

TP-Cloud

Nome: Iago de Araújo Reis

Matrícula: 223119005

Introdução

Este trabalho foi realizado com o objetivo de desenvolver uma aplicação do tipo cliente-servidor utilizando a linguagem c e sockets. O projeto simula um sistema no modelo requisição-resposta, que atua com apoio tanto ao IPv4 quanto ao IPv6, visando avaliar conceitos de comunicação de redes e computação na nuvem.

Metodologia

O projeto foi implementado utilizando sockets e protocolo TCP com o intuito de funcionar como um sistema cliente-servidor simples, que visa principalmente medir o tempo de envio e recepção de mensagens de diferentes tamanhos. O objetivo final foi comparar o desempenho relacionado à comunicação entre os protocolos IPv4 e IPv6.

A realização dos testes ocorreram na mesma máquina, onde foi utilizado o sistema operacional Ubuntu 24.04.1 LTS. Para isso, foi necessário instalar o Ubuntu em uma máquina que rodava no windows. Através da ferramenta WSL, que permite rodar distribuições Linux diretamente no windows, o código foi compilado e executado. Isso foi necessário porque os códigos usavam funções e bibliotecas que são nativas do Linux.

Tanto o cliente quanto o servidor se concentraram na mesma máquina, para isso foi utilizado o endereço de loopback, sendo o 127.0.0.1 para o IPv4 e ::1 para o IPv6, a comunicação aconteceu na máquina, simulando uma rede real, mas de forma totalmente local e controlada, permitindo resultados mais estáveis e previsíveis.

Durante os testes, foram enviados pacotes com tamanhos diferentes que variam exponencialmente desde 2^0 até 2^{16} , foram realizados seis testes, sendo três para IPv4 e três para o IPv6, visando garantir a confiabilidade nos resultados. Logo após, foram calculados os tempos médios de cada tamanho, assim como o desvio padrão e a banda média.

A análise foi demonstrada de forma visual através de gráficos, relacionando o tamanho da mensagem com o tempo médio, permitindo uma melhor comparação entre o desempenho do IPv4 e IPv6.

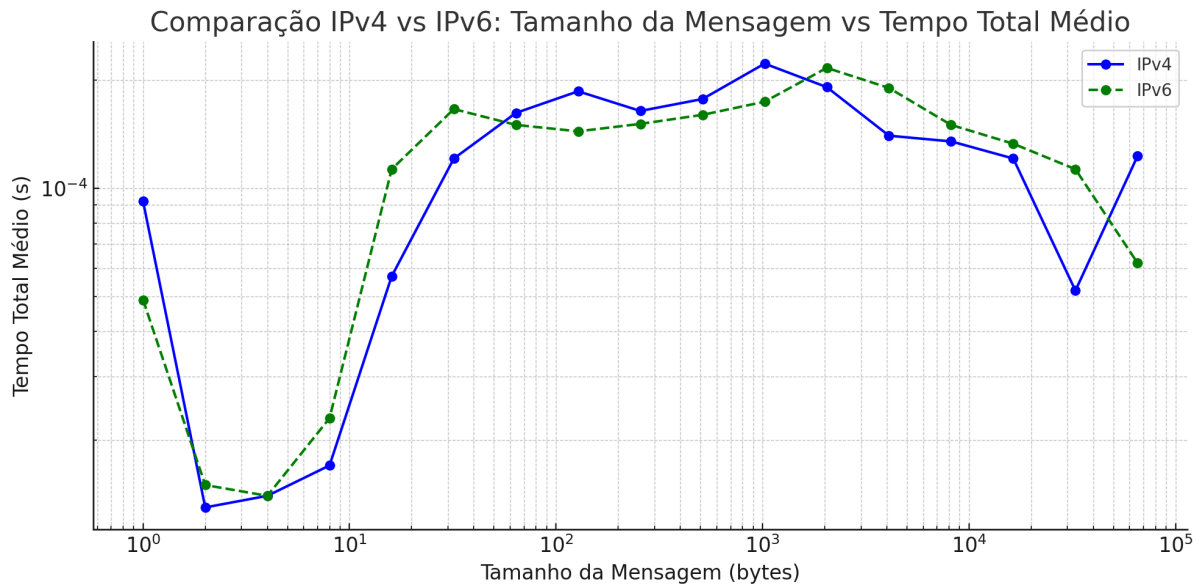
Resultados:

IPv4:

