LAB 3 - TASK 2 (EXTERNAL RESOURCE) (1.0 points possible)

Lab 3 - Task 2

In this task, you will learn about the process of encapsulation. The transport layer takes messages application layer, divides them into segments and adds header information to the segments. In this mainly concerned with the problem of reliable data transfer, so we will use a simplified header that information about the sequence number of the segment and the total number of segments that con message. If we were concerned about multiplexing/demultiplexing, we would also include informatio source/destination ports.

INSTRUCTION

The MATLAB code in the window below is similar to the code in Task 1, but you will implement the f **packetize()**. As shown in Task 1, this function converts the message bit sequence (**tx_bs**) into a li

In this case, the application wishes to send the text message 'PACKET', which is contained in the va **tx_msg**. It first generates the bit sequence **tx_bs** from the text message **tx_msg** using the functio **text2bitseq()**, which we studied in Part 1. Each character in the text message is represented using ASCII code. Thus, if the text message has **n** characters, then **tx_bs** will be **n*8** bits long.

The function **packetize()** from Task 1 converts **tx_bs** to a list of **n** packets, where each packet car character from the message. In this task, your job is to implement this function.

Each packet contains a 16 bit header and 8 bits of data. Thus, each packet is 24 bits long. The 16 consists of an 8 bit sequence number and an 8 bit number encoding the number of packets in the The sequence numbers run from 1 to **n**. The 8 bit data is taken from the bit sequence **tx_bs**. The contains the first 8 bits of **tx_bs**, the second packet contains the second 8 bits, and so on. For exa the first 8 bits of tx_bs encode the letter 'P' and there are n = 6 characters in the text message, the will be

[0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0]

where extra spaces have been inserted in between sets of four bits to improve readability. The ASC 'P' is 80 in decimal.

Your task is to create the list of packets for the given message and save it inside the variable

send_packet_list, which is an **n**-by-24 element matrix where each of the **n** rows contains one pac row contains the first packet and so on. In the initial code, **send_packet_list** contains only one pa data is the ASCII code for 'P' in binary, and the message from the application layer was assumed tc one character.

In order to convert the index of the packet and the total number of packets into binary vector, you r function **dec2binvec**. This function takes as input a vector of **k** decimal numbers and returns a **k** b where each row is the 8 bit binary representation of one element of the input vector. Note that the s of the packet, the total number of packets in the list, should be the same for all packets.

The remainder of the code is the same as that in task 1. Revise the code between the lines

```
% % % % Revise the following code   % % % %
```

and

```
% % % % Do not change the code below % % % %
```

Please do not change other parts of the code.

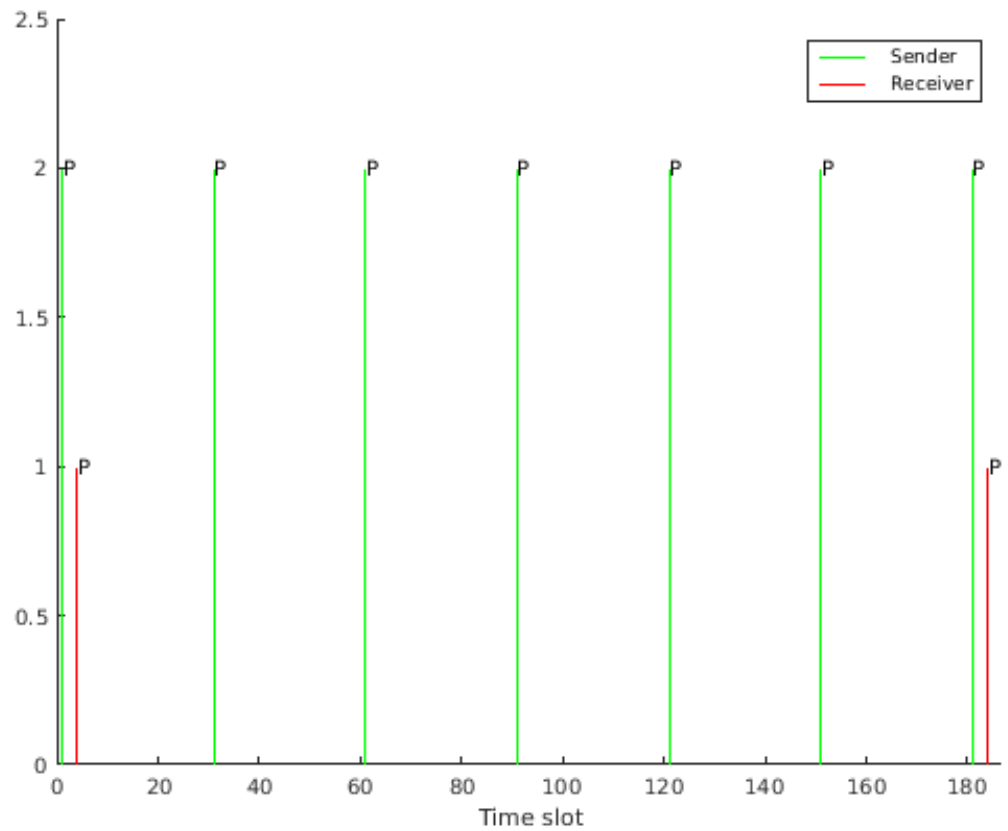
Your Solution

Reset MATLAB Documentation (https://www.mathworks.

```
display (['Numb. of packets sent correctly: ' num2str(noSentPackets )])
% todo: manage duplicated and/or unordered data
rx_msg = reconstruct_msg(receiver_packet_list)

% plot the traffic in the network
if noSentPackets>0
    transmission_display();
end
```

Output

```
Simulation Time: 187
Numb. of packets sent correctly: 1

rx_msg =

PP
```



Assessment Tests