequenas variações.

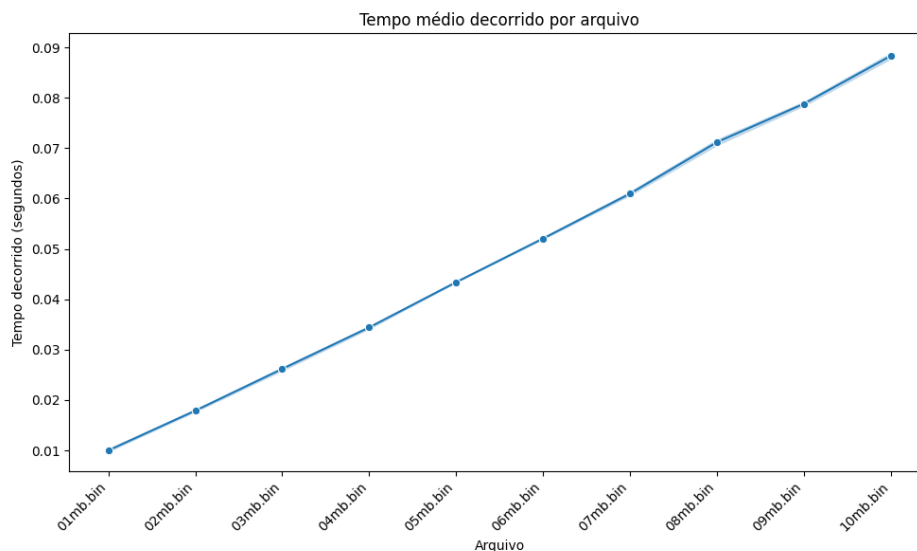


Figura 6. Gráfico de tempo decorrido - UDP

Analisando os resultados do gráfico da Figura 7 podemos analisar que o protocolo UDP se manteve fiel ao que ele promete oferecer: alta velocidade e eficiência em transmissões em cenários onde a confiabilidade não é garantida.

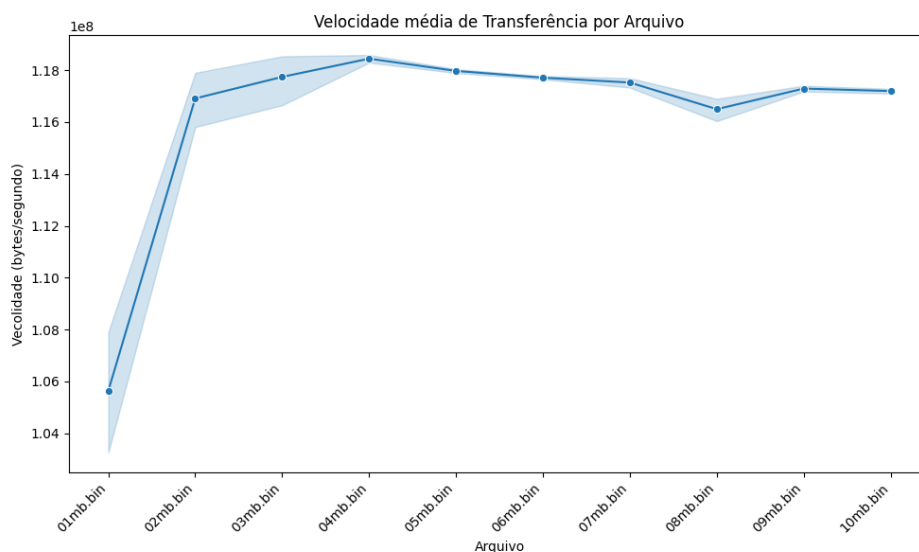


Figura 7. Gráfico de tempo decorrido - UDP

Com base na figura do gráfico 8, observa-se que a perda média de pacotes para o protocolo UDP varia conforme o tamanho do arquivo transmitido.

