|  |  |  |  |
| --- | --- | --- | --- |
| BN | Section | Student ID | اسم الطالب |
| **18** | **2** | **9220354** | **ساندى نجيب لبيب نجيب** |
| **13** | **4** | **9220869** | **منار كمال مرسي أحمد** |

# System Design

## Overview of the System

## Our system consists of 3 sender tasks and one receiver. Sender 3 has the highest priority while the other two sender tasks are the same priority. Each sender task has a timer of random period generated by uniform distribution function, while the period of the receiver task is fixed to 100msec. The uniform distribution function is controlled by 6 upper and lower bounds. Each sender task sends a message to the queue containing “Time is XYZ” , from then a counter is incremented for the number of sent messages. If a sender attempts to send while the queue is full , the message is counted as blocked message. The receiver can receive maximum 1000 messages.

A diagram of a diagram

Description automatically generated

Figure 1: System design

A diagram of a system

Description automatically generated

Figure 2: System flowchart

# Results and Discussions

The code has two runs whether with queue equals to 3 or 10, each with 6 iterations according to the values of upper and lower bounds.

## Queue of size 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteration** | **Average sender timer period** | **Total messages** | **Total sent messages** | **Total blocked messages** | **Received messages** |
| 1 | 98.3333 | 3035 | 1002 | 2033 | 1000 |
| 2 | 139.333 | 2146 | 1002 | 1144 | 1000 |
| 3 | 181 | 1657 | 1002 | 655 | 1000 |
| 4 | 221.333 | 1353 | 1002 | 351 | 1000 |
| 5 | 260.333 | 1155 | 1002 | 153 | 1000 |
| 6 | 302.333 | 1008 | 1001 | 7 | 1000 |

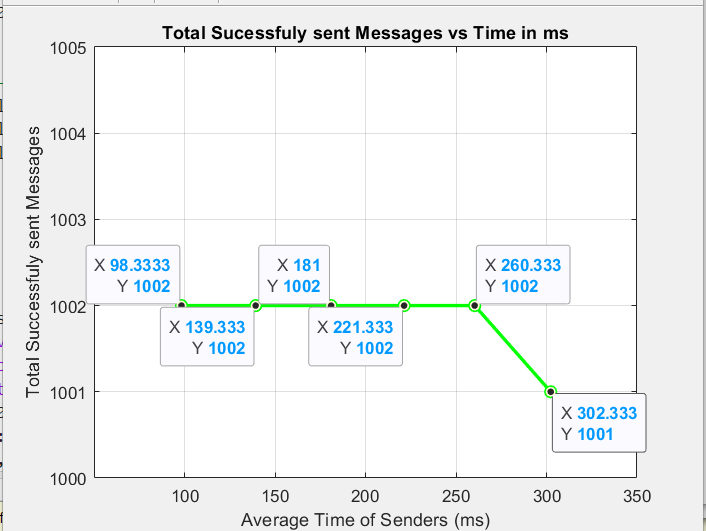
**Table 1: Statistics of all messages when queue equals 3**

### Total Number of Sent messages and blocked messages from all senders when queue equals 3

**A graph with numbers and a line

Description automatically generated**

Figure 3: Total sent messages by all senders at queue equals 3

****A graph with numbers and a line

Description automatically generated

Figure 5: Successfully sent messages from all senders (i.e: put in the queue)

Figure 4: Blocked messages from all senders when queue equals 3

**Discussion:** The gap between the number of blocked messages and successfully sent gives us an indication that the rate of sending is more than the rate of receiving messages. This leads to an accumulation of messages in the queue, resulting in a larger number of sent messages compared to the received messages.

