BLG 433E Computer Communications Project 3

Due: Tuesday, May 28, 2024

Lecturer: Prof. Abdul Halim Zaim CRN: 21358

Fatih Baskın
150210710

Contents

Explanation of Simulator **3**
- Setup . . . 3
- Simulation Inputs . . . 3
- Simulation Outputs . . . 3
- How does it work? . . . 4
  - Datagram Generators . . . 4
  - Router Scheduling . . . 4
  - Statistics Calculation . . . 4

Simulation Results **4**
- ABR Service with FIFO Scheduling . . . 4
- CBR Service with FIFO Scheduling . . . 7
- VBR Service with FIFO Scheduling . . . 10
- ABR Service with RR Scheduling . . . 13
- CBR Service with RR Scheduling . . . 16
- VBR Service with RR Scheduling . . . 19
- ABR Service with WFQ Scheduling . . . 22
- CBR Service with WFQ Scheduling . . . 25
- VBR Service with WFQ Scheduling . . . 28

Problems with Simulation **31**

Explanation of Simulator

Setup

In order to run the simulation, first, you need to generate a Python virtual environment. You can create one using `python -m venv .venv` and activate it. Then you should install the needed packages using `pip install -r requirements.txt`. After this is done, you can run simulation.py inside the Python virtual environment

Simulation Inputs

When the simulation program is run, it asks for several inputs from the console. Those are:

- Burst rate of a router port
- Queue size of a router port
- Number of ports (queues)
- Bit rate of the service
- Datagram generator type (service type)
  - CBR (Constant Bit Rate)
  - VBR (Variable Bit Rate)
  - ABR (Available Bit Rate)
- Router scheduling strategy
  - FIFO (First In First Out)
  - RR (Round Robin) aka Fair Scheduling
  - WFQ (Weighted Fair Queue)

Simulation Outputs

The simulation will run for 10 seconds with 0.1-second iterations. The simulation would yield several files. Those are:

- Datagrams log containing information on every datagram in the simulation.
- Average queue times per port as a graph.
- Average response times per port as a graph.
- Average waiting times per port as a graph.
- General picture of waiting times, response times, and queue times as a graph.
- Average throughput per port as a graph.
- Overall average throughput as a graph.

Explanation of Simulator continued on next page...

How does it work?

Datagram Generators

Datagram generators represent the sources of the datagrams in the ATM network. They have three different types, each generating datagrams with the service they represent. Constant bit rate datagram generators will generate constant bit-size datagrams. Available bit-rate datagram generators will generate datagrams as the size available in the router queue. Variable bit-rate datagram generators will generate datagrams under a Poisson distribution, with the mean (λ) being the bit-rate of the service.

Router Scheduling

Router has several ports and each port has a queue. In FIFO, the earlier packet in the queues is selected and sent out of the network, then the next earliest is found, and so on. In Round Robin, the burst rate is divided evenly into every queue/port, and data is sent evenly. In the weighted fair queue, the burst rate is divided according to the weights (priorities) of the ports.

Statistics Calculation

Each datagram has three time-stamp sections. Those are in-queue time, out-queue time, and sent time. Queue time is calculated by $in - queue - time \ - \ out - queue - time$, and waiting time is the total time spent by a datagram to be sent, therefore it is calculated by $sent - time \ - \ in - queue - time$. Response time is the total time the router requires to process and send the datagram. It is calculated as $sent - time \ - \ queue - out - time$.

Simulation Results

ABR Service with FIFO Scheduling

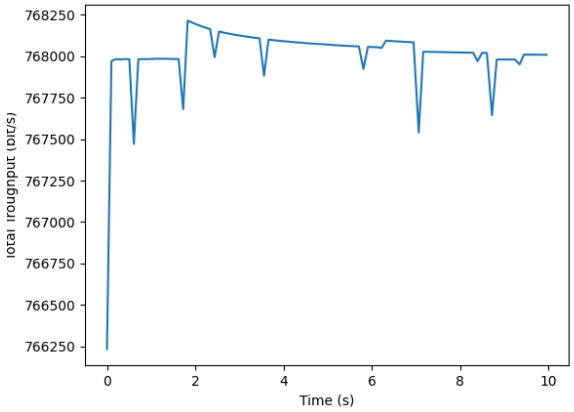

Figure 1: Overall throughput in the simulation with ABR service with FIFO Scheduling

