

## UFOP-DECOM-BCC361 (Redes) Nº 06/2020-2

6º TP 2020-2 (para 15-07-2021)

Estamos caminhando em nossa jornada no mundo do conhecimento de redes de computadores. Nesta etapa, você já deve viu coisas muito interessantes e estamos agora na camada de transporte.

A camada de transporte permite que vários “programas” utilizem um host IP. Isto é, um host IP (tipo ufop.br) possa ter vários serviços. Boa parte dos serviços que usamos “normalmente” (tipo http/ https / dns) estão em portas pré-definidas. Entender quais são as portas “Bem Conhecidas” (*Well Known*), “Registradas” (*Registered*) e “Dinâmicas” (*Dynamic*) são importantes quando configuramos sistemas como UNIX/LINUX. Veja essa página da Red Hat Enterprise Linux 4 [http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-pt\\_br-4/ch-ports.html](http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-pt_br-4/ch-ports.html) Mas, na realidade, todas as definições destas portas estão na RFC6335 <https://datatracker.ietf.org/doc/html/rfc6335> Lembra-se que eu disse que as RFCs da IETF são as que normas que definem realmente a Internet? Pois é, graças ao IETF e a Internet realmente é interoperável e livre de normas proprietárias de fabricantes.

Disponibilizei os capítulos 1,2,e 3 do livro MASTERING\_PYTHON FOR NETWORKING AND SECURITY de José Ortega. Exemplos e códigos fonte estão disponíveis no repositório GitHub <https://github.com/PacktPublishing/Mastering-Python-for-Networking-and-Security> . Observe que estes exemplos dizem respeito ao Python versão 2.7 . Use o Python versão 3.x, ok? Vai dar alguns problemas de sintaxe, mas nada que seja difícil de resolver.

**Este trabalho é individual.** O que tem que fazer?

- 1- Instalar o Python 3.x
- 2- Rodar o programa socket\_data.py na página 68 em python 3.x
- 3- Fazer um texto em PDF (veja abaixo)
- 4- Escolher seu grupo (grupo de, no máximo 2 elementos) para que eu configure o Moodle para o Próximo TP (veja abaixo)

You can find the following code in the socket\_data.py file:

```
import socket
print 'creating socket ...'
# create a socket object
```

[ 68 ]

Socket Programming

Chapter 3

```
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'socket created'
print "connection with remote host"
s.connect(('www.google.com',80))
print 'connection ok'
s.send('GET /index.html HTTP/1.1\r\n\r\n')
while 1:
    data=s.recv(128)
```

Entregáveis:

- a. Um texto em formato PDF explicando o que é **1)** camada de transporte, **2)** TCP e UDP, **3)** biblioteca sockets (para que serve tmb) **4)** o programa acima e **5)** quem será seu companheiro de grupo de trabalho em redes para os próximos TPs em grupo (este TP é individual). Não se esqueça da bibliografia, , ok?
- 1) POR FAVOR, NÃO ZIPE e atente para o que foi dito anteriormente sobre a entrega de vídeos como trabalho e o PRAZO;
- 2) Como falado anteriormente, as respostas e exposição **do que foi solicitado** (entenda, antes de começar, o que foi solicitado). Neste TP, **NÃO** mande vídeo ou código em separado.. ou qualquer outra coisa fora do ARQUIVO pdf.
- 3) As respostas e a exposição devem ser técnicas, corretas e completas.
- 4) Não coloque nenhum link nos COMENTÁRIOS, pf.

Bom trabalho.

© 2021, Prof. Dr. Carlos Frederico M.C. Cavalcanti  
DECOM/ICEB/UFOP