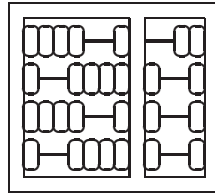


# UNIVERSIDADE ESTADUAL DE CAMPINAS

## INSTITUTO DE COMPUTAÇÃO



### SERVIDOR DE AGENDA BASEADO EM SOCKET TCP

*Relatório do primeiro laboratório de MC823*

**Aluno:** Marcelo Keith Matsumoto    **RA:** 085937

**Aluno:** Tiago Chedraoui Silva    **RA:** 082941

#### Resumo

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Servidor de agenda</b>	<b>1</b>
<b>3</b>	<b>Ambiente de implementação</b>	<b>2</b>
<b>4</b>	<b>Tempos de comunicação e total</b>	<b>2</b>
<b>5</b>	<b>Conclusão</b>	<b>3</b>

# 1 Introdução

O TCP (Transmission Control Protocol) é um protocolo do nível da camada de transporte [3]. Dentre suas principais características temos:

**Orientado à conexão** Cliente e o servidor trocam pacotes de controle entre si antes de enviarem os pacotes de dados.

Isto é chamado de procedimento de estabelecimento de conexão (handshaking).

**Transferência garantida** Dados trocados são livres de erro, o que é conseguido a partir de mensagens de reconhecimento e retransmissão de pacotes.

**Controle de fluxo** Assegura que nenhum dos lados da comunicação envie pacotes rápido demais, pois uma aplicação em um lado pode não conseguir processar a informação na velocidade que está recebendo.

**controle de congestão** Ajuda a prevenir congestionamentos na rede.

**Fim-a-fim** Conexão é sempre entre o host emissor e o host receptor.

Cada um dos segmentos da camada transporte tem em seu cabeçalho campos denominado número de portas que indicam a qual processo o mesmo deve ser entregue, existindo um número de porta do emissor e o número de porta do receptor.

Possuindo as duas portas, pode-se realizar uma conexão entre elas conhecida por socket. Com o socket é possível diferenciar os canais utilizados por cada processo de aplicação.

Os segmentos TCP, através do protocolo IP, são entregues a sistemas terminais, cada um identificado por seu endereço IP. E o protocolo TCP cuida de repassar os dados à cada processo de aplicação de acordo com porta especificada no cabeçalho do segmento.

Devido a importância do protocolo, este laboratório tem o objetivo de medir o tempo total e de comunicação de uma conexão TCP entre um cliente e um servidor.

## 2 Servidor de agenda

O sistema implementado, uma agenda distribuída, se baseia numa comunicação cliente-servidor. Na qual o servidor possui todas as informações da agenda que estão armazenadas em um banco de dados, assim como as opções de interações com os dados que são apresentadas aos clientes em formas de um menu. O cliente só escolhe alguma opção de interação com os dados de acordo com menu. No menu inicial pode-se logar com um usuário ou criar um usuário. E cada usuário possui sua agenda. Assim, dentre as possibilidades de interações para um usuário logado tem-se:

- Inserção de um compromisso que possui um nome, dia, hora, e minuto.
- Remoção de um compromisso através de seu nome
- Pesquisa de compromisso por dia
- Pesquisa de compromisso por dia e hora

- Ver todos os compromisso de mês de abril

### 3 Ambiente de implementação

O sistema de agenda foi implementado na linguagem C. Os dados da agenda foram armazenados em arquivos, onde o servidor lê quando um usuário loga no sistema de agenda e os armazena em memória. A cada alteração na agenda o servidor atualiza as informações dos arquivos.

O sistema de agenda foi implementado e executado nos seguintes sistemas operacionais :

- FC14 - Fedora Laughlin Linux 2.6.35.11

### 4 Tempos de comunicação e total

O round-trip time (RTT) é o tempo que leva-se para um sinal ser enviado mais o tempo que se leva para receber um acknowledgment que o sinal foi recebido. A ferramenta administrativa para as redes de computadores denominada “Ping” [2] é usada para testar se um host é alcançável e para medir o RTT para mensagens enviadas do host remetente para o destinatário.

Inicialmente, implementamos um programa semelhante ao ping para o cálculo da RTT. Com ele foi possível calcular várias vezes o tempo de envio pacote de 1 byte para o servidor e esse responder com um pacote de 4 bytes. Utilizando um script para a coleta dos tempo, obtivemos o seguintes valores:

Tabela I: Ping implementado

Valor	Tempo
Max	17.814 ms
Min	0.045 ms
Média	0.059 ms
Desvio	0.232 ms

Posteriormente, aplicamos o cálculo de tempo ao programa principal de forma a obtermos o tempo total e tempo de comunicação. Para o tempo total, no cliente pega-se o tempo antes do primeiro send e após o último recv. Para o tempo de comunicação, no servidor pega-se o tempo após o primeiro recv e antes do último send. Subtraindo o segundo do primeiro você tem o tempo estimado de ida e volta de um pacote.

O resultado obtido para 100 valores foi:

Comparando os resultados, como o ping é muito rápido nossa precisão é muito baixa. Porém nosso programa não é tão rápido assim os valores são mais precisos. Para certificarmos do tempo, fizemos o teste com três situações, uma mais rápida, uma mais lenta. E obtemos valores muito próximos da média de tempo de ida e volta de um pacote(RTT).

Tabela II: Teste 1: conexão e fechamento de conexão com servidor

Valor	Tempo
Max	4.986 ms
Min	0.097 ms
Média	0.608 ms
Desvio	0.008 ms

Tabela III: Teste 2: conexão, login na conta, ver agenda do mês e fechamento de conexão com servidor

Valor	Tempo
Max	4.959 ms
Min	0.190 ms
Média	0.638 ms
Desvio	0.013 ms

## 5 Conclusão

## Referências

- [1] Brian "Beej" Jorgensen "Hall" Beej's Guide to Network Programming - Using Internet Sockets . Disponível em <http://beej.us/guide/bgnet/>, [Último acesso: 07/04/2011].
- [2] Mike Muuss Packet Internet Grouper (Gropen) . Disponível em <http://linux.die.net/man/8/ping>, [Último acesso: 10/04/2011].
- [3] J. Kurose e K. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Pearson Addison Wesley, 3 ed., 2005.