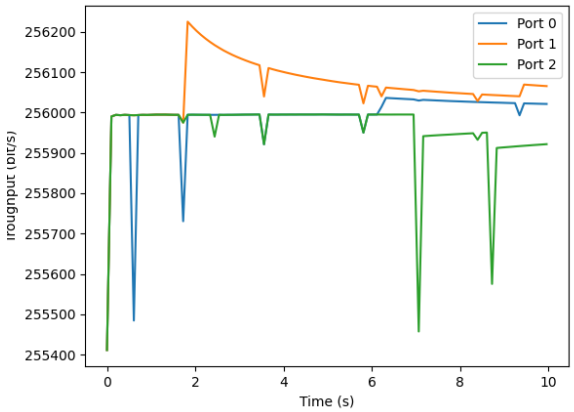

Figure 2: Throughput per port in the simulation with ABR service with FIFO Scheduling

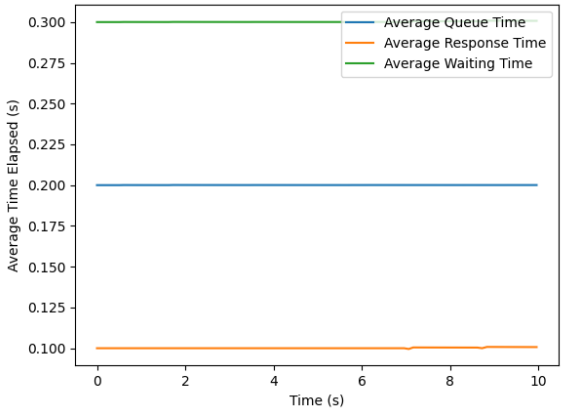

Figure 3: Overall datagram statistics in the simulation with ABR service with FIFO Scheduling

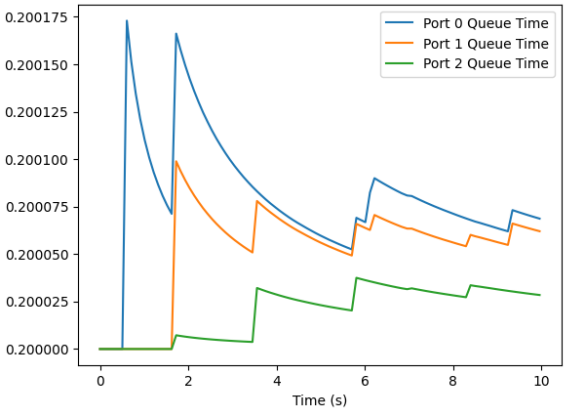

Figure 4: Queue times per port in the simulation with ABR service with FIFO Scheduling

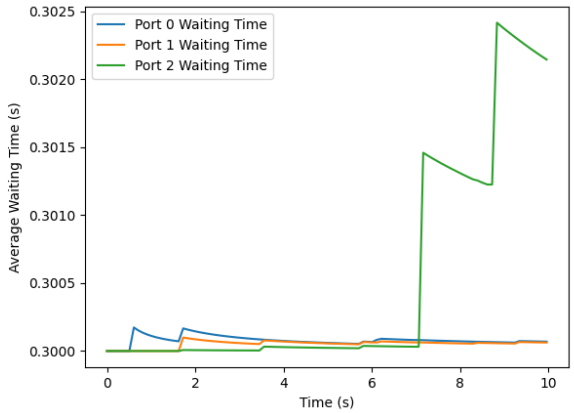

Figure 5: Waiting times per port in the simulation with ABR service with FIFO Scheduling

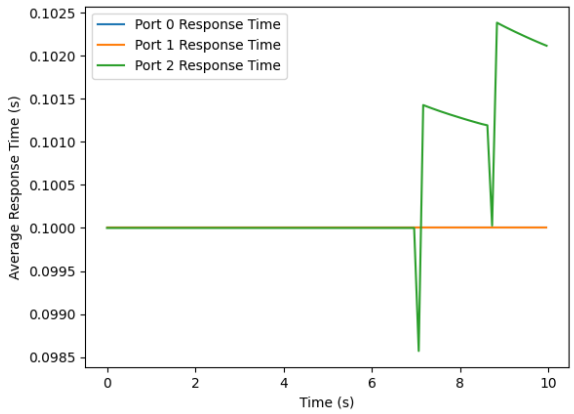

Figure 6: Response times per port in the simulation with ABR service with FIFO Scheduling

