

**HKUSTx: ELEC1200.3x A System View of Communications: From Signals to Packets (Part 3)**

Bookmarks

- ▶ Pre-course Materials
- ▶ Topic 1: Course Overview
- ▶ Topic 2: The Link Layer
- ▶ Topic 3: The Network Layer
- ▶ Topic 4: Routing
- ▶ Topic 5: The Transport Layer
- ▼ **Topic 6: Reliable Transfer Protocols**


Topic 6: Reliable Transfer Protocols > 6.4 Lab 3: Transport Layer > Lab 3 - Task 4




Bookmark

LAB 3 - TASK 4 (EXTERNAL RESOURCE) (1.0 points possible)


6.1 Stop-and-Wait Protocol

Week 3 Quiz due Feb 15, 2016 at 15:30 UTC 


6.2 Throughput of Stop-and-Wait

Week 3 Quiz due Feb 15, 2016 at 15:30 UTC 

6.3 Sliding Window Protocol

Week 3 Quiz due Feb 15, 2016 at 15:30 UTC 

6.4 Lab 3: Transport Layer

Lab due Feb 15, 2016 at 15:30 UTC 

► MATLAB download and tutorials

Lab 3 - Task 4

In this task, you will implement the sender function of the stop-and-wait protocol.

INSTRUCTIONS

The MATLAB code in the window below is similar to that in Task 1, but you will implement the function **sender_stopwait()**. As described in Task 1, this function sends the packets contained in the variable **send_packet_list** to the receiver using the stop-and-wait protocol.

In order to implement this protocol, the sender maintains several variables:

1. **num_packets** stores the total number of packets to be sent
2. **send_packet_num** indicates the sequence number of the packet that is to be sent or be acknowledged
3. if an acknowledgment has been received, **ack_num** indicates the sequence number of that has been acknowledged
4. **send** is a boolean flag indicating whether to send a packet or not.

In each iteration, the sender should first check whether to send a packet or not. There are two conditions that may trigger a transmission/retransmission:

(1) The timeout expires while the sender is waiting for the acknowledgement of the current packet back. The sender checks whether the timeout has expired using the command **timeoutExpired()**, true if the timeout has expired, and false otherwise. If the timeout has expired, the sender should reset packet number **send_packet_num**, by setting **send** equal to true.

(2) An acknowledgement comes back. The sender checks for an acknowledgement using the command **send_ack = sender_get_ack()**. If an acknowledgement has arrived, then **send_ack** contains the acknowledgement packet, otherwise it is empty. The acknowledgement packet contains the 8 bit sequence number of the acknowledged packet. This can be converted into a decimal sequence number **ack_num** using the command **binvec2dec(send_ack)**. If the acknowledgement packet's sequence number (**ack_num**) is the same as the sequence number of the acknowledgement that the sender is waiting for (**send_packet_num**), then the sender should increment **send_packet_num** to indicate that the next packet should be sent. If **send** is false, then the sender should set **send** equal to true to trigger a send. However, if **ack_num** is equal to **num_packets**, then the sender has sent all more packets to send, and the sender should trigger the simulation to stop by setting **run** equal to false.

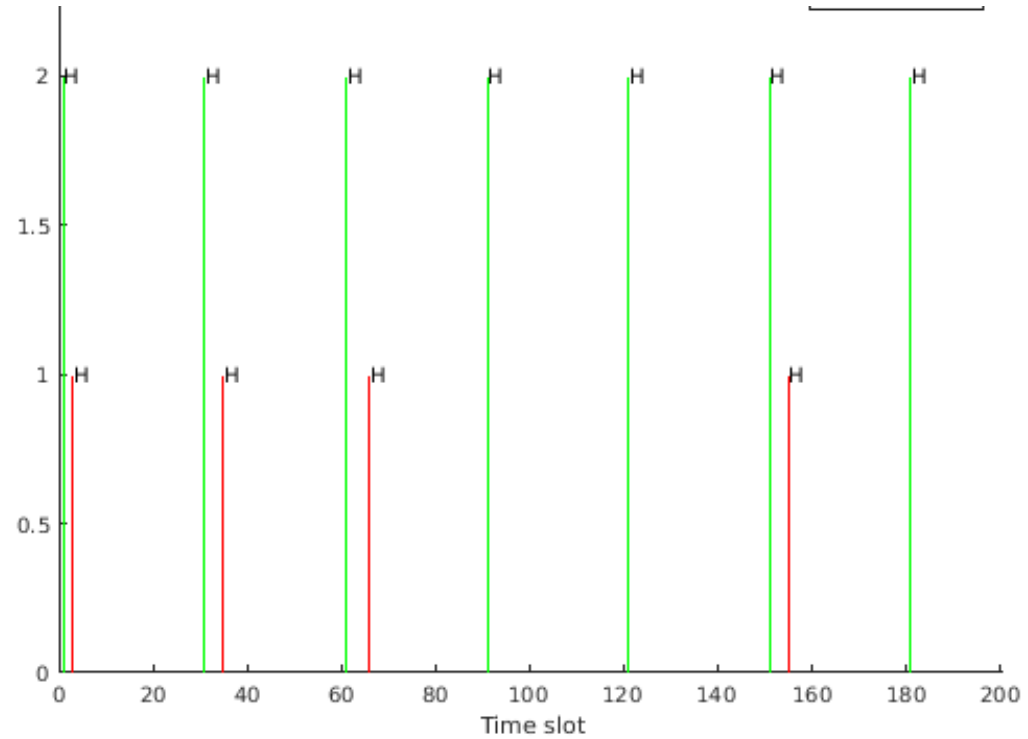
If a packet should be sent (**send** = true), then the sender should get packet number **send_packet_num** from the matrix **send_packet_list** and transmit it using the command **sender_send_packet(packet)**. Then reset the timeout timer to wait for **time_out** iterations with the command **resetTimeout(time_out)**.

