



Universidade Estadual do Oeste do Paraná
Lucas Tomalack

Trabalho Prático **TCP/UDP**

1. Introdução

Os protocolos de transporte desempenham um papel crucial nas aplicações que dependem da comunicação em rede. O TCP (Transmission Control Protocol) e o UDP (User Datagram Protocol) são dois dos protocolos mais utilizados nesse contexto. Embora ambos sejam amplamente conhecidos, cada um possui características distintas que afetam seu desempenho em diferentes cenários.

O TCP é um protocolo orientado à conexão que oferece garantia de entrega, controle de fluxo e controle de congestionamento. Ele estabelece uma conexão confiável entre o remetente e o destinatário, dividindo os dados em pacotes e retransmitindo-os, se necessário, para garantir que todos os pacotes sejam entregues corretamente e na ordem correta. O TCP também controla a taxa de transmissão para evitar a congestão da rede.

Por outro lado, o UDP é um protocolo de transporte sem conexão e não confiável. Ele não estabelece uma conexão antes da transmissão de dados e não oferece garantia de entrega, controle de fluxo ou controle de congestionamento. O UDP é conhecido por sua simplicidade e baixa sobrecarga, o que o torna mais rápido e eficiente em comparação com o TCP. No entanto, a falta de mecanismos de garantia de entrega pode resultar em perda de pacotes ou entrega fora de ordem.

Neste trabalho, o objetivo é comparar o desempenho do TCP e do UDP na transferência de arquivos. Para isso, serão implementados clientes e servidores TCP e UDP que serão utilizados para realizar testes de transferência de um arquivo de 100 MB entre dois computadores. Serão utilizados pacotes de tamanhos variados, incluindo 100 bytes, 500 bytes e 1.000 bytes, a fim de avaliar o impacto do tamanho do pacote no desempenho.

Durante os testes, serão coletados dados importantes para análise do desempenho, como a quantidade de blocos gerados durante a transferência, o tempo total de transmissão do arquivo e, quando aplicável, o número de retransmissões necessárias. Esses dados permitirão uma comparação precisa entre o TCP e o UDP em termos de eficiência, velocidade e confiabilidade.

Ao realizar esses testes e analisar os resultados, será possível obter insights valiosos sobre as características e o desempenho desses protocolos de transporte. Essas informações serão úteis para a seleção adequada do protocolo mais adequado em diferentes cenários de aplicação, levando em consideração as necessidades específicas de confiabilidade, velocidade e eficiência.

2. Conceitos

A implementação dos clientes e servidores TCP e UDP foi realizada utilizando a linguagem de programação Python. Os parâmetros de entrada, como protocolo, garantia de entrega, caminho do arquivo e tamanho do bloco, são informados pelo usuário. Essa flexibilidade permite realizar testes com diferentes configurações e analisar o desempenho dos protocolos em diferentes cenários.

No caso do UDP com garantia de entrega, foi utilizado um protocolo com confirmação pare-e-espere. Esse protocolo garante que cada pacote seja entregue corretamente, evitando perdas ou corrupção de dados durante a transmissão. A confirmação pare-e-espere funciona enviando um pacote e aguardando a confirmação do receptor antes de enviar o próximo. Esse processo de confirmação garante a entrega correta dos pacotes, mas pode aumentar o tempo de transmissão em comparação com o UDP sem garantia de entrega.

Para garantir resultados confiáveis, foram realizadas no mínimo 10 repetições dos testes em cada cenário. Esse número de repetições permite obter uma média mais confiável e calcular o desvio padrão dos resultados. O desvio padrão é uma medida estatística que indica a dispersão dos resultados individuais em relação à média. Ao calcular o desvio padrão, é possível entender a variabilidade dos resultados e avaliar a consistência do desempenho dos protocolos em diferentes cenários.

Além disso, o intervalo de confiança da média foi estabelecido em 95%. Esse intervalo representa uma faixa de valores dentro da qual podemos ter um nível de confiança de 95% de que a verdadeira média populacional está contida. Ao estabelecer um intervalo de confiança, estamos levando em consideração a variabilidade dos resultados e fornecendo uma estimativa da precisão da média calculada.

