

# Curso de Bacharelado em Ciência da Computação

Disciplina: Introdução à Redes de Computadores

Avaliação /2015-1

Prof Mauro Oliveira

Aluno (a): \_\_\_\_\_

## 1) Codifique com TCP, UDP, (A)mbos, (N)enhum dos dois

- |   |  |
|---|--|
| <input type="checkbox"/> Multiplexação / Demultiplexação        | <input type="checkbox"/> Utiliza Sockets           |
| <input type="checkbox"/> Protocolo da camada de Transporte      | <input type="checkbox"/> Orientado à conexão       |
| <input type="checkbox"/> Controle de congestionamento           | <input type="checkbox"/> Connectionless            |
| <input type="checkbox"/> Faz uso do protocolo IP                | <input type="checkbox"/> Faz roteamento            |
| <input type="checkbox"/> Possui IP no cabeçalho                 | <input type="checkbox"/> Faz Controle de fluxo     |
| <input type="checkbox"/> Corrige erro na mensagem transmitida   | <input type="checkbox"/> Detecta erro              |
| <input type="checkbox"/> Possui número das portas no cabeçalho  | <input type="checkbox"/> Lida com Datagramas       |
| <input type="checkbox"/> Cria Fluxo de Entrada ligado ao Socket | <input type="checkbox"/> Usa a biblioteca java.net |
| <input type="checkbox"/> É usado pelo protocolo SMTP            | <input type="checkbox"/> É usado pelo SNMP         |
| <input type="checkbox"/> É usado pelo comando PING              | <input type="checkbox"/> É usado na porta 21 (FTP) |
| <input type="checkbox"/> Entrega confiável e ordenada           | <input type="checkbox"/> Dá garantias de atraso    |
| <input type="checkbox"/> Dá garantias de largura de banda       | <input type="checkbox"/> Usa o protocolo GO-BACK-N |
| <input type="checkbox"/> Entrega não confiável e não ordenada   | <input type="checkbox"/> Estabelece conexão        |
| <input type="checkbox"/> É modelado por máquina de estados      | <input type="checkbox"/> Encapsula pacote IP       |
| <input type="checkbox"/> Comunicação lógica entre processos     | <input type="checkbox"/> É encapsulado pelo IP     |
| <input type="checkbox"/> Comunicação lógica entre máquinas      | <input type="checkbox"/> Encapsula mensagem        |

## 2) Considere abaixo. Fig 1: modelo de um protocolo de transporte simples e Fig 2: seis (A,B,C,D,E e F) eventos/ações que ocorrem neste protocolo

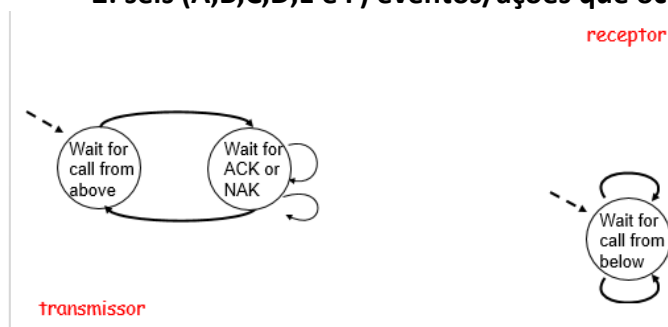


Figura 1: Modelo do protocolo

## Figura 2: Eventos e Ações a serem mapeadas no protocolo acima

- |   |   |   |
|---|---|---|
| A) $\frac{\text{rdt\_rcv(rcvpkt) \&\& notcorrupt(rcvpkt)}}{\text{extract(rcvpkt,data) deliver\_data(data) udt\_send(ACK)}}$ | B) $\frac{\text{rdt\_rcv(rcvpkt) \&\& isNAK(rcvpkt)}}{\text{udt\_send(sndpkt)}}$                | C) $\frac{\text{rdt\_rcv(rcvpkt) \&\& isACK(rcvpkt)}}{\Lambda}$                 |
| D) $\frac{\text{timeout}}{\text{udt\_send(sndpkt) start\_timer}}$   | E) $\frac{\text{rdt\_send(data)}}{\text{snkpkt = make\_pkt(data, checksum) udt\_send(sndpkt)}}$ | F) $\frac{\text{rdt\_rcv(rcvpkt) \&\& corrupt(rcvpkt)}}{\text{udt\_send(NAK)}}$ |

- 2.1 Associe os seis (A,B,C,D,E e F) eventos/ações às transições na figura do protocolo
- 2.2 Que tipos de serviços são disponibilizados pelo protocolo? Ex: perda de informação
- 2.3 O protocolo tem mecanismo de correção de perda de reconhecimento? Justifique
- 2.4 Idem para controle de fluxo.

### 3. Considere agora que o endereço da empresa seja IP 200.127. 32.0 /19

- 5.1 Quantas máquinas podem ser endereçadas na empresa? Justifique!
- 5.2 Identifique, em decimal, o primeiro e o último endereço na empresa.
- 5.3 Considere 4 subredes. Quantos endereços tem cada subrede?
- 5.4 Identifique, em decimal, os endereços inicial e final de cada subrede. Qual o endereço com máscara (A.B.C.D / x) de cada sub rede acima?

### 4. Faça o que se pede

- 3.1 Identifique que tipo de serviço o programa abaixo implementa:  
 Cliente TCP     Servidor TCP     Cliente UDP     Servidor UDP  
 Nível de Aplicação     Aplicação Usuário     Transporte     Rede
- 3.2 Que serviço programa abaixo o implementa (ou faz parte) ?
- 3.3 Preencha os espaços laterais do programa com expressões apropriadas, abaixo:

*Cria SOCKET para Datagrama / Aloca Memória para receber Datagrama  
 Cria Fluxo de Saída ligado ao SOCKET/ Envia Linha ao Servidor  
 Cria Fluxo de Entrada / Cria SOCKET de Cliente, conexão ao Servidor  
 Cria Fluxo de Entrada ligado ao SOCKET / Lê linha do Servidor  
 Cria SOCKET para recepção / Aguarda contato do Cliente no SOCKET de recepção  
 Lê linha do SOCKET / Escreve Linha ao SOCKET / Cria SOCKET do Cliente  
 Envia linha ao Servidor / Cria Datagrama / Lê linha do Servidor  
 Envia Datagrama ao Servidor / Lê Datagrama do Servidor / Recebe Datagrama*

```
import java.io.*;
import java.net.*;
class ClienteTCP {

    public static void main(String argv[]) throws Exception
    {
        String frase;
        String fraseModificada;

        BufferedReader doUsuario =
            new BufferedReader(new InputStreamReader(System.in));

        Socket socketCliente = new Socket("nomeHoso", 6789);

        DataOutputStream paraServidor =
            new DataOutputStream(socketCliente.getOutputStream());

        BufferedReader doServidor =
            new BufferedReader(new
            InputStreamReader(socketCliente.getInputStream()));

        frase = doUsuario.readLine();

        paraServidor.writeBytes(frase + '\n');

        fraseModificada = doServidor.readLine();

        System.out.println("Do Servidor: " + fraseModificada);

        socketCliente.close();

    }
}
```