

Trabalho sobre a camada de transporte utilizando protocolo TCP e UDP

O TCP (*Transmission Control Protocol*) é um protocolo robusto da camada de transporte que provê confiabilidade, garantia de integridade e a ordem das informações. Por outro lado, o UDP (*User Datagram Protocol*) é um protocolo da camada de transporte mais leve, indicado para o fluxo de dados em tempo real para aplicações que admitem tolerância à perda de pacotes ou corrompimento dos dados.

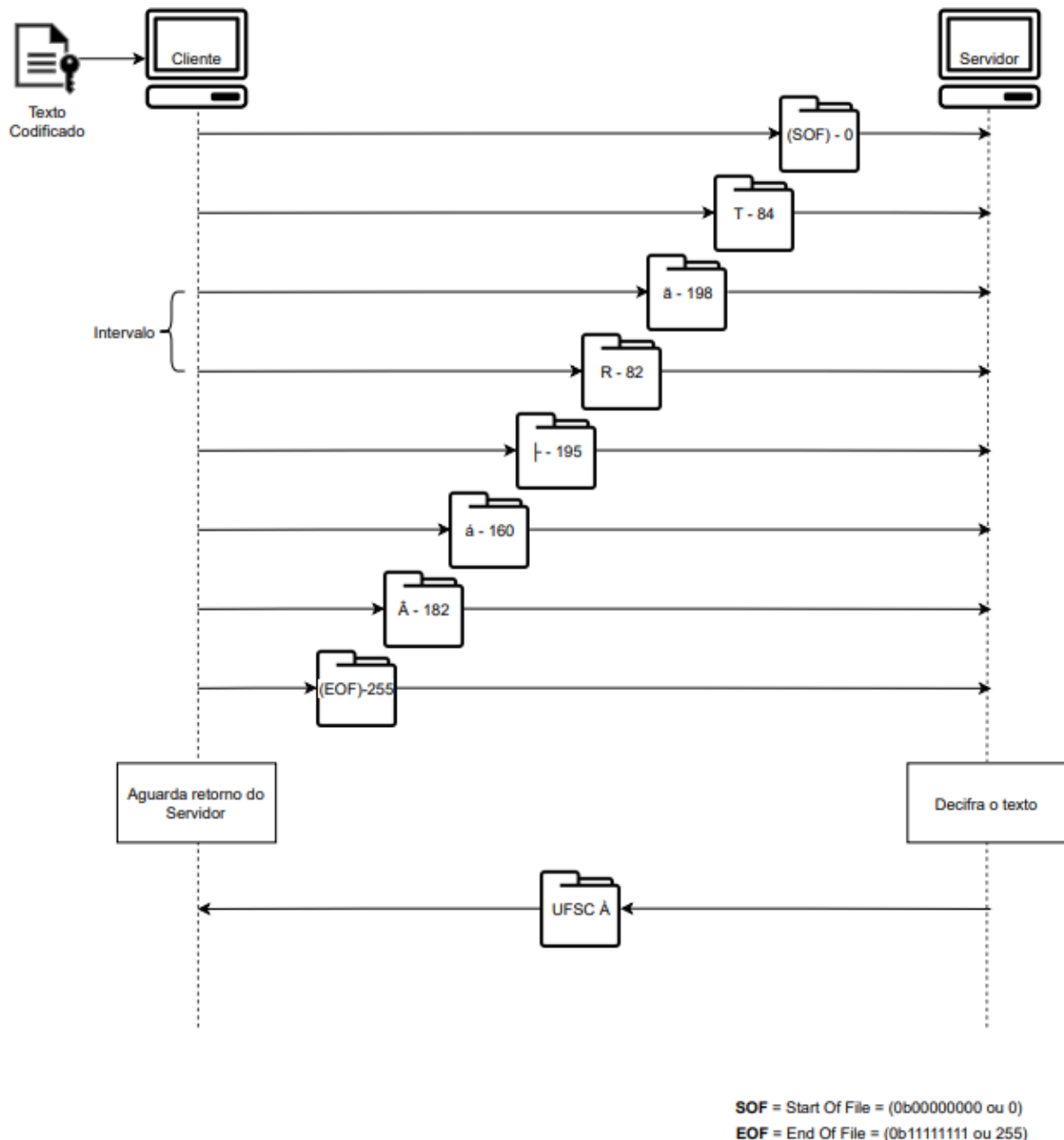
1. Primeira etapa do trabalho:

1.1. Implemente aplicações Cliente e Servidor utilizando o protocolo TCP para troca de mensagens através de *Sockets*, utilize a linguagem de sua preferência (Python, C, ...).