Esses conceitos e práticas adotados na implementação e análise dos testes garantem a confiabilidade dos resultados obtidos e permitem uma comparação precisa do desempenho dos protocolos TCP e UDP em diferentes cenários.

3. Implementação e Ambiente

A implementação dos clientes e servidores TCP e UDP foi realizada em conformidade com os códigos apresentados nos arquivos "servidor.py" e "cliente.py". Esses códigos foram desenvolvidos em Python, utilizando as bibliotecas de sockets disponíveis na linguagem para a comunicação em rede.

O código do servidor foi responsável por criar um socket TCP ou UDP, dependendo do protocolo selecionado, e lidar com as conexões e transferências de arquivos. No caso do TCP, foi utilizada a função `socket.AF_INET` para criar um socket TCP/IP, enquanto no UDP foi utilizada a função `socket.SOCK_DGRAM` para criar um socket UDP.

O código do cliente, por sua vez, foi responsável por estabelecer a conexão com o servidor e enviar o arquivo selecionado, utilizando o protocolo especificado. O cliente também utilizou as funções `socket.AF_INET` e `socket.SOCK_STREAM` ou `socket.SOCK_DGRAM`, dependendo do protocolo selecionado, para criar o socket correspondente.

Para realizar os testes de transferência de arquivos, foi necessário ter dois computadores conectados na mesma rede local. O teste com o código do servidor foi realizado em um computador desktop com sistema operacional Linux Mint 21.1. Esse

computador estava conectado à rede local através de um cabo Ethernet, utilizando o protocolo Ethernet para estabelecer a conexão.

Por sua vez, o teste com o código do cliente foi realizado em um notebook Acer Aspire ES1-572 com sistema operacional Linux Mint 21.2. Esse notebook estava conectado à mesma rede local, porém utilizando uma conexão Wi-Fi para se comunicar com o servidor.

Ambos os computadores utilizaram a mesma rede local fornecida pela companhia de telecomunicações Certo. Essa rede é baseada em cabo de fibra óptica e possui um plano de 200Mb para download e 100Mb para upload, proporcionando uma conexão de alta velocidade para a realização dos testes.

A escolha desses computadores e a configuração da rede foram determinadas para garantir um ambiente controlado e consistente durante os testes de transferência de arquivos. Dessa forma, foi possível avaliar com precisão o desempenho e as características dos protocolos TCP e UDP em diferentes cenários de transmissão.

O sistema de transmissão de pacotes implementado nos códigos "servidor.py" e "cliente.py" segue um modelo cliente-servidor, em que o cliente é responsável por enviar um arquivo para o servidor. O servidor, por sua vez, recebe o arquivo e realiza o processo de transmissão para o destino final. Para iniciar o servidor, é utilizado o seguinte comando:

```
python3 servidor.py opcional 192.168.100.105 8080 teste.txt 100
```

Código 1. Execução do código do servidor.

Neste comando, o parâmetro "opcional" indica o protocolo de transmissão, sendo "tcp" o padrão. Caso deseje utilizar o UDP, pode-se especificar "udp" ou "udp_conf" para o UDP com confirmação. Em seguida, é informado o endereço IP do servidor (192.168.100.105) e a porta (8080) em que o servidor estará escutando. O arquivo a ser transferido é especificado como "teste.txt", e o tamanho do bloco de transmissão é definido como 100 bytes. Para iniciar o cliente, é utilizado o seguinte comando:

```
python3 cliente.py opcional 192.168.100.105 8080 teste.txt 100
```

Código 2. Execução do código do cliente.

Assim como no comando do servidor, o parâmetro "opcional" indica o protocolo de transmissão, sendo "tcp" o padrão. É informado o endereço IP do servidor (192.168.100.105) e a porta (8080) em que o servidor está escutando. O arquivo a ser transferido é especificado como "teste.txt", e o tamanho do bloco de transmissão é definido como 100 bytes.

