Servidor TCP/UDP

G. M. Miranda - 13/0111350
Departamento de Ciência da Computação,
Universidade de Brasília
Email: gabrielmirandatt@gmail.com

***Abstract*—Trabalho computacional envolvendo comunicação entre servidor e cliente TCP e UDP.**

***Index Terms*—socket, TCP, UDP.**

I. Introdução

TCP e UDP são protocolos da camada de transporte do modelo TCP/IP. O primeiro provê serviços de controle de fluxo, controle de congestionamento e ordenamento garantido, enquanto o UDP não provê nenhum destes serviços. O modelo computacional foi programado em python, sendo que para o modelo UDP foi usado o exemplo disponibilizado no moodle como exemplo de servidor em thread, e para o TCP foi abordado o modelo apresentado em [1] e adaptado para se enquadrar ao modelo UDP.

II. Discussão

a) Modelo TCP:

- MultiThreadedTCPServer.py
  Para rodar o servidor TCP, deve-se executar em linha de comando "python MultiThreadedTCPServer.py". O servidor recebe dados de um cliente e retorna a thread aberta devido ao cliente concatenada com os dados recebidos dele em caixa alta.
- MultiThreadedTCPClient.py
  Para fazer uma requisição com o cliente, deve-se executar "python MultiThreadedTCPClient.py <dados_de_envio>", em que dados_de_envio são os dados enviados ao servidor. Um exemplo de comunicação é mostrado nas figuras abaixo.

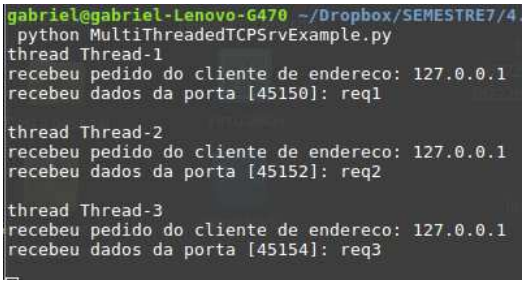

Fig. 1. Lado servidor TCP.

Fig. 2. Lado cliente TCP.

b) Modelo UDP:

- MultiThreadedUDPServer.py
  Para rodar o servidor UDP, deve-se executar em linha de comando "python MultiThreadedUDPServer.py". O servidor recebe dados de um cliente e retorna a thread aberta devido ao cliente concatenada com os dados recebidos dele em caixa alta, assim como no modelo TCP.
- MultiThreadedUDPClient.py
  Para fazer uma requisição com o cliente, deve-se executar "python MultiThreadedUDPClient.py <dados_de_envio>", em que dados_de_envio são os dados enviados ao servidor. Um exemplo de comunicação é mostrado nas figuras abaixo.

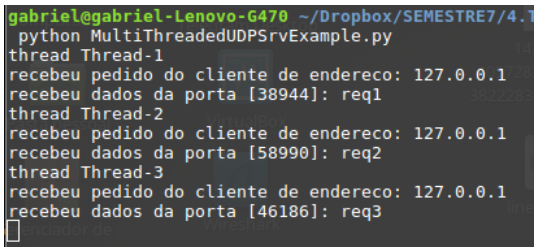

Fig. 3. Lado servidor UDP.

Fig. 4. Lado cliente UDP.

c) Perguntas propostas:

- Em que situações eu devo utilizar (ou é indicado o uso) de threads?

As threads são indicadas nos servidores para permitir que vários clientes possam se comunicador ao servidor no mesmo serviço/processo.

- Faz algum sentido usar threads para o servidor UDP? Justifique sua resposta.
  Depende da situação. O UDP por si só pode conseguir gerenciar requisições de vários clientes, já que não precisa se preocupar com um retorno. Porém em casos onde possa ocorrer atraso na comunicação o uso de threads pode ser aconselhável.

III. Conclusão

Foi possível realizar testes de comunicação usando servidor TCP e servidor UDP. O problema computacional é modelado de forma similar em ambos os casos, sendo necessárias poucas mudanças na plataforma python.

[1] https://docs.python.org/2/library/socketserver.html

[2] http://stackoverflow.com/questions/14041291/problems-implementing-a-multi-threaded-udp-server-threadpool