

Trabalho Prático 1 – Redes de Computadores

Pedro Araujo Pires

O objetivo deste trabalho é implementar um par de programas utilizando a biblioteca de *sockets* do Unix, que exercitem tanto a transmissão unidirecional quanto a comunicação do tipo requisição-resposta sobre o protocolo TCP.

Os programas consistem em um cliente e um servidor, que funcionam da seguinte forma: o cliente envia um nome de arquivo pro servidor, que lê esse arquivo e o envia para o cliente. Ambos os programas contam o total de bytes enviados, e ao final da transferência o cliente imprime na tela informações sobre a transferência, como o tempo gasto e o *throughput* da transmissão.

Para testar esses programas, foram criados arquivos de texto de tamanho 2^i bytes, onde $0 \leq i \leq 16$, além de um arquivo de 3MB para testes com o servidor rodando localmente. O servidor é ligado, e o cliente requisita esses arquivos um a um, computando o tempo gasto e a velocidade de transmissão. O tempo gasto foi medido utilizando a função *gettimeofday*. De posse do tempo gasto e do total de bytes recebidos, o cliente faz o cálculo da velocidade da transmissão.

Os testes executados na mesma máquina foram feitos em um computador com processador Intel Core 2 2167MHz, com 2 GB de memória. Para deixar os testes mais fáceis de serem feitos, o servidor foi modificado para ficar em loop, aceitando a conexão do cliente, enviando o arquivo, e esperando por uma nova conexão novamente. Isso foi feito pois a especificação do trabalho determina que o servidor deve encerrar a execução após enviar o arquivo.

Os testes executados em máquinas diferentes utilizou o computador citado anteriormente para rodar o cliente, e o computador jaguar, presente na rede do DCC. As especificações da máquina jaguar foram retiradas do site do CRC, portanto não é possível listar suas características. O mesmo servidor modificado (em loop ao invés de uma execução por arquivo) foi utilizado. Os testes através da rede foram feitos através de uma conexão *wireless*, pois não havia um computador disponível para criar uma rede através de um cabo.

Abaixo seguem os resultados dos testes realizados no mesmo computador.

Tamanho (bytes)	Tempo (s)	Velocidade (kbps)	Bytes enviados
2	0.000660	2.96	2
4	0.000447	8.74	4
8	0.000502	15.56	8
16	0.000513	30.45	16
32	0.000423	73.86	32
64	0.000412	151.67	64
128	0.000430	290.72	128
256	0.000440	568.05	256
512	0.000485	1031.11	512
1024	0.000571	1752.76	1025
2048	0.000536	3735.14	2050
4096	0.000580	6903.22	4100
8192	0.000465	17220.90	8200
16384	0.000694	23073.89	16400
32768	0.001148	27902.31	32800
65536	0.001844	34739.20	65600