Your task is to revise the code to implement the sender. Specifically, you will implement the code to

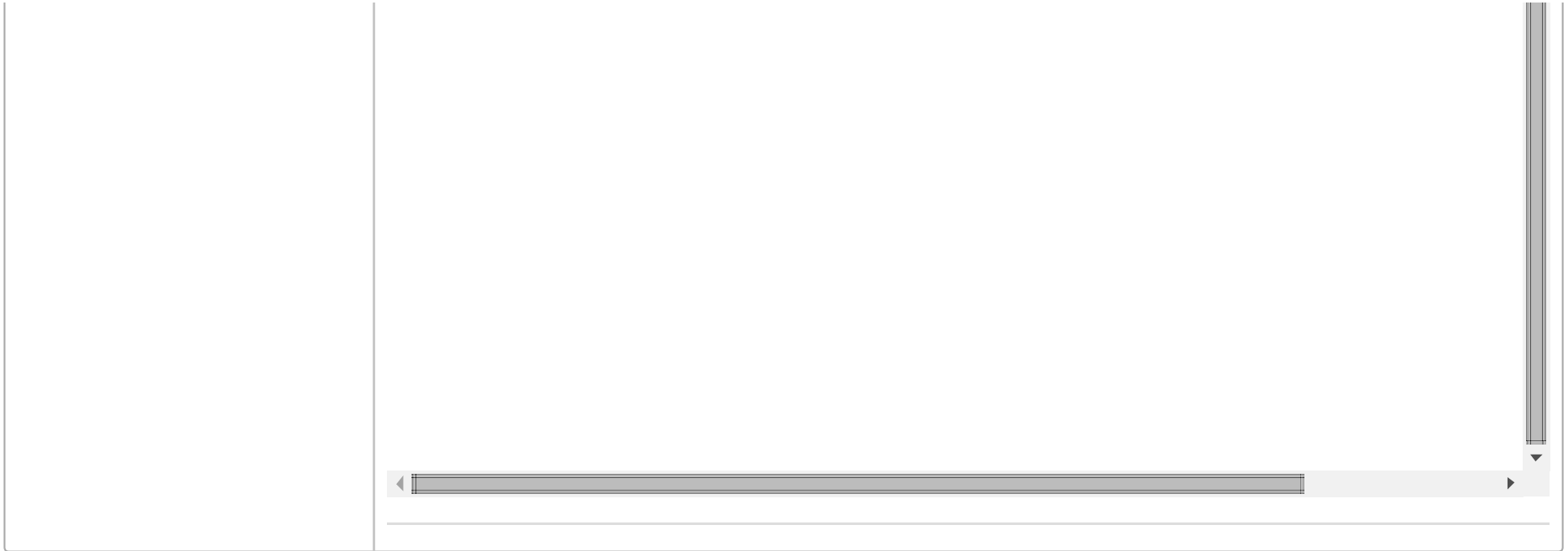
Please do not change other parts of the code.

 MATLAB Documentation (<https://www.mathworks.com/help/optim/ug/optimization-parameters.html>)

— Sender
— Receiver



Assessment Tests



© All Rights Reserved



© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX



