Redes de Computadores – versão ECE

Exercício Programático em Java

**1. Definição**

Nesse exercício deverá implementar em Java um protocolo com paralelismo de transporte confiável, executado acima do UDP, assumindo que o TCP não existe. Para isso, **será obrigatório utilizar os conceitos de Go-Back-N ou de Repetição seletiva (você escolhe um deles).**

O protocolo será utilizado por um programa que realiza a transmissão de mensagens, estilo chat, entre um *sender* (quem envia) e um *receiver* (quem recebe) executados em máquinas diferentes. A mensagem a ser transmitida deverá ter um formato (cabeçalho) que atenda as especificações.

**2. Objetivos**

Nesse exercício será necessário:

1. Implementar o tratamento de pacotes perdidos.
2. Implementar o tratamento de pacotes fora de ordem.
3. Implementar o tratamento de pacotes duplicados.
4. Implementar o tratamento de pacotes lentos.
5. Simular (pelo *sender*) pacotes lentos, perdidos, fora de ordem, duplicados e normal. Nesse ponto, assim que o usuário digitar uma mensagem, o *sender* poderá enviar a mensagem com um retraso (simulando a lentidão), não enviá-la (simulando a perda), enviá-la após outra mensagem (simulando o desordem) ou enviá-la duplicada ou normalmente
6. Implementar o buffer de pacotes (recebidos, ack, enviados, etc) no *sender* e no *receiver*.
7. Mostrar na console do *sender* um menu com as diferentes opções de envio (lento, perda, fora de ordem, duplicados e normal) para cada mensagem. Para o envio duplicado, envia a mesma mensagem duas vezes (o *receiver* deverá notar esses problemas).

Para o caso dos números de sequência e de ack, deve assumir que cada um corresponde a um pacote (cadeia de bytes completa) e não a um byte específico da cadeia (como no TCP). Além disso, não deverá implementar o checksum.

**3. Entrega**

A entrega é individual e consistirá em um relatório, o código fonte e um arquivo que mostre como compilar e executar o programa a ser entregue pelo TIDIA, na aba atividades. Envie no TIDIA um arquivo (obrigatoriamente .zip) que contenha o pdf do relatório, o arquivo como compilar/executar e os .java. Não envie os .class.

3.1. Relatório

O relatório deverá ter obrigatoriamente as seguintes seções:

a) Nome e RA do participante

b) Formato da mensagem transferida

c) Explicação em “alto nível” do tratamento de mensagens lentas.

d) Explicação em “alto nível” do tratamento de mensagens perdidas.

e) Explicação em “alto nível” do tratamento de mensagens fora de ordem.

f) Explicação em “alto nível” do tratamento de mensagens duplicadas.

g) Explicação em “alto nível” do funcionamento e consumo do buffer

h) Links dos lugares de onde baseou seu código (caso aplicável).

3.2. Código fonte

O código fonte deverá apresentar claramente nas respectivas consoles as opções dos diversos casos especificados anteriormente. Ou seja, a) Use /\* \*/ para comentar o código nas partes que trate dos casos 1-7 da seção Objetivos; b) Use o System.out.println para mostrar as mensagens na console.

Por exemplo, se o *sender* está enviando uma mensagem duplicada, a console do *sender* deverá mostrar que está sendo enviada a mensagem duplicadas e o *receiver* deverá mostrar quais são as mensagens duplicadas recebidas.

* Assim, na console do *sender* mostre:
  + Mensagem “oi” enviada de forma duplicada com id X.
  + Mensagem id X recebida pelo *receiver*
* Na console do *receiver* mostre:
  + Mensagem id X “oi” recebida
  + Mensagem id X “oi” recebida de forma duplicada.

O uso de livrarias que realizem parte das funcionalidades pedidas não será aceito. Caso tenha dúvidas, pergunte ao professor se pode usar determinada livraria.

3.3. Execução

Deverá ser entregue um arquivo mostrando o script de como compilar e de como executar seu programa, com os parâmetros de entrada e as livrarias necessárias, etc.

**4. Avaliação**

No código será avaliado:

* O funcionamento do tratamento de mensagens lentas.
* O funcionamento do tratamento de mensagens perdidas.
* O funcionamento do tratamento de mensagens fora de ordem.
* O funcionamento do tratamento de mensagens duplicadas.
* O funcionamento do consumo do buffer

No relatório será avaliado:

* O formato da mensagem valerá 1 ponto.
* O texto valerá 2 pontos (somente caso o código tenha sido implementado)

Na execução será avaliado:

* Caso o código não compile ou não possa ser executado, terá nota zero, independentemente de ter entregue o relatório. Por isso, atente ao ponto 3.3.

**5. Atrasos**

Para os atrasos, a nota máxima (após a avaliação na aula prática) será baseada na tabela abaixo.

|  |  |
| --- | --- |
| **Dias de atraso** | **Nota máxima** |
| 0 | 10 |
| 1 | 7 |
| 2 | 6 |
| 3 | 5 |
| +3 | 0 |

**6. Links recomendados**

Informações sobre programação com UDP podem ser encontradas em:

<https://www.baeldung.com/udp-in-java>

<https://www.geeksforgeeks.org/working-udp-datagramsockets-java/>

<https://docs.oracle.com/javase/tutorial/networking/datagrams/index.html>

**7. Ética**

Cola, fraude, ou plágio implicará na nota zero a todos os envolvidos em todas as avaliações e exercícios programáticos da disciplina.