1.2. Implemente as mesmas aplicações Cliente e Servidor utilizando o protocolo UDP.

1.3. O hospedeiro Cliente deve enviar uma mensagem de texto ao hospedeiro Servidor para ser decifrado, o texto original está codificado. Assim que o Cliente terminar de enviar o texto por completo, ficará aguardando o retorno por ele decifrado, no qual o Servidor deve decodificar e retorná-lo para o Cliente.

1.4. Cada caractere do texto deve ser enviado individualmente em pacotes/mensagens, seguindo um intervalo de envio entre os pacotes predefinido, conforme exemplo abaixo.



1.5. Variar o intervalo de envio entre pacotes (0.1s, 0.01s e 0s/Nenhum), apresente um relatório com os dados recebidos no Servidor (enviados pelo Cliente) e do Cliente (enviado pelo Servidor), tempo total para troca de mensagens e total de pacotes perdidos ou corrompidos para ambos os protocolos TCP e UDP.

1.6. Explique se houve diferença, com base nos resultados entre as implementações utilizando TCP e UDP, e quais os principais motivos.

1.7. Cada grupo de alunos deverá utilizar o seu arquivo de texto correspondente (em formato .TXT) que contém a mensagem codificada.

2. Segunda etapa do trabalho:

2.1. Implemente aplicações Cliente e Servidor utilizando o protocolo UDP para troca de mensagens através de *Sockets*. Desenvolva um protocolo, em nível de aplicação, para recuperar os pacotes perdidos pela camada de transporte a fim de decifrar corretamente o texto codificado. Você poderá adicionar cabeçalhos nas mensagens, enumerar cada mensagem, criar pacotes de confirmação de recebimento, retransmissão de mensagens perdidas ou não recebidas, criar *buffers*, *timeouts*...

2.2. Faça uma descrição do protocolo de troca de mensagens que foi implementado, com a ajuda de fluxogramas e/ou máquinas de estados, e apresente um relatório com as dados/mensagens recebidos no Servidor (enviados pelo Cliente) e do Cliente (enviados pelo Servidor), tempo total para troca de mensagens e total de pacotes perdidos ou retransmitidos.

2.3. Não é necessário descrever como foi implementado o algoritmo para decodificar os textos, apenas o protocolo para troca de mensagens do Cliente e do Servidor.

2.4. Cada grupo de alunos deverá utilizar o seu arquivo de texto correspondente (em formato .TXT) que contém a mensagem codificada.

3. Método para Decodificar:

3.1. Como o Servidor decifra o texto?

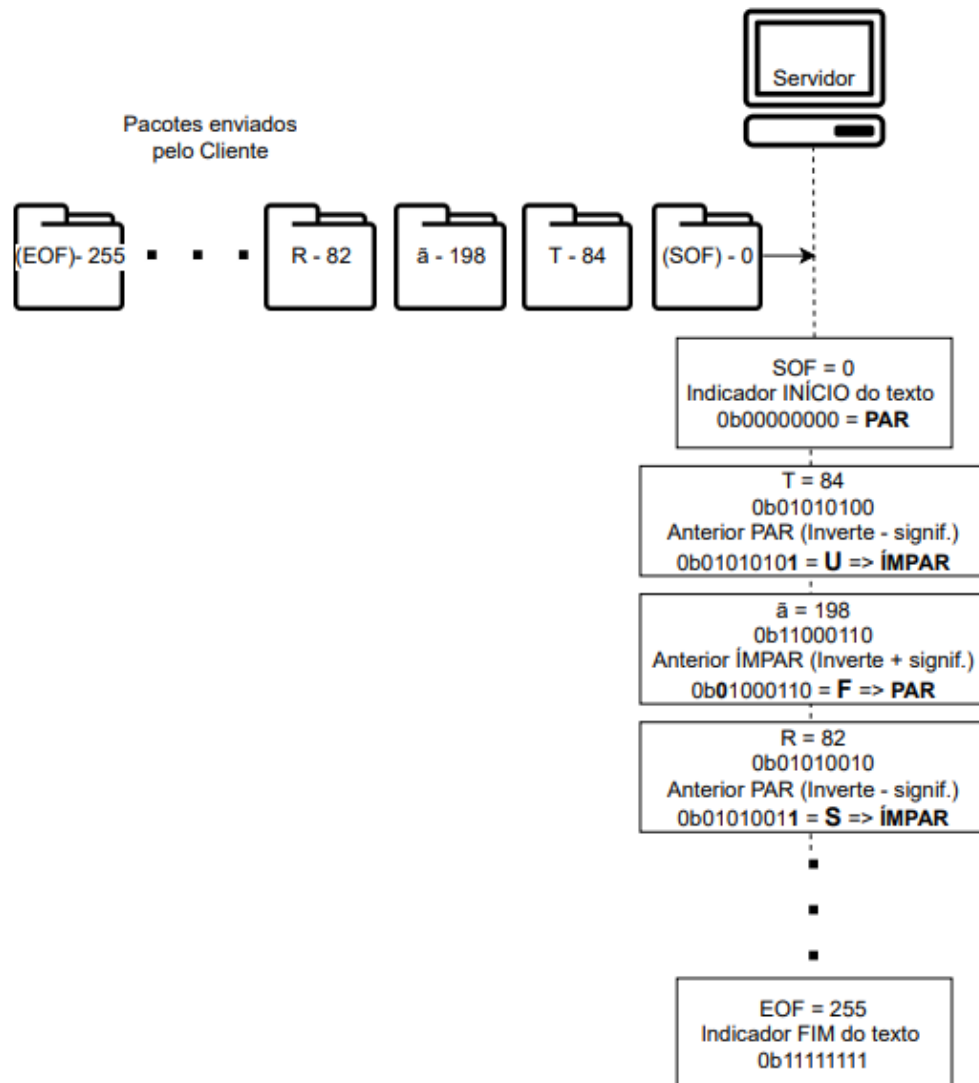
3.1.1. Cada caractere do texto codificado recebido pelo Servidor, enviado do Cliente, contém o "segredo" para decodificar o caractere subsequente (quando recebido em ordem). Ou ainda, o caractere recebido por último precisa do caractere antecessor para ser decodificado.