CBR Service with FIFO Scheduling

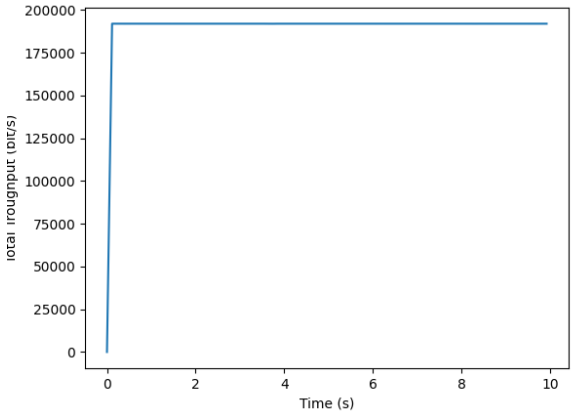

Figure 7: Overall throughput in the simulation with cbr service with FIFO Scheduling

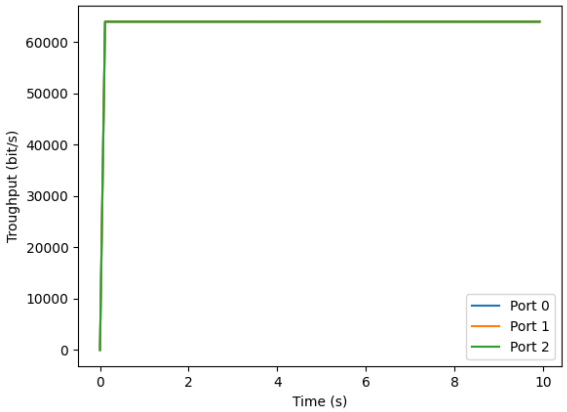

Figure 8: Throughput per port in the simulation with cbr service with FIFO Scheduling

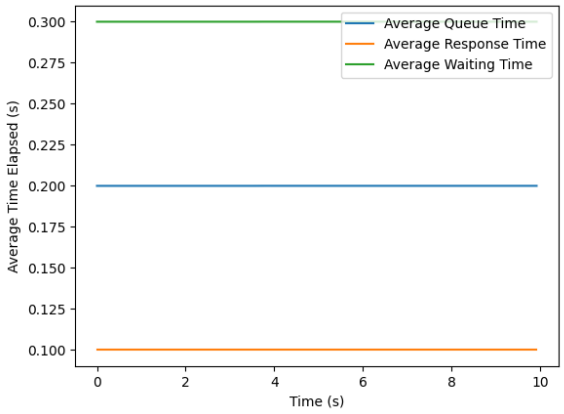

Figure 9: Overall datagram statistics in the simulation with cbr service with FIFO Scheduling

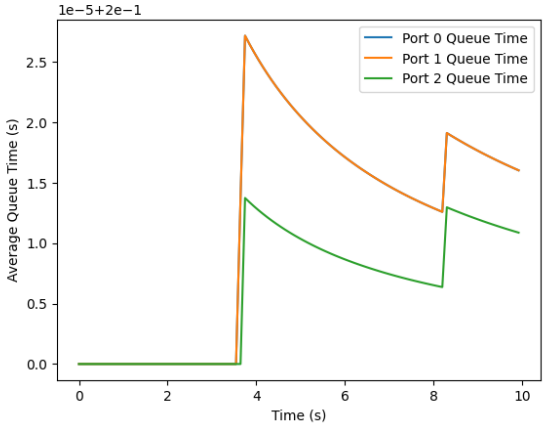

Figure 10: Queue times per port in the simulation with cbr service with FIFO Scheduling

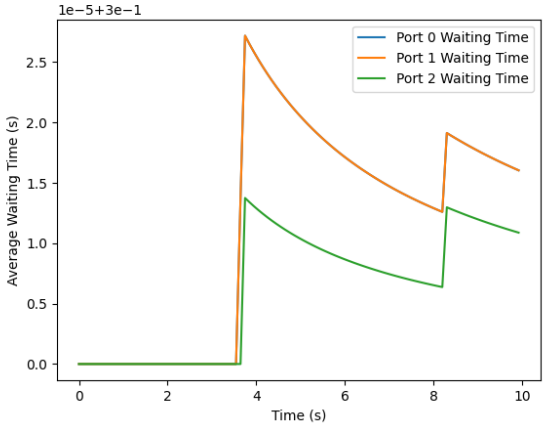

Figure 11: Waiting times per port in the simulation with cbr service with FIFO Scheduling

Simulation Results continued on next page...