### Number of Sent messages and blocked messages per senders when queue equals 3

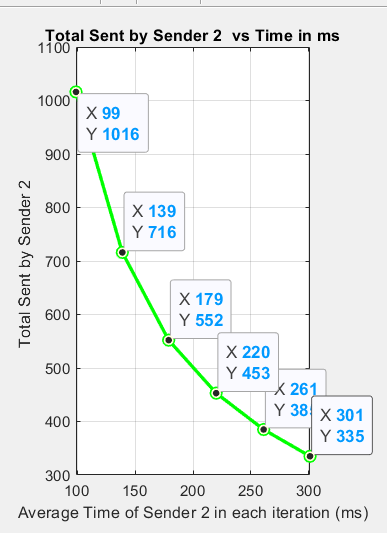
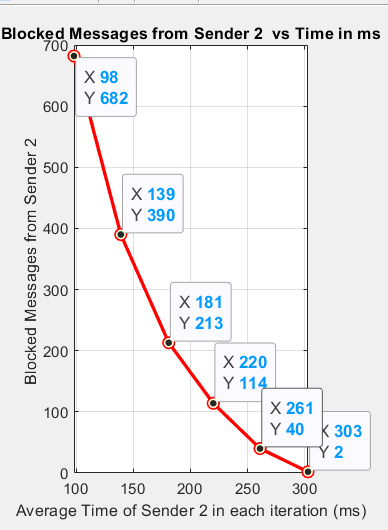
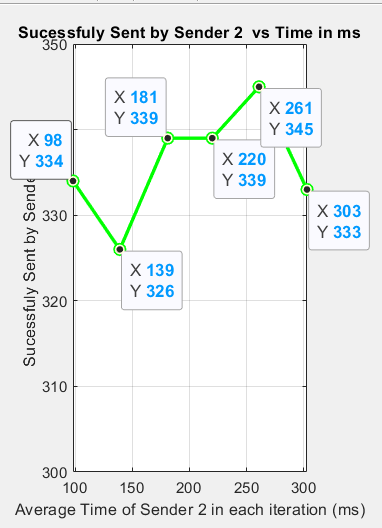
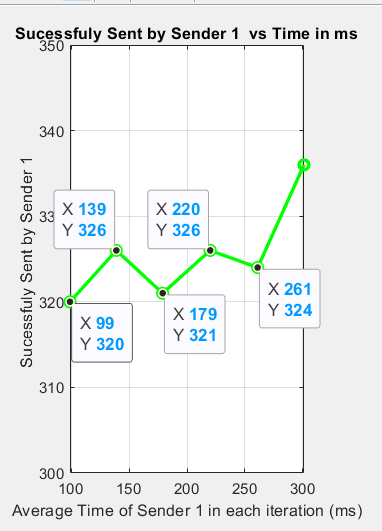
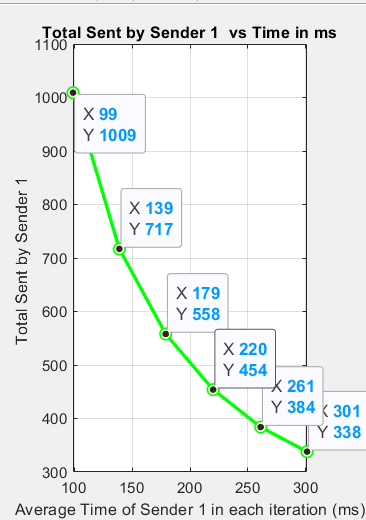
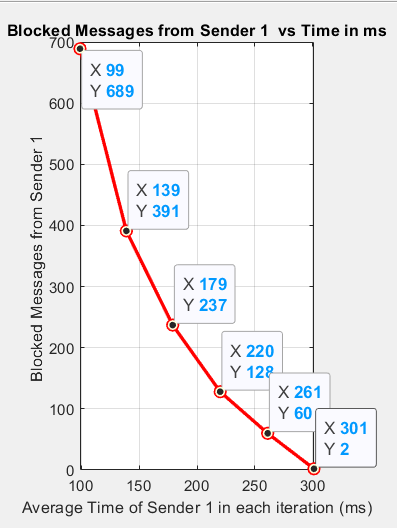


Figure 6: All messages from sender 1 when queue equal 3

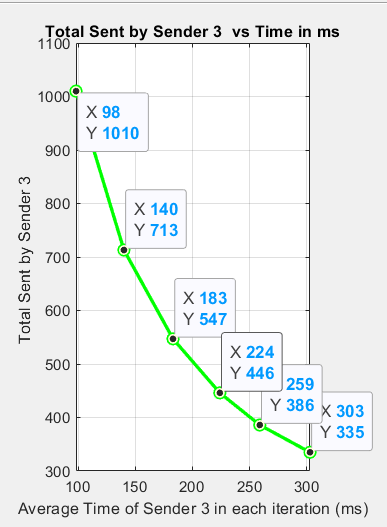
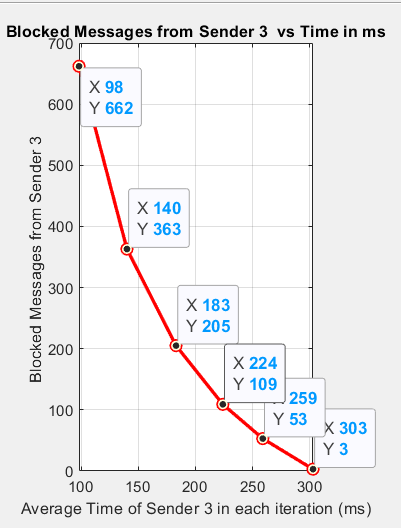
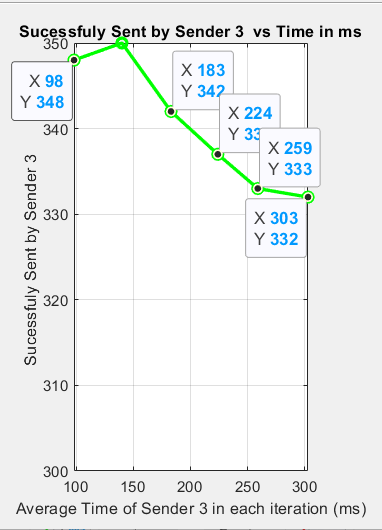


