

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
REDES DE COMPUTADORES

Autor: Janderson Barbeito da Silva

Trabalho Prático 2

BELO HORIZONTE, 2018

1. Introdução

Esse trabalho tem por objetivo implementar dois programas, um cliente e um servidor, que realizem comunicação através de *sockets* UDP para troca de dados.

O programa cliente enviará para o servidor o nome do arquivo que ele deseja receber os dados e guardará o tempo em que a mensagem foi enviada. O servidor, por sua vez, realizará a leitura desses dados e fará o envio deles para o cliente através de um *buffer* de tamanho variável que será informado para ambos os programas no momento da execução de cada um deles. Após receber todos os dados do servidor, o cliente guardará novamente o tempo, para poder obter o tempo transcorrido para a recepção de todos os dados vindos do servidor.

Pelo fato desse trabalho ter sido implementado sobre *sockets* UDP que não possuem entrega confiável de pacotes foi necessário implementar um comunicação *stop-an-wait* entre cliente e servidor, onde o servidor enviava o pacote contendo os dados e um cabeçalho com o número sequencial do pacote e o cliente enviava de volta o ACK com o número da sequência do pacote recebido. O tempo de espera para a troca de pacotes foi definido como de 1 segundo, caso não houvesse recebimento nesse tempo, havia o evento de *timeout* e o pacote era retransmitido.

2. Metodologia

Os testes dos programas criados foram feitos em uma máquina virtual com sistema Linux 16.04 LTS. Os programas foram rodados sequencialmente para os tamanhos de *buffer* variando de 128 a 8192, sendo realizadas 5 execuções em cada um dos cenários diferentes enviando um arquivo “.txt” de aproximadamente 3,3MB. As medições de tempo foram feitas através da obtenção de duas amostras de tempo, uma antes da execução da transferência do arquivo e uma depois no programa cliente. Logo após foi feita a subtração do tempo final e do tempo inicial.

Os programas cliente e servidor foram executados juntos num mesmo computador.

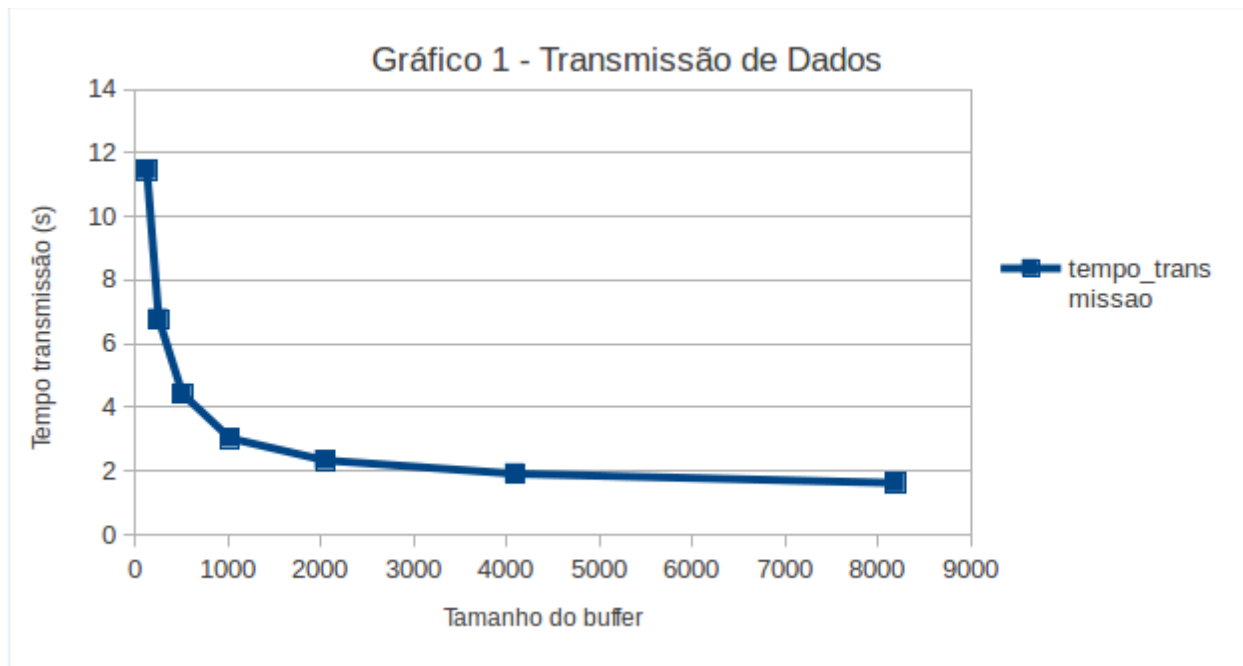
3. Resultados

Os dados foram coletados através de sucessivas execuções dos programas, onde o cliente fazia comunicação com o servidor.

Na tabela abaixo encontra-se os resultados dos tempos medidos em cada uma das execuções do teste de envio dos arquivos.