Tabela 1: testes no mesmo computador com buffer de 1024 bytes

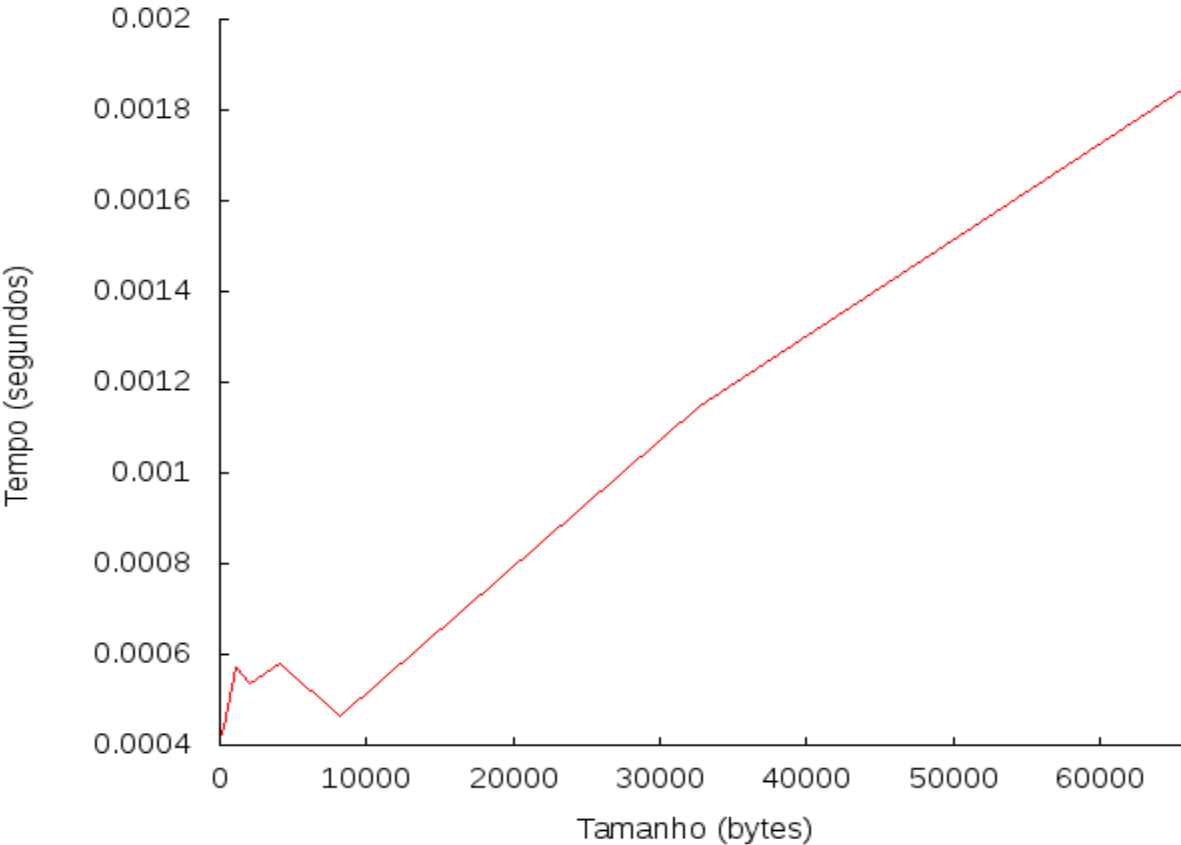


Gráfico 1: testes no mesmo computador com buffer de 1024 bytes

Tamanho (bytes)	Tempo (s)	Velocidade (kbps)	Bytes enviados
2	0.000513	3.81	2
4	0.000443	8.82	4
8	0.000456	17.13	8
16	0.000450	34.72	16
32	0.000443	70.53	32
64	0.000556	112.41	64
128	0.000435	287.37	128
256	0.000535	467.25	256
512	0.000514	974.63	513
1024	0.000481	2082.61	1026
2048	0.000485	4132.10	2052
4096	0.000509	7872.43	4104
8192	0.001507	5318.95	8208
16384	0.001183	13552.00	16416
32768	0.001174	27313.10	32832
65536	0.002262	28350.20	65664