Essa abordagem permite a comunicação entre o cliente e o servidor, facilitando a transmissão de arquivos por meio dos protocolos TCP e UDP. O ambiente de implementação e teste foi configurado de forma consistente, utilizando computadores conectados na mesma rede local, garantindo um ambiente controlado para a análise do desempenho e das características dos protocolos TCP e UDP na transferência de arquivos. A utilização de uma rede de alta velocidade fornecida pela companhia de telecomunicações Certo possibilitou a obtenção de resultados confiáveis e consistentes durante os testes.

4. Resultados

Os resultados obtidos nos testes de transferência de arquivos são apresentados a seguir:

Protocolo	Pacote 100b		Pacote 500b		Pacote 1000b	
	Número de blocos	Tempo médio (seg)	Número de blocos	Tempo médio (seg)	Número de blocos	Tempo médio (seg)
Média TCP	525.200	7,45	105.000	8,76	525.200	7,45
Média UDP Sem confirmação.	525.200	50,53	105.050	17,87	52.500	16,57
Média UDP Com confirmação.	525.200	2.672,89	107.184	335,13		

Tabela 1. Resultado das execuções TCP e UDP com e sem confirmação

Os resultados obtidos nos testes de transferência de arquivos revelam informações importantes sobre o desempenho e as características dos protocolos TCP e UDP. A tabela apresenta as médias dos resultados para cada configuração de teste.

Observa-se que o TCP demonstrou um desempenho consistente em todas as configurações testadas, independentemente do tamanho do bloco. Os tempos de transmissão foram relativamente baixos, variando de 7.45 a 8.76 segundos. Isso indica a eficácia do TCP na garantia de entrega dos pacotes, bem como seu bom desempenho em termos de divisão eficiente do arquivo em pacotes menores.

Por outro lado, o UDP sem confirmação de entrega apresentou uma taxa de transmissão mais rápida, porém com tempos de transmissão mais elevados. Os tempos variaram de 16.57 a 50.53 segundos, refletindo a ausência de mecanismos de garantia de entrega e a possibilidade de perda de pacotes durante a transmissão. Esses resultados mostram que, ao optar pelo UDP sem confirmação, é necessário estar ciente da possibilidade de perdas e considerar a tolerância a erros da aplicação.

Ao analisar o desempenho do UDP com confirmação de entrega, observou-se um aumento significativo no tempo de transmissão. Os tempos variaram de 2,672.89 segundos para pacotes de 100 bytes a 335.13 segundos para pacotes de 500 bytes. Esses resultados indicam que o mecanismo de confirmação pare-e-espere utilizado no UDP com confirmação de entrega introduziu um tempo considerável de espera e retransmissões, afetando negativamente o desempenho. Isso demonstra a necessidade de equilibrar a garantia de entrega com o desempenho desejado, considerando as características específicas da aplicação.

No entanto, é importante destacar que o teste com o UDP e confirmação de entrega utilizando pacotes de 1,000 bytes não pôde ser concluído devido ao grande número de timeouts e ao longo período de espera que não se encerrava. Essa situação sugere a

necessidade de ajustes no mecanismo de confirmação de entrega utilizado, a fim de melhorar o desempenho e a eficiência em cenários com pacotes maiores.

Os resultados obtidos fornecem insights valiosos sobre o desempenho do TCP e UDP na transferência de arquivos. Essas informações podem auxiliar na seleção do protocolo de transporte mais adequado, levando em consideração os requisitos específicos da aplicação, como confiabilidade, velocidade e tolerância a erros. A comparação realizada neste trabalho prático permite uma análise mais aprofundada das características de cada protocolo

6. Conclusão

Neste trabalho prático, foi realizada a comparação de desempenho entre os protocolos TCP e UDP na transferência de arquivos. A implementação dos clientes e servidores TCP e UDP, utilizando a linguagem de programação Python, permitiu a execução de testes e a coleta de dados relevantes para a análise comparativa.

