Insper

Camada Física - Dicas Projeto 2 - COM-Datagrama

Rafael Corsi - rafael.corsi@insper.edu.br

2017

Dicas de implementação :

A seguir algumas dicas para a implementação do projeto 2.

Ordem de execução

1. Defina o protocolo (head e EOP)
   - documente para que ambos possam trabalhar na mesma ideia
2. Estude como o enalceRx e enlaceTx fazem o recebimento e transmissão de dados.
3. Faça primeiramente o encapsulamento dos dados no pacote
4. Com o encapsulamento feito, pense no desencapsulamento.
5. Implemente o desencapsulamento
6. Faça a validação

Server

O server utilizava uma chamada a chamada de função *getData(size)* porém agora não é mais necessário passar a quantidade de bytes a ser recebido para a camada enlace. Atualize a função para não ser necessário mais essa informação : *rxBuffer = getData()*

DEBUG TX , modifique o enlaceTx.py

Modifique o método thread(self) do enlaceTx para imprimir na tela os dados que estão sendo enviados :

```
    def thread(self):
        """ tx thread, to send data in parallel with the code
        """
        while not self.threadstop:
            if(self.threadmutex):
                print(self.buffer)
                self.translen    = self.fisica.write(self.buffer)
                self.threadmutex = false
```

DEBUG RX , modifique o enlaceRx.py

Modifique o método thread(self) do enalceRx para imprimir na tela os dados que estão sendo recebidos :

```
    def thread(self):
        """ RX thread, to send data in parallel with the code
        """
        while not self.threadStop:
            if(self.threadMutex == True):
                rxTemp, nRx = self.fisica.read(self.READLEN)
                if (nRx > 0):
                    self.buffer += rxTemp
                print(self.buffer)
                time.sleep(0.001)
```

enalce.py , enalceRx.py, enalceTx.py

Agora o enlace será responsável pela manipulação dos dados, deverão analisar a fundo como o enalceRx faz o recebimento dos dados e como o enalceTx faz o envio para então poderem propor o empacotamento dos dados. Uma boa dica é começar pelo enlaceTx.py

HEAD e EOP

Utilizar o pacote construct para criar o HEAD e o EOP. A seguir um exemplo de um HEAD simples com o construct :

```
headSTART  = 0xFF
headStruct = Struct("start" / Int8ub,
                    "size"  / Int16ub )


def buildHead(self, dataLen):
```

```
    head = headStruct.build(dict(
                                start = self.headSTART,
                                size  = dataLen))
    return(head)
```

Esse pacote permite termos controle de quantos bytes serão utilizados para cada parte parte da mensagem.