

# **MAC0448/5910**

# **Programação para Redes de Computadores**

## **Exercício-Programa 2**

Gustavo Teixeira da Cunha Coelho  
Henrique Gemignani Passos Lima

Tanto o cliente como o servidor separam comandos usando quebras de linha (`\n`). Logo, as mensagens a seguir assumem que existe uma quebra de linha ao final da mensagem.

**Servidor**

# Mensagens enviadas ao Servidor:

LOGIN <USERNAME> <LISTEN\_PORT>

Um cliente requer login no servidor com nome de <USERNAME>. Ele diz ao servidor que estará escutando na porta <LISTEN\_PORT>.

# Mensagens enviadas ao Servidor:

HEARTBEAT

Um cliente mandou uma mensagem ao servidor para evitar que a conexão seja fechada por timeout.

# **Mensagens enviadas ao Servidor:**

LISTUSERS

Um cliente requer ao servidor a lista de todos os usuários.

# Mensagens enviadas ao Servidor:

QUERYUSERINFO <USERNAME>

Um cliente requer ao servidor o endereço ip e porta do cliente <USERNAME>.

# **Mensagens enviadas ao Servidor:**

LOGOUT

Um cliente requer a finalização da conexão.



**Cliente**

# **Mensagens recebidas do Servidor:**

WELCOME

O servidor aceitou a nossa conexão.

# Mensagens recebidas do Servidor:

ALREADY\_LOGGED\_IN

Recebida após requisitar conexão com o servidor quando já estamos conectados com ele.

# **Mensagens recebidas do Servidor:**

`USER_ALREADY_EXISTS`

O nome de usuário requisitado já está em uso.

# Mensagens recebidas do Servidor:

UNKNOWN\_USER

O usuário com o qual requisitamos uma conexão não está conectado ao servidor.

# **Mensagens recebidas do Servidor:**

OK

Resposta padrão do servidor a um heartbeat.

## **Mensagens de comunicação entre clientes:**

CHATREQUEST <USERNAME>

O usuário <USERNAME> requer uma sessão de troca de mensagens com o nosso cliente.

# **Mensagens de comunicação entre clientes:**

OK

O cliente ao qual solicitamos uma conexão aceitou nossa requisição de conversa.



# **Mensagens de comunicação entre clientes:**

**BUSY**

O cliente ao qual solicitamos uma conexão já está conversando com outro cliente.

## **Mensagens de comunicação entre clientes:**

**SAY <MESSAGE>**

O cliente remoto nos mandou uma mensagem de conversa <MESSAGE>.

## Mensagens de comunicação entre clientes:

SENDFILE <FILE\_NAME> <FILE\_SIZE>  
<NEW\_PORT>

Um cliente requer uma transferência de arquivo com nome <FILE\_NAME> e tamanho <FILE\_SIZE> com uma nova conexão, aonde ele estará escutando no port <NEW\_PORT>. Podemos responder a essa mensagem com um TRANSFEROK ou um

## **Mensagens de comunicação entre clientes:**

**TRANSFEROK <RECEIVE\_PORT>**

A requisição de transferência de arquivo para o cliente remoto foi aceita. Ele escutará a conexão na porta <RECEIVE\_PORT>.

# **Mensagens de comunicação entre clientes:**

TRANSFERREFUSED

A requisição de transfência de arquivo para o cliente remoto foi recusada.

# **Mensagens de comunicação entre clientes:**

CLOSE

O cliente remoto requer o fim da conversa.

