



Universidade Federal
de São João del-Rei

Redes de Computadores

Implementação e Comparação de Sockets TCP e UDP

Paulo Eduardo Campos Junior Rafael Perini Giacomini
Ramon Coelho Júnior

Novembro de 2024

1 Introdução

O objetivo deste trabalho é implementar e comparar servidores e clientes TCP e UDP para a transferência de arquivos, com foco na análise da taxa de download e da perda de pacotes. Para os testes, utilizamos um arquivo de 10 MB como base de dados.

2 Implementação

A seguir, detalhamos as implementações dos servidores e clientes para os protocolos TCP e UDP, destacando suas principais características e funcionalidades.

2.1 Servidor e Cliente TCP

O servidor TCP, que utiliza o protocolo orientado à conexão, garante a entrega confiável do arquivo binário em blocos de 4 KB. O cliente TCP se conecta ao servidor, realiza o download do arquivo e calcula a taxa de transferência.

```
1 void send_file(FILE *file, int client_socket) {
2     char buffer[BUFFER_SIZE];
3     size_t bytes_read;
4
5     while ((bytes_read = fread(buffer, 1, BUFFER_SIZE, file))
6           > 0) {
7         if (send(client_socket, buffer, bytes_read, 0) == -1)
8             {
9                 perror("Erro ao enviar dados");
10                exit(EXIT_FAILURE);
11            }
12    }
```

Listing 1: Trecho do servidor TCP: Envio de arquivo

No cliente TCP, a taxa de download é calculada medindo o tempo total da transferência, conforme mostrado abaixo:

```
1 start = clock();
2 while ((bytes_received = recv(client_fd, buffer, BUFFER_SIZE,
3                               0)) > 0) {
4     fwrite(buffer, 1, bytes_received, file);
5 }
6 end = clock();
7
8 time_spent = (double)(end - start) / CLOCKS_PER_SEC;
9 printf("Taxa de download: %.2f bytes/segundo\n", 10 * 1024 *
10        1024 / time_spent);
```

Listing 2: Cálculo da taxa de download no cliente TCP

2.2 Servidor e Cliente UDP

O servidor UDP transmite o arquivo em pacotes de 4 KB e utiliza um pacote especial com a mensagem "END" para sinalizar o fim da transmissão. No cliente, cada pacote recebido é gravado em disco e a taxa de download é calculada da mesma forma que no cliente TCP.

Para contabilizar as perdas de pacotes, o cliente UDP compara os pacotes esperados com os pacotes recebidos:

```
1  expected_packets = (10 * 1024 * 1024 + BUFFER_SIZE - 1) /  
    BUFFER_SIZE;  
2  
3  printf("Pacotes esperados: %d, Pacotes recebidos: %d\n",  
4        expected_packets, received_packets);  
5  printf("Porcentagem de perda de pacotes: %.2f%%\n",  
6        100.0 * (expected_packets - received_packets) /  
            expected_packets);
```

Listing 3: Cálculo de perda de pacotes no cliente UDP

O servidor UDP utiliza o seguinte método para enviar pacotes ao cliente:

```
1  void send_file(FILE *file, int server_socket, struct  
    sockaddr_in *client_addr, socklen_t addr_len) {  
2      char buffer[BUFFER_SIZE];  
3      size_t bytes_read;  
4  
5      while ((bytes_read = fread(buffer, 1, BUFFER_SIZE, file))  
6              > 0) {  
7          if (sendto(server_socket, buffer, bytes_read, 0,  
8                    (struct sockaddr *)client_addr, addr_len)  
9                  == -1) {  
10             perror("Erro ao enviar pacote");  
11             exit(EXIT_FAILURE);  
12         }  
13     }  
14  
15     strcpy(buffer, "END");  
16     sendto(server_socket, buffer, strlen(buffer), 0,  
            (struct sockaddr *)client_addr, addr_len);  
}
```

Listing 4: Envio de pacotes no servidor UDP

3 Resultados

Os testes foram realizados com um arquivo de 10 MB, obtendo os seguintes resultados:

- **Taxa de Download média:**
 - TCP: 8123,91 Mb/s.
 - UDP: 13276,43 Mb/s.
- **Perda de Pacotes (UDP): 45,02%.**