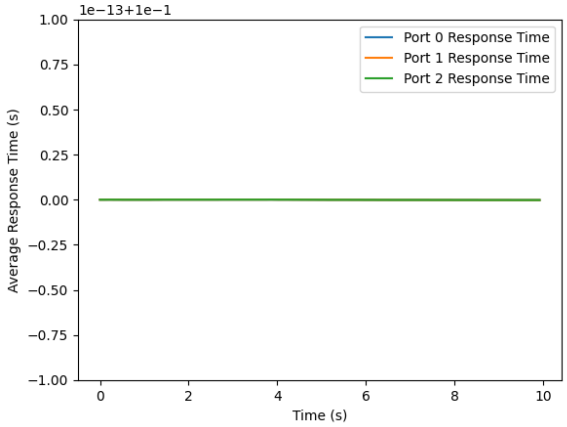

Figure 12: Response times per port in the simulation with cbr service with FIFO Scheduling

VBR Service with FIFO Scheduling

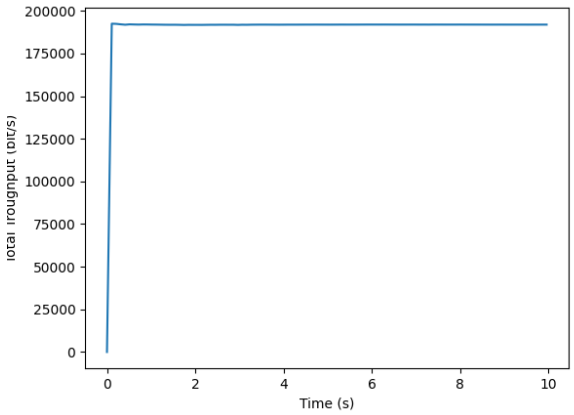

Figure 13: Overall throughput in the simulation with vbr service with FIFO Scheduling

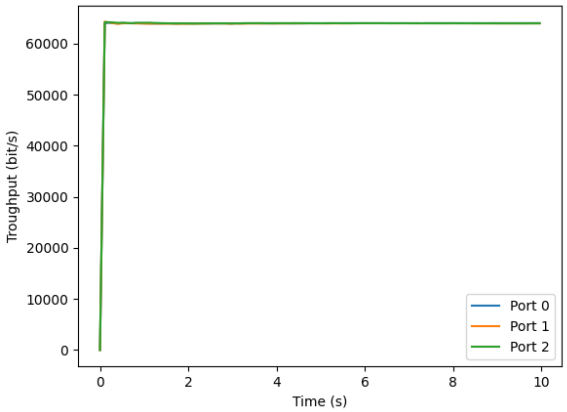

Figure 14: Throughput per port in the simulation with vbr service with FIFO Scheduling

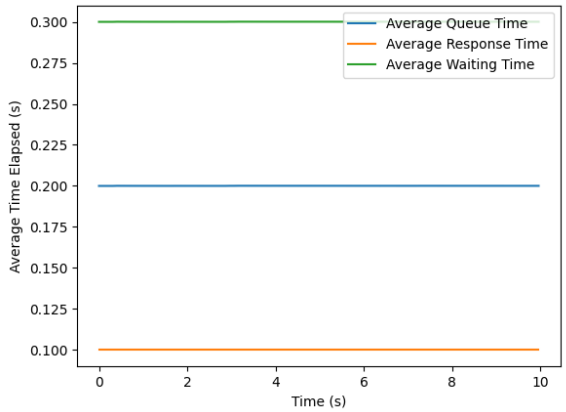

Figure 15: Overall datagram statistics in the simulation with vbr service with FIFO Scheduling

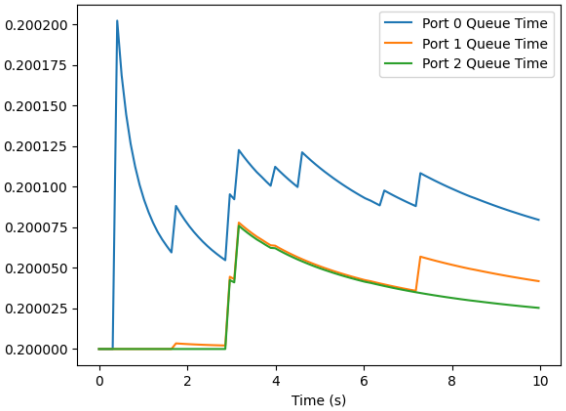

Figure 16: Queue times per port in the simulation with vbr service with FIFO Scheduling