# Ambiente

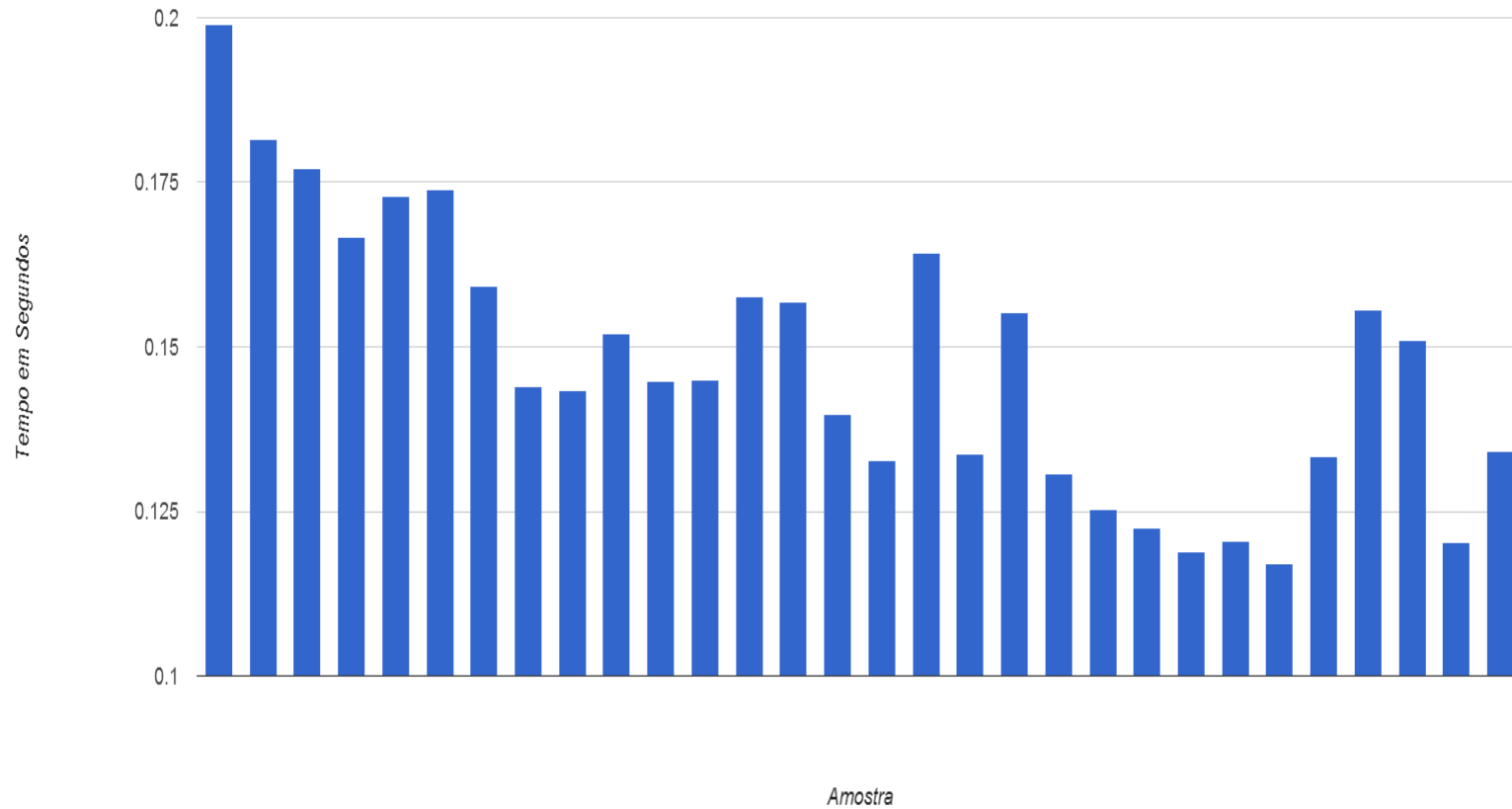
Os testes foram realizados enviando arquivos de máquina virtual para a máquina física, utilizando um servidor que fica em outra máquina.

O sistema de virtualização utilizado foi o xen, com interfaces bridged, rodando em Debian Squeeze.

