Redes I
Alunos: Gustavo Ceconelli, Túlio Bittar, Marcelo Candido, Pedro Belisário
CEFET-MG
Data: 06/09/2019

Documentação da camada física

A camada física foi implementada em python, e consta com as classes: client.py, server.py, frame.py e utils.

O projeto segue o seguinte fluxo:

1 - O servidor fica escutando em uma porta, esperando por uma conexão do cliente;
2 - O cliente recebe a mensagem a ser enviada do usuário, juntamente com o ip de destino, monta um frame em modo de leitura humana, salva, converte para binário (através da função string2bits) e também o salva;
3 - O frame convertido é enviado para o servidor através da conexão aberta previamente. O cliente fecha o seu envio de mensagens (para evitar um *bug* de razão desconhecida);
4 - O servidor lê o frame recebido e o salva em um arquivo;
5 - O frame salvo é lido e desconvertido para o modo de leitura humana (através da função bits2string). A mensagem desconvertida é salva em um arquivo;
6 - O servidor então envia uma mensagem (‘Recebido!’) para o cliente através da conexão aberta. Essa mensagem é previamente salva num frame (arquivo) em modo de leitura humana, depois é convertida para binário e salva em outro arquivo, então é enviada ao cliente.
7 - O cliente recebe a mensagem binária do servidor, escreve a mensagem num arquivo em modo de leitura humana, desconverte-a, salva em outro arquivo e exibe para o usuário.

As mensagens enviadas e recebidas são armazenadas em arquivos nas pastas *sent* e *received*, respectivamente, de cada processo (cliente e servidor). O arquivo em formato de leitura humana tem por nome *frame.txt* e o em formato binário é chamado por *bin-frame.txt*.

API Documentation

API Documentation

September 6, 2019

Contents

1 Module client

1.1 Variables

Name	Description
___package___	**Value:** `None`

1.2 Class client

object
└
client.client

Defines client class

1.2.1 Methods

___init___(*self*)

Defines default constructor for client class

Overrides: object.___init___

receiveFromAbove(*self*, *connectionSocket*, *ipAddr*, *message*)

Receives message and destination ip from upper layers, put it into a frame, save this frame and convert it to binary-mode. The converted frame is also saved and fowarded to sendToServer method.

receiveFromServer(*self*, *connection*)

Receives binary frame from server and sends it to upper layers

sendToAbove(*self*, *cvtdPDUPath*)

Sends given frame (which path is pointed by cvtdPDUPath) to upper layers

sendToServer(*self*, *connection*, *payloadSize*, *cvtdPDUFile*)

Sends to server addressed by connection the converted PDU File cvtdPDUFile with payload size of payloadSize

connectToServer(*self*, *ipAddr*, *port*)

Returns socket connection with given ip in the given port

Inherited from object

___delattr___(), ___format___(), ___getattribute___(), ___hash___(), ___new___(), ___reduce___(), ___reduce_ex___(), ___repr___(), ___setattr___(), ___sizeof___(),

___str___(), ___subclasshook___()

1.2.2 Properties

Name	Description
Inherited from object	
___class___	

2 Module frame

2.1 Variables

Name	Description
___package___	**Value:** `None`

2.2 Class frame

object
└ **frame.frame**

Defines a Ethernet frame with MAC addresses for destine, origin, payload's size and payload

2.2.1 Methods

___init___(*self*, *destMACAddr*, *origMACAddr*, *message*, *payloadSize*=0, *payloadStrSize*='', *frameType*='')

Defines frame's default construct

Overrides: object.___init___

readPDU(*self*, *pathDesired*)

Reads file indicated by pathDesired and populates the frame object with data obtained from the file

writePDU(*self*, *pathDesired*)

Writes original or converted PDU to disk in the pathDesired location

convertPDU(*self*)

Returns frame object in simulated binary way

deconvertPDU(*self*)

Converts frame from binary to human readable format

Inherited from object

___delattr___(), ___format___(), ___getattribute___(), ___hash___(), ___new___(), ___reduce___(), ___reduce_ex___(), ___repr___(), ___setattr___(), ___sizeof___(),

___str___(), ___subclasshook___()

2.2.2 Properties

Name	Description
Inherited from object	
___class___	

2.2.3 Class Variables

Name	Description
HEADER_SIZE	**Value:** `112`
MAX_PAYLOAD_SIZE	**Value:** `1500`

3 Module server

3.1 Variables

Name	Description
__package__	**Value:** `None`

3.2 Class server

Defines a server class

3.2.1 Methods

__init__(*self*)

Defines server's default constructor

Overrides: object.__init__

receiveFromAbove(*self*, *connectionSocket*, *ipAddr*, *message*)

Receives message from upper layer, saves it as human readable and binary frames and send it to client

receiveFromClient(*self*, *clientSocket*)

Returns the frame data received from client as file

sendToAbove(*self*, *cvtdPDUPath*)

Sends message received from client to the upper layer

sendToClient(*self*, *connection*, *payloadSize*, *cvtdPDUFile*)

Sends binary frame to client

run(*self*)

Turns server's socket binded to hostname and PORT and that listens to up to one connections

Inherited from object

___delattr___(), ___format___(), ___getattribute___(), ___hash___(), ___new___(), ___reduce___(), ___reduce_ex___(), ___repr___(), ___setattr___(), ___sizeof___(), ___str___(), ___subclasshook___()

3.2.2 Properties

Name	Description
Inherited from object	
___class___	

4 Module util

4.1 Functions

ping(*ipAddr*)

Returns if ping command was successful for given ipAddr

genMACAddr(*ipAddr*)

Tests if the given ipAddr is on the local network and returns the MAC Address of the respective device

string2bits(*s*=`''`, *fill*=`8`)

Converts given string to the respective ascii-binary version (using 8 bits)

bits2string(*b*=`None`)

Converts given ascii-binary string to the respective ascii version (using 8 bits)

calcCollisionProb(*down*, *up*)

Return if there is a collision for given interval

4.2 Variables

Name	Description
___package___	**Value:** `None`

Index