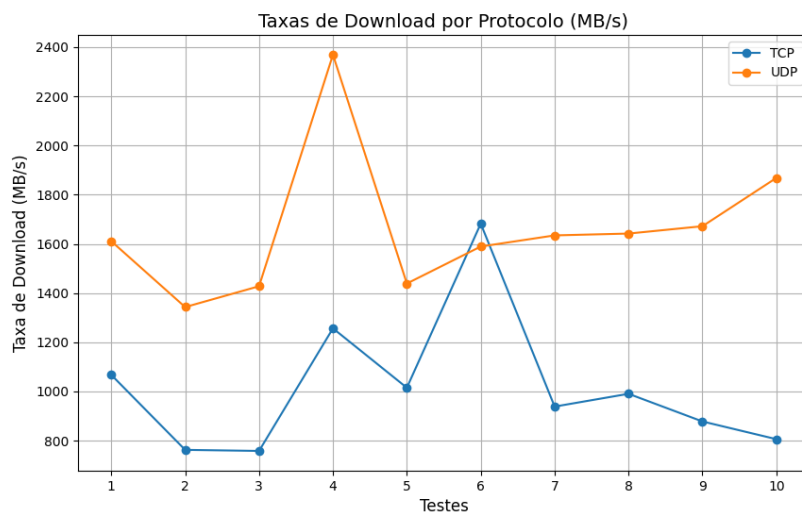


Figura 1: Comparação da Taxa de Download entre TCP e UDP.

4 Discussão

A comparação entre TCP e UDP revelou as seguintes conclusões:

- **TCP:** Apresenta uma menor taxa de download devido à sobrecarga dos mecanismos de confiabilidade, mas garante a entrega completa do arquivo, sem perdas.
- **UDP:** Apresentou maior taxa de download, porém com perdas de pacotes, visto que o protocolo não realiza retransmissões. A ausência de controle de fluxo afeta a integridade dos dados em condições adversas de rede.

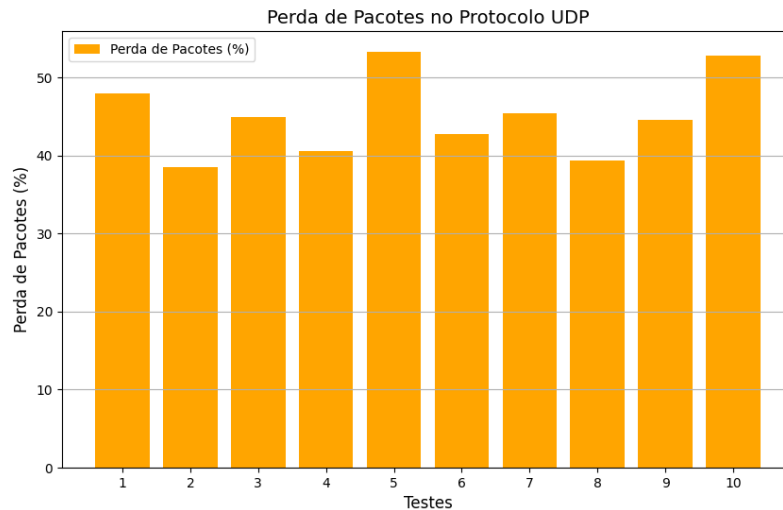


Figura 2: Perda de Pacotes no Protocolo UDP.

5 Conclusão

A implementação e os testes realizados evidenciam as principais características dos protocolos TCP e UDP e suas aplicações ideais. O TCP, devido à sua confiabilidade, apresentou uma taxa de download menor (8123,91 Mb/s em média), mas assegurou a integridade dos dados, sendo essencial em aplicações que exigem alta confiabilidade, como transferências de arquivos, transações financeiras e comunicações críticas.

O UDP, por operar sem conexão e sem retransmissão de pacotes, teve uma taxa de download significativamente maior (13276,43 Mb/s em média), tornando-se ideal para aplicações onde a velocidade é mais importante que a integridade absoluta dos dados, como streaming de vídeo, transmissões ao vivo e jogos online. Contudo, a perda de pacotes observada (45,02% em média) sugere que, em redes congestionadas ou instáveis, a performance pode ser comprometida.

A escolha entre TCP e UDP depende diretamente das necessidades da aplicação em termos de confiabilidade e desempenho. Trabalhos futuros poderiam explorar estratégias para reduzir as perdas no UDP, como a implementação de camadas adicionais de controle na aplicação, além de investigar o impacto de diferentes condições de rede na performance dos protocolos.