

UNIVERSIDADE FEDERAL DE OURO PRETO

CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RELATÓRIO TRABALHO PRÁTICO I

Redes de Computadores

GABRIEL FERNANDES NIQUINI – 19.1.4113

Ouro Preto – Minas Gerais – Brasil

2021

1 - Camada de transporte:

Segunda a Wikipedia, A camada de transporte, no modelo TCP/IP, é a camada responsável pela transferência de dados entre duas máquinas, independente da aplicação usada e o do tipo, topologia ou configuração das redes físicas existentes entre elas. A camada de transporte reúne protocolos de transporte *end-to-end* entre máquinas, isto é, ima entidade (software/hardware) que utilize os protocolos desta camada só se comunica com a sua entidade destino, sem comunicação com máquinas intermediárias na rede, como pode ocorrer com as camadas inferiores.

2 – TCP & UDP:

User Datagram Protocol, ou UDP, é um protocolo simples da camada de transporte. Esse protocolo disponibiliza um procedimento para programas mandarem mensagens para outros programas com mecanismo de protocolo mínimo. O protocolo é orientado para transações, e o envio e a proteção não são garantidos.

Transmission Control Protocol, ou TCP, é um dos protocolos de comunicação da camada de transporte, ele, diferente do UDP, verifica se os dados são enviados na sequencia correta e sem erros via rede. Ele funciona como um aperto de mão, ambas as partes tem que estar “cientes do acordo”, o “acordo” acontece e só quando finalizado, as mãos se soltam.

3 – Socket | Socket lib

Sockets permitem a comunicação entre dois processos diferentes na mesma ou em diferentes máquinas. Para ser mais preciso, é uma forma de comunicar com outros computadores usando de “arquivos” descritores de padrão Unix. No Unix, toda ação de entrada/saída é feita escrevendo ou lendo um desses “arquivos”. Esse “arquivo” é apenas um inteiro associado a um arquivo aberto e ele pode ser uma conexão de rede, um arquivo texto, um terminal ou alguma outra coisa.

Ele, o Socket Unix, é usado em um framework de aplicação cliente-servidor.

4.1 – Programa:

```
1  import socket
2  print('creating socket...')
3  #create a socket object
4
5  client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6  print('socket created')
7  print("connection with remote host")
8
9  client.connect(('google.com', 80))
10
11 print ('conection ok')
12
13 client.send('GET / index.html HTTP/1.1\r\n\r\n')
14
15 while 1:
16     data = client.recv(128)
```

4.2 – Erro:

```
creating socket...
socket created
connection with remote host
connection ok
Traceback (most recent call last):
  File "d:/Aulas EAD/REDES/tp6/tp6.py", line 13, in <module>
    client.send('GEI / index.html HTML/1.1\r\n\r\n')
TypeError: a bytes-like object is required, not 'str'
```

4.3 – Solução para problema:

```
1  import socket
2  print('creating socket...')
3  #create a socket object
4
5  client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6  print('socket created')
7  print("connection with remote host")
8
9  client.connect(('google.com', 80))
10
11 print('connection ok')
12
13 client.send(b'GEI /index.html HTML/1.1\r\n\r\n')
14
15 while 1:
16     data = client.recv(128).decode('utf-8')
17     print(data)
18     if data == "":
19         break
20 print('closing socket...')
21 client.close()
22 print('socket closed')
```

4.5 – Saída:

```
PS D:\Aulas EAD\REDES\tp6> & D:/Python-Conda/python.exe "d:/Aulas EAD/REDES/tp6/tp6.py"
creating socket...
socket created
connection with remote host
connection ok
HTTP/1.0 400 Bad Request
Content-Type: text/html; charset=UTF-8
Referrer-Policy: no-referrer
Content-Length: 1555
Date: Tue,
20 Jul 2021 01:50:14 GMT

<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content="initial-scale=
1, minimum-scale=1, width=device-width">
  <title>Error 400 (Bad Request)!!1</title>
  <style>
    *{margin:0;padding:0}html,code{font:15px/22px arial,sans-serif}html{background:#fff;color:#222;padding:15px}body{margin:7% auto 0;max-width:390px;min-height:180px;padding:30px 0 15px}* > body{background:url(//www.google.com/images/errors/robot.png) 100% 5px no-repeat;padding-right:205px}p{margin:11px 0 22px;overflow:hidden}ins{color:#777;text-decoration:none}a img{border:0}@media screen and (max-width:772px){body{background:none;margin-top:0;max-width:none;padding-right:0}}#logo{background:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat;margin-left:-5px}@media only screen and (min-resolution:192dpi){#logo{background:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat 0% 0%/100% 100%;-moz-border-image:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) 0}}@media only screen and (-webkit-min-device-pixel-ratio:2){#logo{background:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat;-webkit-border-image:url(//www.google.com/images/branding/googlelogo/2x/googlelogo_color_150x54dp.png) no-repeat;}}#logo{display:inline-block;height:54px;width:150px}
  </style>
  <a href=//www.google.com/><span id=
logo aria-label=Google></span></a>
  <p><b>400.</b> <ins>That's an error.</ins>
  <p>Your client has issued a malformed or illegal request. <ins>That's all we know.</ins>

closing socket...
socket closed
PS D:\Aulas EAD\REDES\tp6>
```

4.4 – Outro exemplo de conexão:

```
1  import socket
2  import sys
3  print('creating socket...')
4  #create a socket object
5
6  try:
7      client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8      print('Socket created')
9  except socket.error as err:
10     print('Socket creation failed with error %s ' %(err))
11
12     port = 80
13
14     try:
15         host_ip = socket.gethostbyname('www.google.com')
16     except socket.gaierror:
17         print('There was an error resolving the host')
18         sys.exit()
19
20     client.connect((host_ip,port))
21
22     print("the socket has successfully connected to google")
```

```
creating socket...
Socket created
the socket has successfully connected to google
```

Referencias:

<https://www.geeksforgeeks.org/socket-programming-python/>

https://pt.wikipedia.org/wiki/Transmission_Control_Protocol

https://pt.wikipedia.org/wiki/User_Datagram_Protocol

https://pt.wikipedia.org/wiki/Camada_de_transporte

https://www.tutorialspoint.com/unix_sockets/what_is_socket.htm