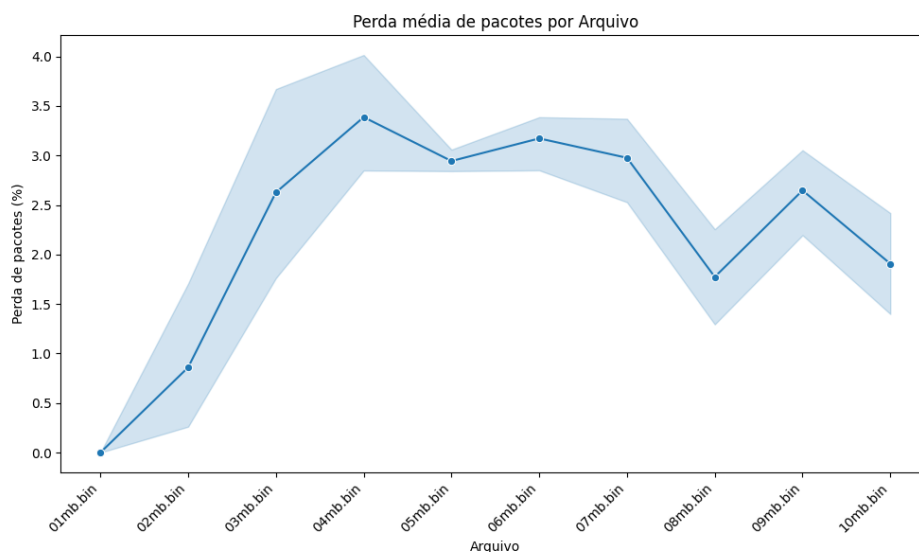


Figura 8. Gráfico da perda de pacotes - UDP

Inicialmente, para o arquivo de 1MB, a perda é insignificante, confirmando que em transmissões de menor porte, o UDP se mantém extremamente eficiente. À medida que o tamanho dos arquivos aumenta, a taxa cresce, atingindo um pico de 3,4%.

Após esse ponto, o comportamento oscila, com pequenas reduções e aumentos na taxa de perda dos pacotes. O gráfico evidencia essas variações com uma curva que reflete a instabilidade natural do UDP em ambientes com aumento de carga e tráfego. Essa oscilação é característica de um protocolo que não realiza controle de congestionamento ou retransmissão.

4. Conclusão

Neste projeto houve a implementação de um sistema de transferência de arquivos baseado nos protocolos TCP e UDP, destacando-se pela modularidade, eficiência e simplicidade no uso. A análise dos resultados experimentais permitiu compreender os comportamentos característicos de cada protocolo em diferentes cenários, oferecendo estudos valiosos sobre suas aplicações práticas.

Os testes realizados evidenciaram que o protocolo TCP é confiável, mantendo uma taxa de transferência constante de aproximadamente 111MB/s, mesmo com o aumento do tamanho dos arquivos. Essa estabilidade é acompanhada por uma entrega garantida dos dados, o que o torna ideal para cenários que demandam integridade e segurança na transmissão, como a transferência de grandes arquivos ou aplicações sensíveis a falhas. Além disso, a utilização de mecanismos de verificação de integridade, como o hash MD5, reforça a confiabilidade do sistema, assegurando que os arquivos transferidos permaneçam intactos.

Por outro lado, o protocolo UDP destacou-se pela eficiência e velocidade, apresentando tempos de transferência menores em comparação ao TCP. Apesar de sua natureza sem conexão e da ausência de mecanismos de controle de erros, o UDP demonstrou ser uma escolha viável para transmissões que priorizam desempenho, como streaming de dados em tempo real. Contudo, foi observado que a perda de pacotes aumenta com o tamanho dos arquivos, atingindo um pico de 3,4% para arquivos de 4MB, seguido de oscilações. Essa característica é uma limitação do protocolo, que deve ser considerada ao escolher seu uso em contextos onde a confiabilidade é algo crucial.

Por fim, o sistema desenvolvido não apenas fornece uma base sólida para estudos comparativos entre os dois protocolos, mas também pode ser aplicado em ambientes reais, como servidores de armazenamento distribuído e transmissões de dados em tempo real. A estrutura modular e as boas práticas de programação em C garantiram a escalabilidade e a adaptabilidade do sistema, permitindo sua evolução para atender à demanda. Este trabalho reforça a importância de compreender as características dos protocolos de transporte para projetar soluções de comunicação eficientes e alinhadas às necessidades de cada cenário.

Referências

- [1] Tanenbaum, Andrew S., and Wetherall, David J. (2010). *Redes de Computadores*. 5ª Edição. Pearson.
- [2] Kurose, James F., and Ross, Keith W. (2010). *Redes de Computadores e a Internet: Uma Abordagem Top-Down*. 5ª Edição. Pearson.