

Projeto 3 - Fragmentação, hand-shake e datagrama

Links para os vídeos:

Caso 1:

<https://drive.google.com/file/d/1ekg4WdwuPCsngoJskJeJgdxTBn0TJw2H/view?usp=sharing>

Caso 2:

<https://drive.google.com/file/d/1UOhO9bQBUQsmNOcFevXSUtYNEiDR6iA/view?usp=sharing>

Caso 3:

<https://drive.google.com/file/d/1PyVy7EgKUCHPOvw5fhzdHu5kNLvszg/view?usp=sharing>

Caso 4:

<https://drive.google.com/file/d/1ugxyL43Q43BXG31PfdZP6M6JTaYKdRFv/view?usp=sharing>

Header:

0	Tipo de Mensagem
1	Tamanho do Payload (de 0 a 114)
2	Número do pacote
3	Quantidade total de pacotes
4	Qual pacote deu erro (ex: erro no 15)
5	Qual pacote deve ser reenviado (então reenvia o 14)
6	0
7	0
8	0
9	0

➤ Tipo de Mensagem:

0	HandShake (pode enviar, Server?)
1	Resposta do Server (sim, pode enviar estou pronto)
2	Mensagem de dado (payload ≠ 0)
3	Server: pacote está correto
4	Server: erro no numero do pacote
5	Server: erro no payload
6	Término de envio de pacotes
7	Resposta do Server ao handshake(não pode enviar)
8	Server: terminou deu tudo certo

End Of Packed = 1, 1, 1, 1

Link do git do projeto: <https://github.com/LaisNSilva/Camadas-P3.git>

Para esse projeto eu comecei criando 2 classes, uma para o cliente e uma para o servidor. Na classe do cliente foram feitas várias funções para serem usadas na aplicação, como uma para enviar o handshake, uma para analisar quantos pacotes no total o dado irá precisar para ser enviado, uma para preparar os pacotes, também para analisar as mensagens do server. Na classe do server também existem várias funções específicas como para responder o handshake, para enviar mensagens erro/verificação. Nas aplicações eu fui chamando as funções de cada classe e fui verificando tudo o que estava ocorrendo durante a transmissão, observando cada resposta do server para o cliente (enviada a cada pacote) e no server observando se o pacote estava certo.

Para gerar os erros os quais foram apresentados nos vídeos “caso 3” e caso 4” os erros foram inseridos (simulados via código):

```

else:
    # VERIFICA SE PODE ENVIAR
    #print("entrei no else")
    if correcao != 2:
        v, qual_erro, qual_reenviar = com1.verifica_envia(pacote_verificacao)
    if v== True:
        print("vou enviar o pacote {}".format(numero_pac))
        if correcao ==0:
            numero_pac+=1
        correcao = 1
        header_pacote, pay_end_pacote= com1.prepara_dado(txBuffer, numero_pac, byte_dado, qtd_pac, ultimo)
        print("preparou pacotes")
        time.sleep(2)
        com1.envia_pacote(header_pacote)
        print("enviou")

```

Figura 1: Erro no número do pacote

```

else:
    # VERIFICA SE PODE ENVIAR
    #print("entrei no else")
    if correcao != 2:
        v, qual_erro, qual_reenviar = com1.verifica_envia(pacote_verificacao)
    if v== True:
        print("vou enviar o pacote {}".format(numero_pac))
        if correcao ==0:
            #numero_pac+=1
            EOP = int_byte(0) + int_byte(1)+ int_byte(1) +int_byte(1)
        correcao = 1
        header_pacote, pay_end_pacote= com1.prepara_dado(txBuffer, numero_pac, byte_dado, qtd_pac, ultimo, EOP)
        print("preparou pacotes")
        time.sleep(2)
        com1.envia_pacote(header_pacote)
        print("enviou")
        time.sleep(0.3)

```

Figura 2: Erro no tamanho do payload