

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
REDES DE COMPUTADORES

1º semestre de 2018

Professor: Rafael Sachetto Oliveira

Trabalho Prático 3

Data de Entrega: 12 de Dezembro 2018.

Trabalho em Grupo (3 pessoas)

1 O Trabalho

O programador de aplicações para a Internet que opta pelo UDP como seu protocolo de transporte tem um trabalho adicional, se compararmos com o TCP. O UDP, bem como o IP, é um protocolo não-confiável, não-orientado à conexão. Assim o UDP permite que todos os problemas eventualmente surgidos na rede IP se estendam à camada de aplicação. O programador tem que, portanto, projetar sua aplicação para detectar se os dados (1) se perderam; (2) duplicaram; (3) chegaram fora de ordem, e tomar as medidas necessárias para resolver estes problemas.

Neste trabalho, como não teremos TCP para garantir a ordem, vocês serão responsáveis por criar um protocolo confiável sobre UDP usando stop-and-wait. Para isso, vocês devem definir quais serão os campos do cabeçalho do seu pacote (que será enviado dentro do pacote UDP), que encapsulará os dados do arquivo.

Recomendo que os cabeçalhos sejam de tamanho fixo e representados em binário, por ser mais simples e eficiente que usar strings para isso. Mais detalhes sobre o uso de cabeçalhos em binário estão disponíveis na página do curso.

Um grande desafio deste trabalho é o tratamento de temporizações. Isso pode ser feito de diferentes maneiras, usando uma temporização associada ao `recv` (man socket) ou usando alarmes e tratadores de sinais com sockets. A primeira é mais simples, mas não é garantida em ambientes que não o Linux e o FreeBSD (Mac).

Neste Trabalho Prático você vai implementar uma aplicação cliente-servidor sobre UDP. Cada datagrama UDP/IP carrega um número inteiro positivo que identifica a ordem sequencial do datagrama. O identificador do primeiro pacote é escolhido aleatoriamente. O último datagrama tem identificador -1. O cliente tem uma interface que permite que o usuário escolha se quer (1) perder dados; (2) duplicar dados; (3) embaralhar dados. Se a opção é perder ou duplicar dados, deve ser adotada uma estratégia aleatória para escolher quais dados são perdidos ou duplicados. Se a opção é perder os dados, o servidor deve fazer um pedido explícito de retransmissão.

O servidor deve imprimir os dados que recebeu (com dados perdidos, duplicados, fora de ordem) bem como os dados corretos, com todos os problemas resolvidos.

A linguagem de programação permitida é de livre escolha dos alunos. Todos os detalhes, como configuração, mensagens e o que não estiver especificado aqui deve ser definido e documentado no relatório. Interface, organização, criatividade são valorizadas! Capriche!

Avaliação

Deverão ser entregues:

- listagem das rotinas;
- descrição breve dos algoritmos e das estruturas de dados utilizadas;
- análise dos resultados obtidos.

Distribuição dos pontos:

- execução

execução correta: 10%

saída legível: 10%

- estilo de programação

código bem estruturado: 15%

código legível: 15%

- documentação

comentários explicativos: 25%

análise de resultados: 25%