Tabela 2: testes no mesmo computador com buffer de 512 bytes

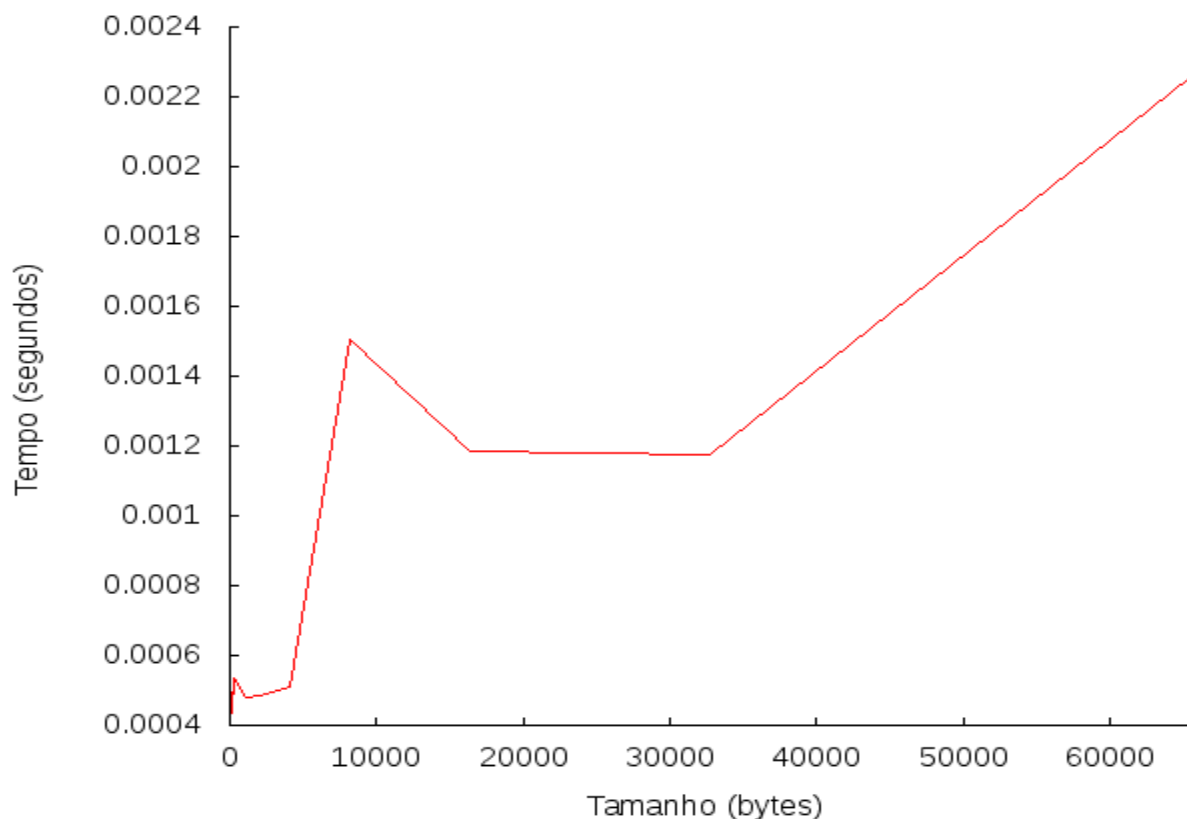


Gráfico 2: testes no mesmo computador com buffer de 512 bytes

Tamanho (bytes)	Tempo (s)	Velocidade (kbps)	Bytes enviados
2	0.024966	0.08	2
4	0.023357	0.17	4
8	0.021021	0.37	8
16	0.031431	0.50	16
32	0.012742	2.45	32
64	0.015855	3.94	64
128	0.017032	7.34	128
256	0.067662	3.69	256
512	0.028061	17.82	512
1024	0.028904	34.63	1025
2048	0.015613	128.22	2050
4096	0.040260	99.45	4100
8192	0.035217	227.38	8200
16384	0.050653	316.18	16400
32768	0.059804	535.60	32800
65536	0.281471	227.60	65600

