UNIVERSIDADE FEDERAL DE OURO PRETO – UFOP

CIÊNCIA DA COMPUTAÇÃO

REDES DE COMPUTADORES

TRABALHO PRÁTICO 6

Marcus Vinícius Souza Fernandes

19.1.4046

Ouro Preto

2021

Camada de transporte

Os processos descritos na camada de Transporte do modelo OSI aceitam dados da Camada de Aplicação e os preparam para endereçamento na camada de Rede. A camada de Transporte é responsável pela transferência fim-a-fim geral de dados de aplicação. Ela tem como função habilitar a comunicação de múltiplas aplicações na rede ao mesmo tempo em um único dispositivo e assegura, se necessário, todos os dados sejam recebidos confiavelmente e em ordem pela aplicação correta. Além disso, ela proporciona a segmentação de dados e o controle necessário para reagrupar esses segmentos em fluxos de comunicação.

TCP e UDP

TCP é uma sigla que significa Transmission Control Protocol (Protocolo de Controle de Transmissões, em uma tradução livre), que faz referência ao sistema de envio de pacotes mais comum da internet. Ao acessar um site, seu computador manda dados ao servidor pedindo que ele envie os conteúdos da página à máquina que está sendo utilizada, as informações enviadas de volta são “costuradas” pelo seu navegador para mostrar aquilo que você deseja. Esse processo de envio e recebimento de pacotes se repete toda vez que você clica em um link, faz um login, publica um comentário ou faz basicamente qualquer coisa. A principal característica do TCP é o fato de que ele não somente envia dados como também recebe informações de volta para se assegurar que os pacotes foram recebidos corretamente.

Espécie de “irmão” do protocolo TCP, o UDP (User Datagram Protocol) também se baseia no envio de pacotes de informações, mas remove toda a parte de verificação de erros da outra tecnologia. O objetivo dessa opção é acelerar o processo de envio de dados, visto que todas as etapas de comunicação necessárias para verificar a integridade de um pacote (e para ré enviá-lo, se necessário) contribuem para deixá-lo mais lento. Quando o protocolo UDP é acionado, ele simplesmente manda informações a um destinatário, sem se preocupar se elas foram recebidas devidamente — em caso de erros, simplesmente ocorre o envio do próximo pacote programado pelo sistema, e os anteriores não podem ser recuperados. Embora esse método de funcionamento potencialize a ocorrência de erros, ele garante uma comunicação rápida entre dois computadores.

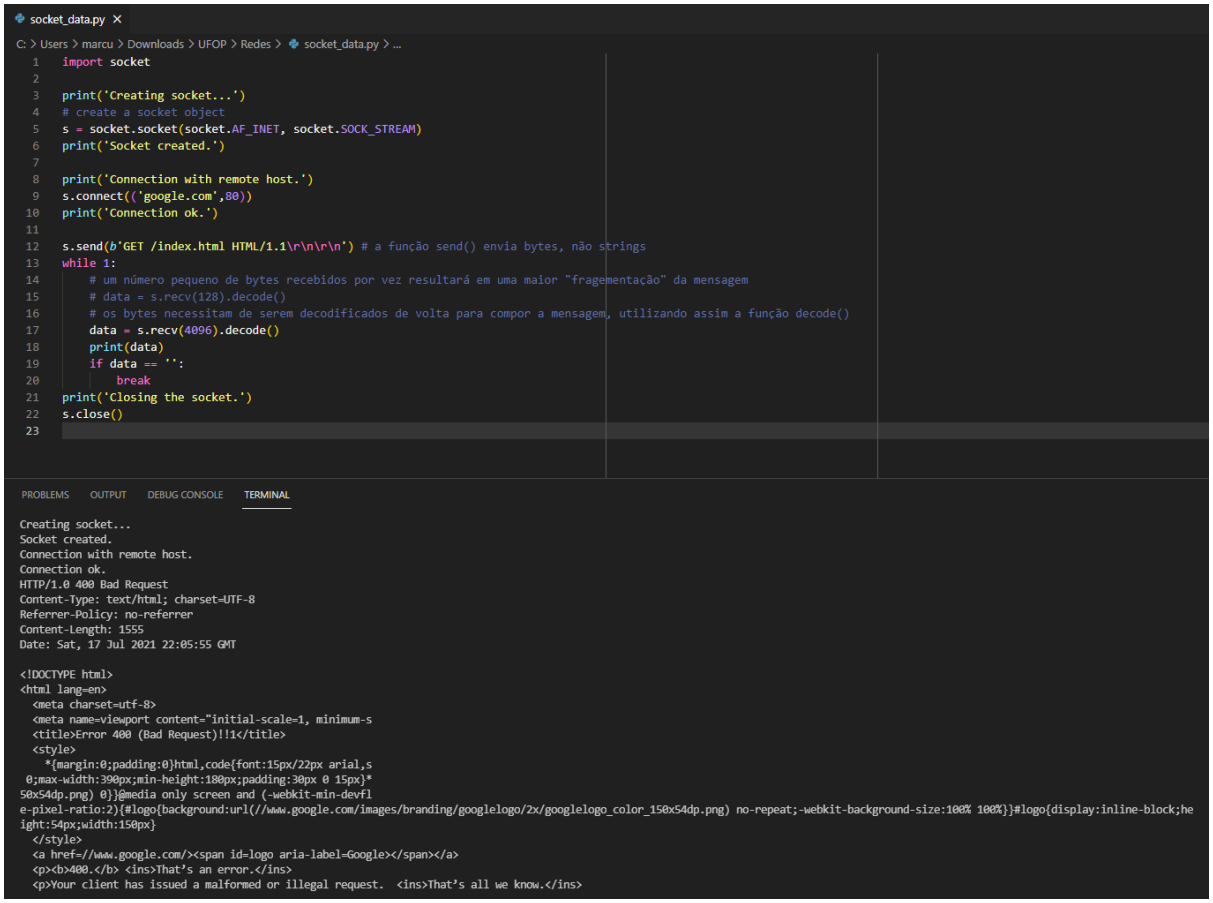
Socket - Biblioteca

Assim como diversas outras linguagens de programação, o Python tem algumas especificações que nos permite trabalhar com sockets. No caso, temos um módulo inteiro, o próprio socket. O módulo socket abriga uma classe essencial, que representa, justamente, um socket. A classe tem o mesmo nome do módulo - socket.

Uso: Importando a *lib* como no exemplo abaixo, você terá acesso a todas as funcionalidades que ela fornece, basta consultar a documentação.

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Código – Exemplo



Dupla para o trabalho

Nome: Gabriel Mace dos Santos Ferreira.

Referências

https://docente.ifrn.edu.br/tadeuferreira/disciplinas/2016.1/arq-tcp-ip/Aula16.pdf

https://www.tecmundo.com.br/internet/57947-internet-diferenca-entre-protocolos-udp-tcp.htm

https://docs.python.org/3/library/socket.html