

Trabalho 1 de Redes

Sumário

- Descrição
- Tecnologias a serem usadas
- Modelos

Descrição

- Vamos criar 2 protocolos para troca de mensagens
- Vamos abrir conexões da Camada de transporte usando Sockets
- 1 protocolo deve usar o modelo cliente servidor
- 1 protocolo deve ser Peer to Peer
- O protocolo deve armazenar todas as mensagens de uma sessão em um arquivo de texto ao final
- OS PROTOCOLOS SÃO DA TURMA - CASO 1 GRUPO NÃO SE COMUNIQUE ISSO AFETA A TURMA COMO UM TODO.

Descrição - entregáveis

- Relatório usando o template de artigo da sociedade brasileira de computação
- Até 6 páginas
- Descrevendo o protocolo proposto em termos de
 - Troca de mensagens - sintaxe do protocolo e justificando as escolhas de design
 - Requisitos de tráfego; ex: sensível a atraso? A perda?
- Mostrando um trace do wireshark com o protocolo operando na rede real
- Link do github para o código proposto
- Demo com a turma toda no dia 16/10.

Tecnologias a serem usadas

- Python 3.
- Biblioteca Socktes
- Wireshark - <https://www.wireshark.org/>
- Template da SBC - <https://www.sbc.org.br/documentos-da-sbc/category/169-templates-para-artigos-e-capitulos-de-livros>
- Exemplos com sockets: <https://youtu.be/6sHGBXwkFQU>

Modelos - python cliente

```
1. import socket
2.
3. client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4. client.connect(('0.0.0.0', 8080))
5.
6. client.send("I am CLIENT\n")
7.
8. from_server = client.recv(4096)
9.
10. client.close()
11.
12. print from_server
```

Modelo - Python servidor

```
1. import socket
2.
3. serv = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4.
5. serv.bind(('0.0.0.0', 8080))
6. serv.listen(5)
7.
8. while True:
9.     conn, addr = serv.accept()
10.    from_client = ''
11.
12.    while True:
13.        data = conn.recv(4096)
14.        if not data: break
15.        from_client += data
16.        print from_client
17.
18.        conn.send("I am SERVER\n")
19.
20.    conn.close()
21.    print 'client disconnected'
```