Tabela 3: testes em computadores diferentes com buffer de 1024 bytes

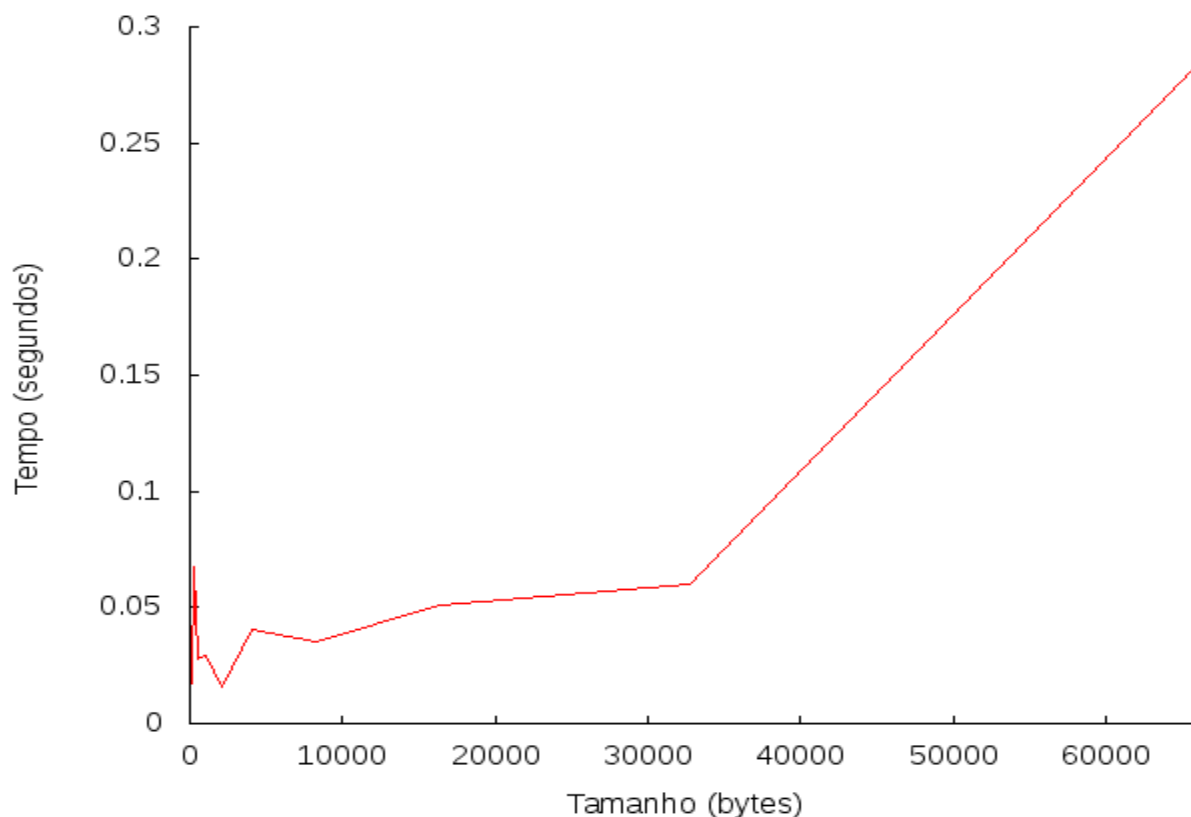


Gráfico 3: testes em computadores diferentes com buffer de 1024 bytes

Tamanho (bytes)	Tempo (s)	Velocidade (kbps)	Bytes enviados
2	0.004266	0.46	2
4	0.004624	0.84	4
8	0.004963	1.57	8
16	0.003742	4.18	16
32	0.003894	8.03	32
64	0.003826	16.34	64
128	0.004879	25.62	128
256	0.005871	42.58	256
512	0.004409	113.63	513
1024	0.007047	142.18	1026
2048	0.009501	210.92	2052
4096	0.009296	431.13	4104
8192	0.016911	473.99	8208
16384	0.020618	777.54	16416
32768	0.032123	998.12	32832
65536	0.081799	783.93	65664