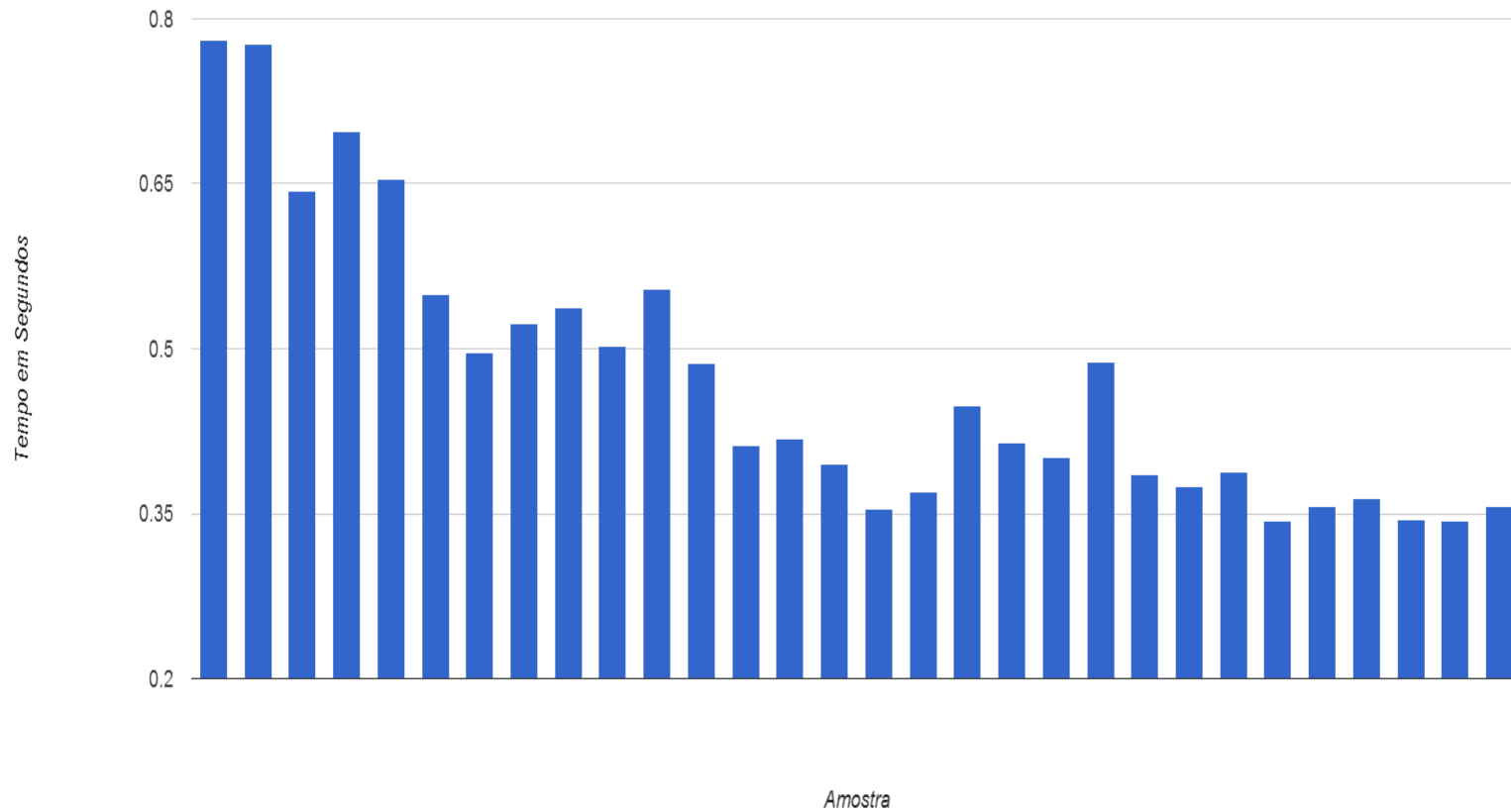
# **Resultados das Transferências de Arquivos**