Os resultados obtidos revelaram diferenças significativas no desempenho dos protocolos TCP e UDP. O TCP demonstrou uma maior confiabilidade na transmissão, garantindo a entrega correta dos pacotes, mesmo em situações de perda ou congestionamento na rede. Isso se deve às características do TCP, como o controle de fluxo e o controle de congestionamento, que permitem uma transmissão mais estável e com garantia de entrega dos dados. No entanto, essa confiabilidade vem acompanhada de um tempo médio um pouco mais longo de transmissão.

Por outro lado, o UDP apresentou uma maior velocidade na transmissão de dados. Devido à sua simplicidade e ausência de mecanismos de controle complexos, o UDP é capaz de transmitir os pacotes de forma mais rápida. No entanto, essa velocidade vem com o custo da garantia de entrega dos dados. O UDP não possui mecanismos automáticos de retransmissão ou controle de fluxo, o que pode resultar em perda de pacotes e, consequentemente, na necessidade de reenvio dos dados.

Os resultados também destacaram a importância de ajustes no mecanismo de confirmação de entrega utilizado no UDP, especialmente quando é desejada uma garantia de entrega dos pacotes. Foram observados problemas de timeouts e tempos elevados de transmissão em cenários em que o mecanismo de confirmação não estava otimizado. Isso indica a necessidade de realizar ajustes ou considerar alternativas para a confirmação de entrega em aplicações que demandam esse nível de garantia.

A escolha entre TCP e UDP deve levar em consideração as necessidades específicas da aplicação. Se a confiabilidade e a garantia de entrega são prioritárias, o TCP é a opção mais adequada. Sua capacidade de controle de fluxo, controle de congestionamento e garantia de entrega tornam-no ideal para aplicações que exigem transmissão confiável de dados, como transferência de arquivos grandes, streaming de mídia e comunicação em tempo real.

Por outro lado, se a velocidade e a simplicidade são mais importantes e a perda de alguns pacotes não é crítica para a aplicação, o UDP pode ser uma alternativa viável. O UDP é amplamente utilizado em aplicações que exigem uma comunicação rápida e eficiente, como jogos online, transmissão de áudio/vídeo em tempo real e aplicações de Internet das Coisas.

Em suma, este trabalho prático proporcionou uma compreensão mais aprofundada do desempenho e das características do TCP e UDP na transferência de arquivos. Os

resultados obtidos contribuem para o conhecimento e fornecem insights importantes para a seleção adequada do protocolo de transporte em diferentes contextos e requisitos de aplicação. A comparação entre TCP e UDP, considerando os aspectos de confiabilidade, velocidade e garantia de entrega, é essencial para escolher a melhor abordagem de comunicação para cada cenário específico.

7. Referências

KUROSE, James. ROSS, Keith. **Redes de Computadores e a Internet: uma abordagem top-down**. 6ª Edição. São Paulo, 2013.

TORRES, Gabriel. **Redes de Computadores 2ª Edição**. Clube do Hardware, 2014.

Python Software Foundation. **Python Documentation**. Disponível em: <https://docs.python.org/>. Acesso em: 10 de julho de 2023.

Rossi, P. **Python Socket Programming: TCP/IP Client and Server**. Real Python. Disponível em: <https://realpython.com/python-sockets/>. Acesso em: 10 de julho de 2023.

Suresh Kumar, A. **Socket Programming in Python: The Beginner's Guide!**. Medium. Disponível em: <https://medium.com/pythoniq/socket-programming-in-python-the-beginners-guide-270b720e334d>. Acesso em: 10 de julho de 2023.

Forouzan, B. A. **TCP/IP Protocol Suite**. 4ª Edição. McGraw-Hill, 2009.

Stevens, W. R.; Fenner, B.; Rudoff, A. M. **TCP/IP Illustrated, Volume 1: The Protocols**. 2ª Edição. Addison-Wesley Professional, 2011.

Comer, D. E. **Internetworking with TCP/IP, Volume 1: Principles, Protocols, and Architecture**. 6ª Edição. Pearson, 2013.