



Roteiro de Aula Prática – Estudo dos protocolos TCP e UDP

DISCIPLINA: DCA0130 – Redes de Computadores
PROFESSOR: Carlos Manuel Dias Viegas

Esta prática consiste em uma introdução aos protocolos TCP e UDP por meio de programação com Sockets.

- Os requisitos para a realização desta prática são a instalação do Python na versão 3 e ter assistido às videoaulas sobre os protocolos TCP e UDP disponibilizadas no SIGAA;
- Esta prática consiste em realizar as tarefas descritas abaixo e responder às questões propostas neste documento;
- Este documento, com as devidas respostas, deverá ser submetido em uma tarefa específica no SIGAA até o dia **03/12/2021**;
- Esta prática deve ser realizada em duplas, podendo ser formadas por alunos de diferentes turmas da disciplina de redes de computadores (DCA0130) do semestre 2021.2.

Os códigos fonte (em Python) necessários para iniciar esta prática estão disponíveis na seguinte página:

<https://www.dca.ufrn.br/~viegas/disciplinas/DCA0130/files/Sockets/>

Nome do discente (1): **Isaac de Lyra Junior** Turma: **02**

Nome do discente (2): **Rodrigo de Lima Santana** Turma: **02**

Tarefa A: Experimento TCP

Faça o download do `clienteTCP.py` e `servidorTCP.py` na página indicada (acima).

1. Execute o Wireshark, inicie a captura e aplique o filtro para TCP na porta 65000: `tcp.port == 65000`
2. Execute o `servidorTCP` e conecte o `clienteTCP` ao mesmo (lembre de editar o IP e as portas conforme necessário para esta tarefa);
3. Envie uma mensagem ao servidor e aguarde a resposta;
4. Pare a captura de pacotes no Wireshark;
5. Analise os pacotes capturados e responda aos itens abaixo:

Responda às questões abaixo:

- a) Quantas mensagens TCP foram “troçadas” entre o cliente e o servidor antes da sua mensagem ser propriamente enviada?

3 mensagens foram troçadas antes de eu enviar uma mensagem ao servidor para realizar a conexão three-way handshake

b) Que mensagens são estas? (falar das *flags/codebits*)

A flag da primeira foi SYN e foi enviada do cliente para o servidor, essa flag significa que o cliente quer estabelecer uma conexão com o servidor, o servidor responde com as flags SYN e ACK, o que significa que o servidor recebeu a mensagem e aceita a conexão, o cliente ao receber a mensagem retorna com ACK o que estabelece a conexão.

c) Qual o nome desse processo de troca de mensagens? E qual a sua finalidade?

O nome desse processo é three-way handshake, a finalidade dele é fazer com que a conexão entre o cliente e o servidor funcione de forma segura, pois a cada troca de mensagem é necessário que seja retornado um ACK de ambos os lados, dessa forma o processo só continua caso tenha uma confirmação de que a mensagem foi entregue.

d) Como ocorreu o encerramento da conexão?

A conexão foi encerrada através do processo four-way handshake, onde cada lado encerra a conexão independentemente, foi primeiramente enviado um FIN e ACK do servidor para o cliente, indicando que o servidor quer finalizar a conexão, o cliente responde com ACK, logo após o cliente envia um FIN e ACK para o servidor, indicando que quer encerrar a conexão e encerra a conexão.

Tarefa B: Experimento UDP

Faça o download do `clienteUDP.py` e `servidorUDP.py` na página indicada (acima).

1. Execute o Wireshark, inicie a captura e aplique filtros para UDP na porta 65000: `udp.port == 65000`
2. Execute o `servidorUDP` e conecte o `clienteUDP` ao mesmo (lembre de editar o IP e as portas conforme necessário para esta tarefa);
3. Envie uma mensagem ao servidor e aguarde a resposta;
4. Pare a captura de pacotes no Wireshark;
5. Analise os pacotes capturados e responda aos itens abaixo:

Responda às questões abaixo:

a) Alguma outra mensagem foi “trocada” entre o cliente e o servidor antes da sua mensagem ser enviada?

Nenhuma outra mensagem foi “trocada” além da própria mensagem enviada para o servidor e a resposta do servidor à esta mensagem (resposta programada no servidor).

b) Quantas mensagens foram efetivamente capturadas?

Duas mensagens foram capturadas, sendo elas a mensagem do cliente para o servidor e a resposta do servidor para o cliente.

c) Houve troca de mensagens para encerramento da conexão?

Nenhuma mensagem de encerramento foi capturada. Ou seja, o servidor continua livre para receber mensagens indefinidamente e o cliente pode se conectar novamente em qualquer momento, sem necessidade de encerramentos.

d) Faça uma breve comparação entre o funcionamento do TCP e do UDP.

O UDP não necessita de qualquer procedimento para iniciar ou finalizar uma conexão, enquanto o TCP necessita, havendo diferenças de complexidade entre ambos. Além disso, o TCP mantém uma conexão sempre segura, retornando uma confirmação toda vez que uma mensagem é enviada corretamente, enquanto que o UDP não verifica e não tem qualquer confirmação se uma mensagem foi corretamente recebida no destino ou não.