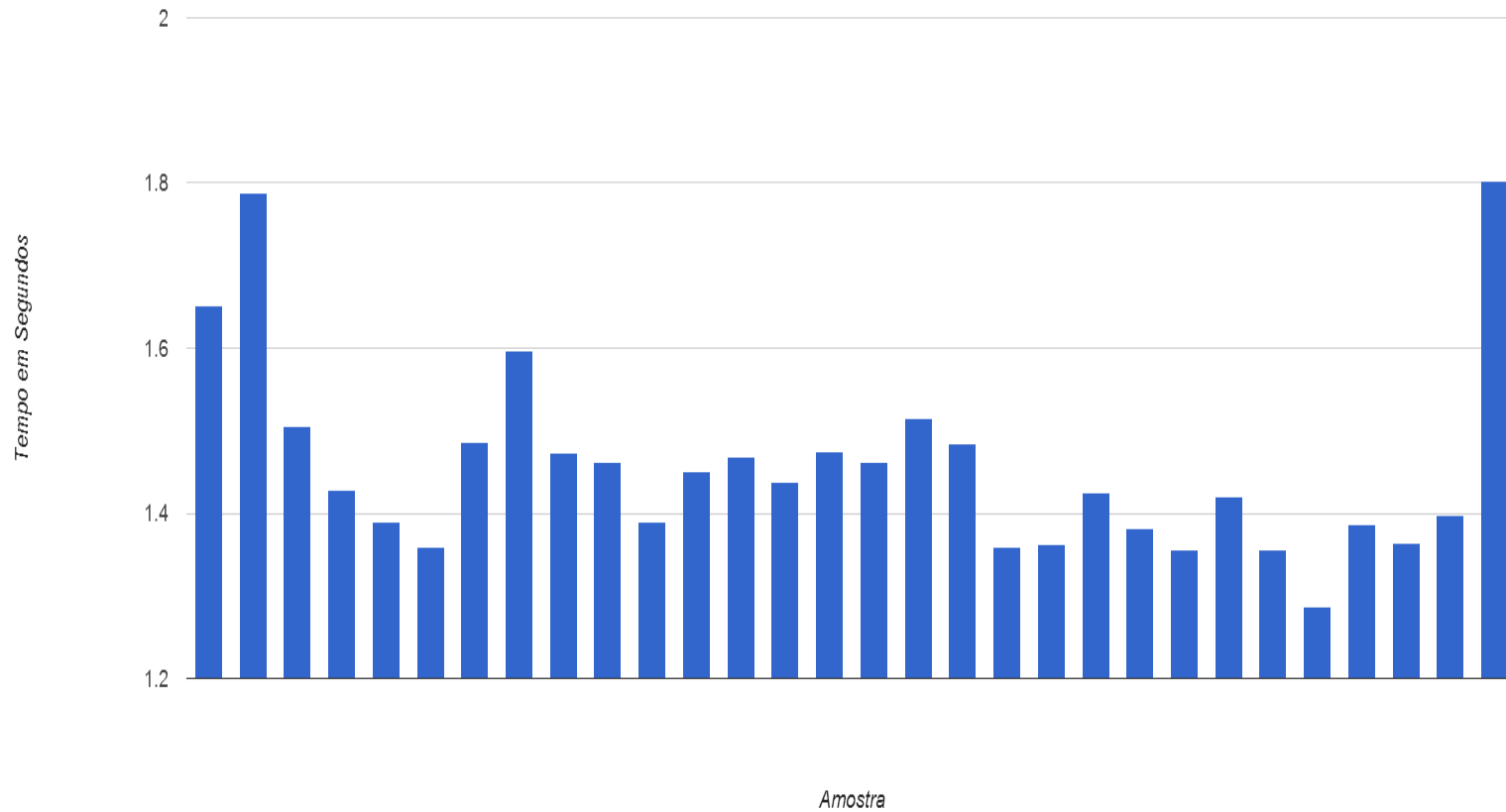
Arquivos de 10 Mega usando TCP



Arquivos de 50 Mega usando TCP



Arquivos de 100 Mega usando TCP



# Resultados

- TCP: velocidade de transferência foi abaixo da esperada para o link disponível.
- UDP: fracasso total na transferência de arquivos. Somente em loopback todos os pacotes eram recebidos.