3.1.2. Os caracteres do texto são elementos de 8 bits (1 byte) da tabela ASCII. Por exemplo o caractere "A" é representado pelo número decimal 65 na tabela ASCII, este elemento é um número ímpar que pode ser representado em binário pela sequência '0b01000001'.

3.1.3. O "segredo" está no caractere ser número PAR ou ÍMPAR. Se for PAR, o próximo elemento recebido pelo Servidor deverá ser invertido o bit menos significativo da sequência binária, se for ÍMPAR, o próximo elemento recebido deverá ser invertido o bit mais significativo da sequência binária. Após esta alteração o caractere modificado estará decodificado e poderá ser analisado se seu valor decimal é PAR ou ÍMPAR para decodificar o próximo elemento recebido no Servidor.

3.1.4. Os textos codificados sempre começam com **SOF** (representado por '0b00000000' ou 0 decimal) e finalizam com **EOF** (representado por '0b11111111' ou 255 decimal). Dessa forma o Servidor e Cliente sabem quando um texto é iniciado ou finalizado na troca de mensagens. **SOF** e **EOF** não precisam ser decodificados.

Por exemplo:



4. Recomendações:

4.1. Utilize como base para implementação das aplicações Cliente e Servidor (TCP e UDP) para troca mensagens os exemplos em Python do livro (Kurose 6ª, cap. 2). Para desenvolvimento do seu protocolo utilize os exemplos dos protocolos **pare e espere** do livro (Kurose 6ª, cap. 3).

4.2. Utilize dois computadores conectados em redes diferentes, um com a aplicação Cliente e outro como Servidor. Para redirecionar uma porta TCP/UDP em seu roteador siga os passos (é necessária esta configuração apenas na rede do Servidor):

<https://www.iperiusbackup.net/pt-br/aprenda-a-redirecionar-portas-para-a-internet/>

4.3. Os arquivos de textos codificados estão no formato UTF-8, utilize esse padrão para abrir o arquivo em sua implementação.

4.4. O arquivo "teste.txt" tem como texto decifrado: **"UFSC À"**. Utilize ele para confirmar se a implementação no Servidor está correta, conforme exemplo **1.4**.

5. Avaliação

5.1. O trabalho deve ser realizado em grupo (máximo 4 participantes).

5.2. Cada grupo deve entregar um relatório escrito com os dados gerados e “*prints*” de tela, se for conveniente, informando os tempos e números de pacotes perdidos em cada teste, conforme intervalos de envio. Relatar os principais motivos que podem ter ocasionado os problemas (caso existam) e as diferenças que ocorreram entre as conexões TCP e UDP com base nos resultados obtidos.

Enviar como foi implementado o protocolo (etapa 2 do trabalho), juntamente com um relatório semelhante à etapa 1 (tempos, pacotes perdidos e retransmissão), utilizando o seu protocolo.

Enviar os códigos das aplicações.

5.3. Data de entrega: **09/04/21**

Links úteis:

Tutorial redirecionamento de porta:

<https://www.iperiusbackup.net/pt-br/aprenda-a-redirecionar-portas-para-a-internet/>

Meu IP externo: <https://www.yougetsignal.com/>

Serviço DNS: <https://www.noip.com/>

elielmarcos@hotmail.com