Tamanho de mensagem	Nº de mensagens enviadas	Tempo total médio	Desvio padrão dos tempos medidos	Banda média	
1	3	0.000092	0.000086	10830.32	
2	3	0.000013	0.000002	153846.15	
4	3	0.000014	0.000003	292682.93	
8	3	0.000017	0.000001	461538.46	
16	3	0.000057	0.000053	279069.77	
32	3	0.000121	0.000024	264462.81	
64	3	0.000162	0.000073	395876.29	
128	3	0.000186	0.000021	689407.54	
256	3	0.000164	0.000020	1557809.33	
512	3	0.000177	0.000004	2887218.05	
1024	3	0.000222	0.000099	4619548.87	
2048	3	0.000191	0.000001	10722513.09	
4096	3	0.000140	0.000030	29257142.86	
8192	3	0.000135	0.000015	60831683.17	
16384	3	0.000121	0.000009	135032967.03	
32768	3	0.000052	0.000002	634219354.84	
65536	3	0.000123	0.000070	531372972.97	

IPv6:

Tamanho de mensagem	Nº de mensagens enviadas	Tempo total médio	Desvio padrão dos tempos medidos	Banda média	
1	3	0.000049	0.000004	20.24	
2	3	0.000015	0.000001	133.33	
4	3	0.000014	0.000003	285.71	
8	3	0.000023	0.000005	347.83	
16	3	0.000113	0.000062	141.59	
32	3	0.000166	0.000072	192.77	
64	3	0.000150	0.000057	426.67	
128	3	0.000144	0.000050	888.89	
256	3	0.000151	0.000067	1695.26	
512	3	0.000160	0.000037	3200.00	
1024	3	0.000174	0.000046	5885.06	
2048	3	0.000216	0.000065	9481.48	
4096	3	0.000190	0.000059	21557.89	
8192	3	0.000150	0.000023	54613.33	
16384	3	0.000133	0.000010	123187.97	
32768	3	0.000113	0.000024	289885.49	
65536	3	0.000062	0.000007	1058000.00	



Análise e Discussão dos Resultados:

Após a realização dos testes, com os gráficos já gerados tanto para IPv4 quanto para IPv6, foi possível verificar que o tempo total médio de envio das mensagens não cresce de maneira linear ao decorrer que os tamanhos das mensagens são aumentados. Muito pelo contrário, os resultados demonstraram que ocorrem oscilações, principalmente para tamanhos pequenos, isso porque esses valores iniciais são mais influenciados por fatores como o overhead.

Com o decorrer do crescimento dos tamanhos das mensagens, os tempos médios começaram e se estabilizaram apresentando menos flutuação. Isso acontece porque o custo de envio começa a ser mais influenciado pelo volume real de dados do que pelo overhead fixo, tornando o tempo mais relacionado ao tamanho da mensagem.

Ao comparar o IPv4 e IPv6, foi percebido que os dois protocolos apresentam tempos bastantes próximos para os diferentes tamanhos de mensagens. Isso mostra que em ambiente local, como o projeto que foi usado o loopback, não há grande diferença entre IPv4 e IPv6, as pequenas variações percebidas podem ter sido influenciadas por alguns fatores como as pilhas de rede do sistema operacional, mas é algo natural. No geral, o desempenho foi bem semelhante.

Além disso, no eixo X(tamanho da mensagem) e no eixo Y(tempo médio) foram utilizados escalas logarítmicas para facilitar a visualização das tendências, permitindo ver os picos e quedas de maneira mais clara pois todos os valores passam a possuir representação visual mais equilibrada. As flutuações, mesmo em ambiente controlado, reforçaram a importância de realizar múltiplas medições e usar valores médios para representar os dados com mais precisão, por isso, foram realizadas 3 repetições por protocolo, depois calculado a média e o desvio padrão, para ter uma visão mais precisa do desempenho.

Conclusão:

Depois de realizar esse projeto, percebi o quanto é importante entender como funciona a comunicação entre cliente e servidor, e como essa comunicação é mais detalhada do que eu imaginava. No começo, eu achava que se tratava apenas de um “envia e recebe” super simples, mas o processo de criação do projeto, desde o código até a parte da interpretação dos resultados dos testes, me mostrou o quanto tem coisa acontecendo por trás. Além disso, aprendi a trabalhar com sockets, a usar o `gettimeofday()` para medir tempo, além de entender melhor sobre protocolos como IPv4 e IPv6, algo que eu nunca tive contato direto antes.

Outro ponto importante foi entender as oscilações do sistema, e que mesmo em testes no localhost, sem rede externa, o desempenho tende a variar. Isso me fez refletir sobre o quanto os sistemas reais precisam considerar para realizar seu funcionamento de forma correta.

No fim, posso dizer que esse projeto me fez entender de maneira mais clara conceitos que antes me pareciam abstratos.