Tabela 4: testes em computadores diferentes com buffer de 512 bytes

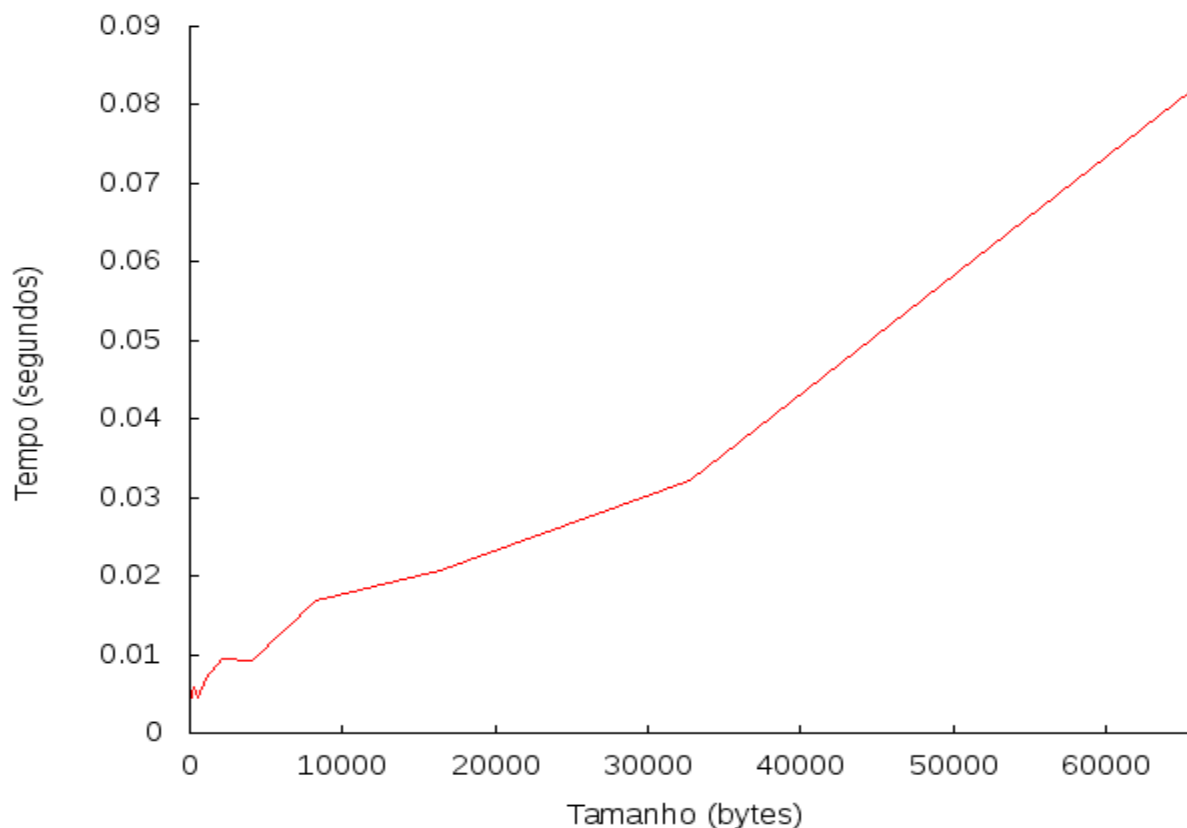


Gráfico 4: testes em computadores diferentes com buffer de 512 bytes

Ao analisar os resultados, a primeira informação observada é que, enquanto o arquivo enviado é menor do que o *buffer*, é enviado exatamente o total de bytes do arquivo. Porém, quando o tamanho do arquivo excede o tamanho do *buffer*, alguns bytes a mais são enviados. Mais especificamente, quando o arquivo é do mesmo tamanho que o *buffer*, é enviado exatamente 1 byte a mais, e essa proporção (1 byte por *buffer*) continua indefinidamente (pelo menos até onde os testes foram). Como somente os bytes do arquivo que estão no *buffer* são enviados, esse byte adicional deve ser algum tipo de indicação de que ainda existem pacotes a serem enviados.

Outro dado que chama a atenção é o fato de, nos testes locais (e em parte nos testes através da rede), a velocidade aproximadamente dobrar sempre que o tamanho do arquivo dobra. Devido ao fato de os programas estarem muito próximos (na mesma máquina), o tamanho da mensagem não é o que limita o tempo de transmissão, mas sim o processamento necessário para enviar a mensagem. É possível ver que isso também ocorre com as mensagens enviadas através da rede, porém esse efeito para de ocorrer antes.

Também é possível perceber que a velocidade de transmissão nos testes feitos através da rede abtiveram uma velocidade de transmissão muito alta para os padrões da internet brasileira. Isso se deve ao fato de os dois computadores estarem a uma máquina de distância. Abaixo está a saída do programa `traceroute` para a máquina *jaguar.dcc.ufmg.br*:

```
1  chronos-usuarios.dcc.ufmg.br (150.164.7.129)  2.838 ms  3.676 ms  3.813 ms
2  jaguar.dcc.ufmg.br (150.164.0.199)  4.029 ms  4.227 ms  4.406 ms
```

Os resultados obtidos com esse trabalho estava de acordo com o esperado, exceto o fato que ocorreu com os bytes enviados a mais quando o tamanho do arquivo sendo enviado excedia o tamanho do *buffer*. Os testes poderiam ter sido mais extensos, por exemplo utilizando uma rede conectada por cabo, mas não foi possível. Mesmo assim foi possível aprender como programar utilizando a biblioteca de *sockets* do Unix, e como que um par de programas cliente/servidor funciona. Todos os gráficos foram feitos utilizando o programa `gnuplot`, disponível em <http://www.gnuplot.info>.