Figure 7: All messages from sender 2 when queue equal 3

Figure 8: All messages from sender 3 (highest priority) when queue equal 3

**Discussion:** As seen from the graphs of successfully sent messages, sender 3 sent more messages than the other two senders. This is because it has more priority than the others. Moreover, the blocked messages of sender 3 (highest priority) is the least.

## Queue of size 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteration** | **Average sender timer period** | **Total messages** | **Total sent messages** | **Total blocked messages** | **Received messages** |
| 1 | 98.3333 | 3035 | 1009 | 2026 | 1000 |
| 2 | 139.333 | 2146 | 1009 | 1137 | 1000 |
| 3 | 181 | 1657 | 1009 | 648 | 1000 |
| 4 | 221.333 | 1353 | 1009 | 344 | 1000 |
| 5 | 260.333 | 1155 | 1009 | 146 | 1000 |
| 6 | 302.667 | 1000 | 1000 | 0 | 1000 |

**Table 2: Statistics of all messages when queue equals 10**

### Total Number of Sent messages and blocked messages from all senders when queue equals 10

**A graph with numbers and a line

Description automatically generated**

Figure 9: Total sent messages from all senders when queue equals 10

A graph with numbers and a line

Description automatically generatedA graph with a green line

Description automatically generated

Figure 10: Blocked Messages from all senders when queue equals 10

Figure 11: Successfully sent messages from all senders when queue equals 10

**Discussion:** In both queues, we find that the total number of sent messages is the same so the rate of sending is same in both. While we see that the successful sent messages when queue equals 10 is more than when queue equals 3 because the larger queue size enables more messages to be received.

### Number of Sent messages and blocked messages per senders when queue equals 10

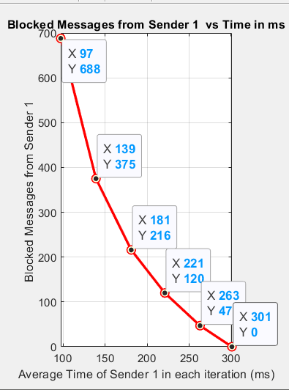
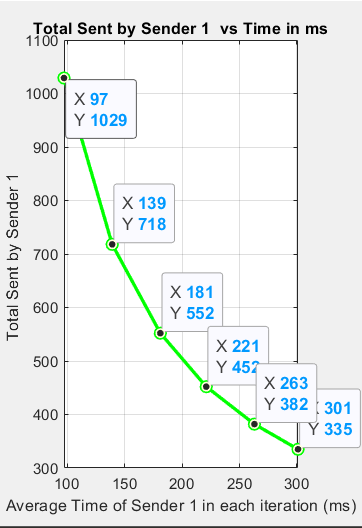
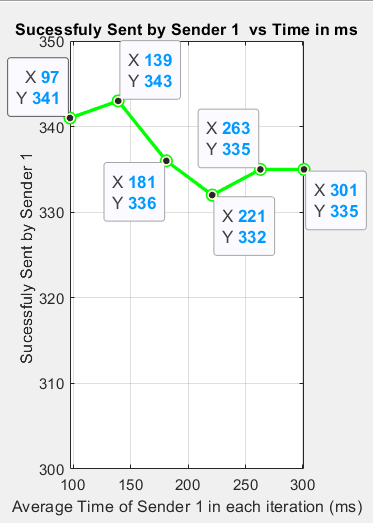


