

# 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 TCP foi usado o exemplo disponibilizado no moodle como exemplo de servidor em thread, e para o UDP foi abordado o modelo apresentado em [1] e adaptado para se enquadrar ao modelo TCP.

## II. DISCUSSÃO

### a) Modelo TCP:

- **MultiThreadedTCPSrvExample.py**  
 Para rodar o servidor TCP, deve-se executar em linha de comando “python MultiThreadedTCPSrvExample.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.
- **MultiThreadedTCPClientExample.py**  
 Para fazer uma requisição com o cliente, deve-se executar “python MultiThreadedTCPClientExample.py <dados\_de\_envio>”, em que dados\_de\_envio são os dados enviados ao servidor. Um exemplo de comunicação é mostrado nas figuras abaixo.

```
gabriel@gabriel-Lenovo-G470 ~/Dropbox/SEMESTRE7/4
python MultiThreadedTCPSrvExample.py
thread Thread-1
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [45150]: req1

thread Thread-2
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [45152]: req2

thread Thread-3
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [45154]: req3
```

Fig. 1. Lado servidor TCP.

```
gabriel@gabriel-Lenovo-G470
./tcp.sh
Enviado: req1
Recebido: THREAD-1: REQ1

Enviado: req2
Recebido: THREAD-2: REQ2

Enviado: req3
Recebido: THREAD-3: REQ3
```

Fig. 2. Lado cliente TCP.

### b) Modelo UDP:

- **MultiThreadedUDPSrvExample.py**  
 Para rodar o servidor UDP, deve-se executar em linha de comando “python MultiThreadedUDPSrvExample.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.
- **MultiThreadedUDPClientExample.py**  
 Para fazer uma requisição com o cliente, deve-se executar “python MultiThreadedUDPClientExample.py <dados\_de\_envio>”, em que dados\_de\_envio são os dados enviados ao servidor. Um exemplo de comunicação é mostrado nas figuras abaixo.

```
gabriel@gabriel-Lenovo-G470 ~/Dropbox/SEMESTRE7/4
python MultiThreadedUDPSrvExample.py
thread Thread-1
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [38944]: req1
thread Thread-2
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [58990]: req2
thread Thread-3
recebeu pedido do cliente de endereco: 127.0.0.1
recebeu dados da porta [46186]: req3
```

Fig. 3. Lado servidor UDP.

```
gabriel@gabriel-Lenovo-G470
./udp.sh
Enviado: req1
Recebido: THREAD-1 REQ1
Enviado: req2
Recebido: THREAD-2 REQ2
Enviado: req3
Recebido: THREAD-3 REQ3
```

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 comunicar ao servidor no mesmo serviço/processo.

- Faz algum sentido usar threads para o servidor UDP? Justifique sua resposta.

Sim. Porém, UDP não é orientado a conexão. Se todos os clientes usarem endereço multicast, não existe forma natural de decidir qual thread deve receber qual pacote. A forma de contornar isto é ler o pacote proveniente do cliente.

### 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>