Tamanho do <i>buffer</i>	Medidas de tempo obtidas (em segundos)					Média	Desvio Padrão
128	11,346	11,277	11,576	11,276	11,357	11,456	0,189
256	6,671	6,968	6,804	6,603	6,842	6,778	0,144
512	4,342	4,352	4,450	4,480	4,519	4,429	0,078
1024	3,082	3,079	3,035	3,064	2,913	3,034	0,070
2048	2,439	2,340	2,352	2,205	2,362	2,339	0,084
4096	2,215	1,825	1,802	1,859	1,900	1,920	0,169
8192	1,672	1,571	1,582	1,669	1,665	1,632	0,050

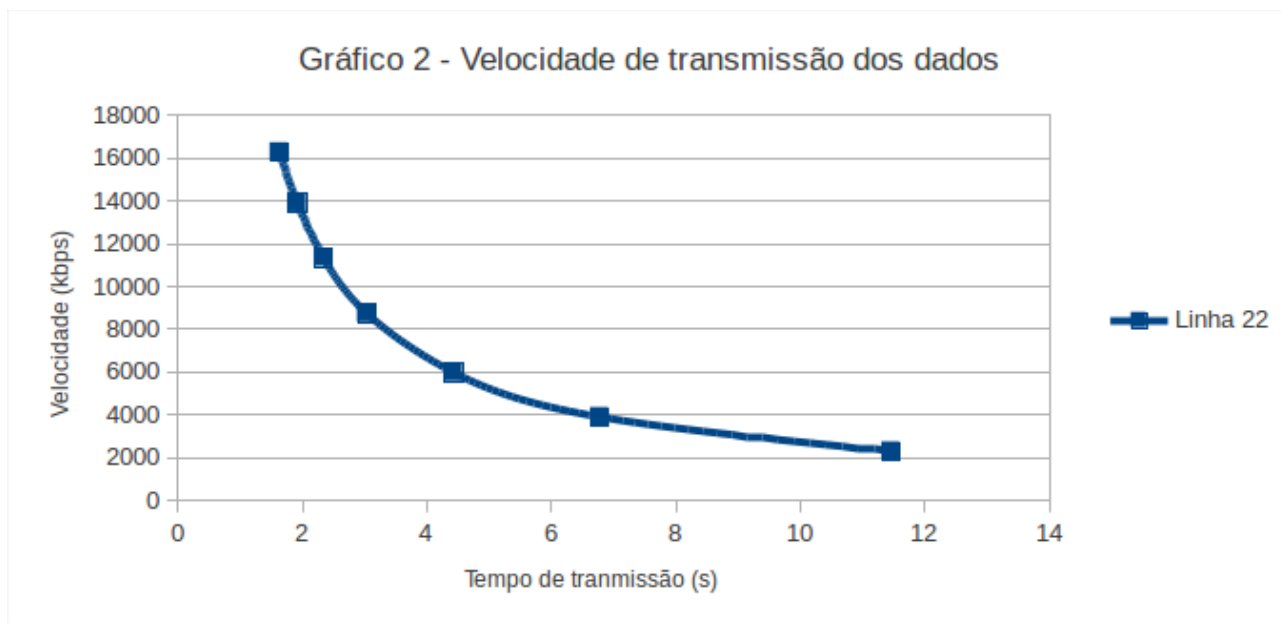
O gráfico abaixo mostra a variação do tempo de transmissão em função do tamanho do buffer.



A tabela abaixo fornece as informações sobre a velocidade de transmissão dos dados trocados entre o cliente e o servidor.

Tamanho do <i>buffer</i>	Medidas de velocidade de transmissão para cada tempo (em kbps)					Média	Desvio Padrão
	128	256	512	1024	2048		
128	2338	2262	2292	2353	2336	2316	38
256	3977	3808	3900	4018	3878	3916	83
512	6111	6097	5963	5923	5872	5993	106
1024	8610	8618	8743	8660	9109	8748	208
2048	10879	11340	11282	12034	11234	11354	420
4096	11980	14540	14726	14274	13966	13897	1109
8192	15870	16891	16773	15899	15937	16274	511

O gráfico abaixo mostra o comportamento da velocidade de transmissão em função do tempo de transmissão do arquivo, que está atrelado ao tamanho do *buffer*.



4. Análise

Com base nos resultados obtidos pode-se perceber que a partir de um dado tamanho de *buffer* as variações no tempo de transmissão e, conseqüentemente na velocidade de transmissão, mudam muito pouco. Isso se deve ao fato do comportamento exponencial dos valores do *buffer* em potências de 2^i . As variações são bem maiores para valores pequenos de *buffer*. O mesmo fato ocorre para os valores grandes de *buffer*, sendo assim, para um determinado tamanho de *buffer*, fica inviável o seu aumento pois acarretará em um uso grande da banda de transmissão (um *buffer* muito grande) sem um ganho expressivo na velocidade de transmissão.

O gráfico 2 apresentado, que corresponde às velocidades de transmissão não expressa muito bem o comportamento do efeito, sendo assim deveriam ser tomadas um número maior de medidas, a fim de expressar melhor o comportamento da velocidade de transmissão do *buffer*.

5. Conclusão

O presente trabalho pode mostrar o comportamento de um arquivo trafegando por uma conexão entre dois *sockets* UDP para diferentes tamanhos de *buffer*. Pode ser observado que há uma determinada faixa de tamanho de *buffer* que possibilita o uso adequado da rede sem sobrecarregá-la e sem estar desperdiçando banda, deixando-a ociosa.

O tamanho do pacote que trafega pela rede não pode ser muito pequeno, de forma que irá levar um tempo relativamente grande para que o mesmo percorra o caminho até chegar ao destinatário, desperdiçando banda da rede. E esse também não pode ser muito grande de forma que possa congestioná-la sem que haja de fato um ganho considerável no tempo de transmissão do arquivo.