Figure 12: All messages by sender 1 when queue equals 10

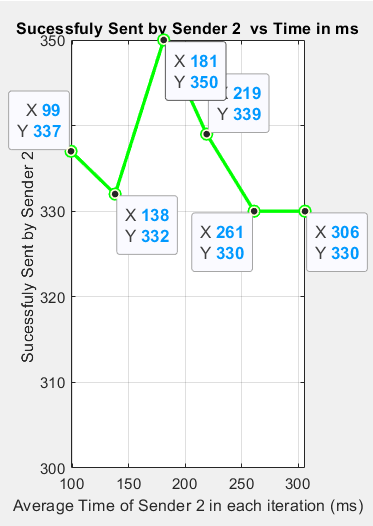
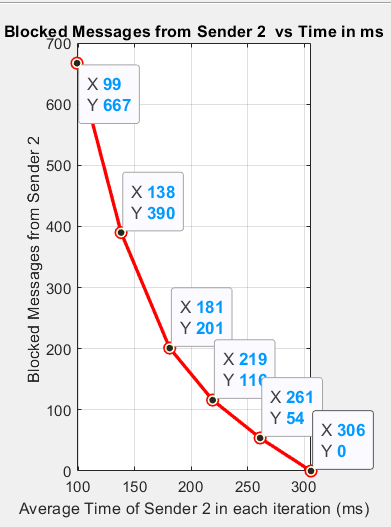
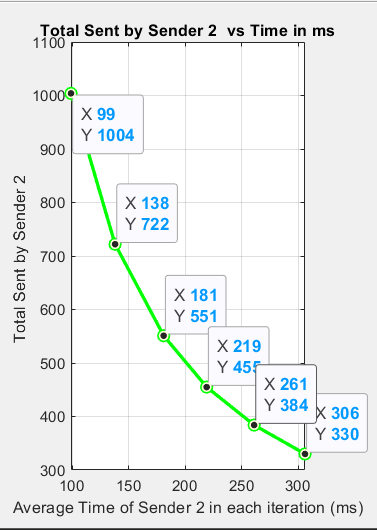


Figure 13:All messages by sender 2 when queue equals 10

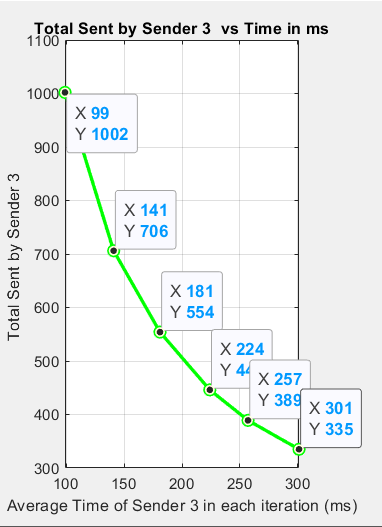
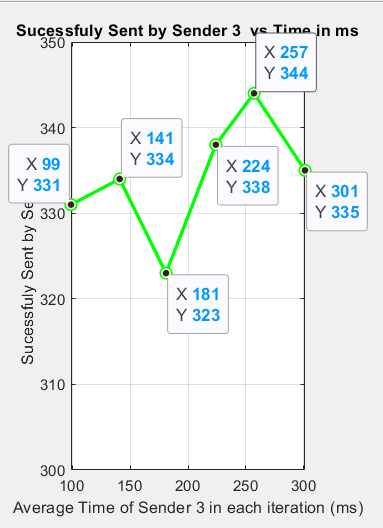
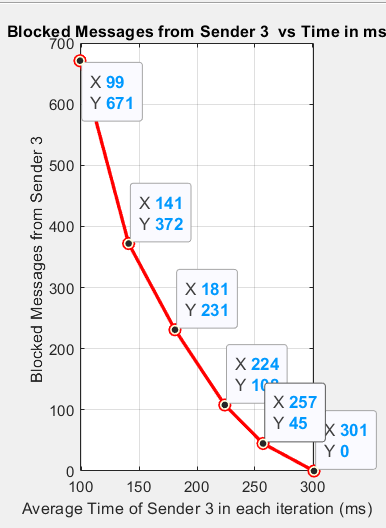


Figure 14:All messages by sender 3 when queue equals 10

# Code snippets

|  |  |
| --- | --- |
| Function: SenderTask(parameters)  Assign parameters to taskNum  While True:  Generate and store random period for timer  Start timer  Wait for semaphore[taskNum] then take it  Send message “Time is XYZ” to queue  If sending is successful:  Increment sentMessages[taskNum]  Else:  Increment blockedMessages[taskNum]  End if  End while  End function | Function: ReceiverTask (parameters):  Assign parameters to taskNum  While True:  Start timer  Wait for receiverSemaphore then take it  Receive message from queue  If receiving is successful:  Increment receivedMessages and print the message  Else:  Print “Could not receive from the queue.”  End if  End while  End function |

**Table 3: Pseudo code for SenderTask and ReceiverTask functions**

# Appendix: Extra Materials

In case there is any problem in the resolution of the graphs, they can be directly accessed on google drive through this link :

[Graphs in](https://drive.google.com/drive/folders/14ywizVE9j-RGNg-7VBULZNsu1okW8Uh6?usp=sharing) high resolution

[System Design and Flowchart in high resolution](https://drive.google.com/drive/folders/1VgzO8SeoLjhgk14IjPg1FbjJ0LGtPYfI?usp=drive_link)

# References

1. “FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions,” *FreeRTOS*. https://freertos.org/index.html