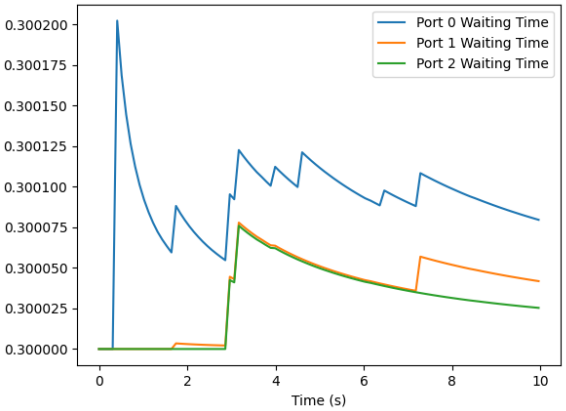

Figure 17: Waiting times per port in the simulation with vbr service with FIFO Scheduling

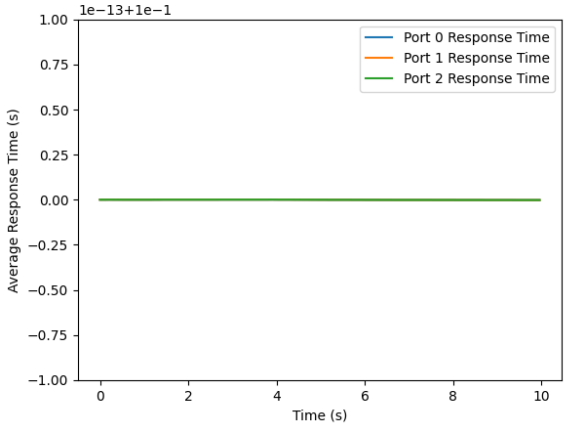

Figure 18: Response times per port in the simulation with vbr service with FIFO Scheduling

ABR Service with RR Scheduling

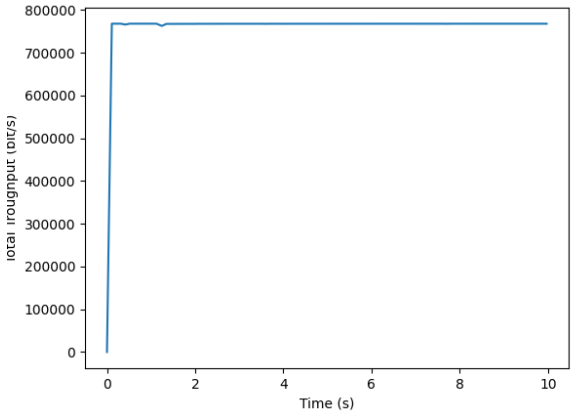

Figure 19: Overall throughput in the simulation with abr service with RR Scheduling

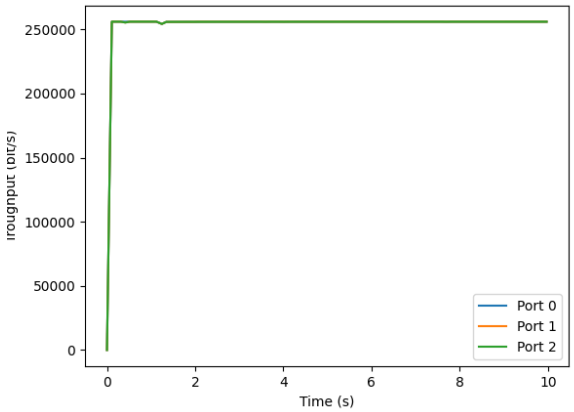

Figure 20: Throughput per port in the simulation with abr service with RR Scheduling

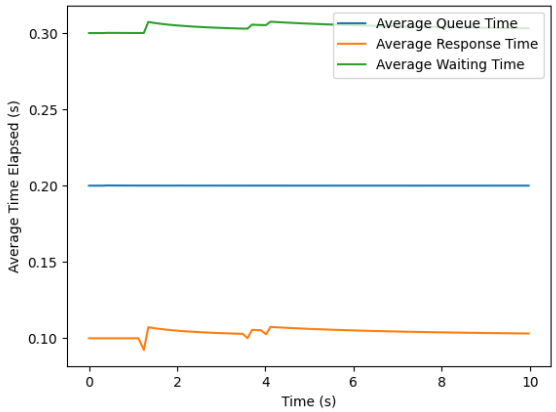

Figure 21: Overall datagram statistics in the simulation with abr service with RR Scheduling

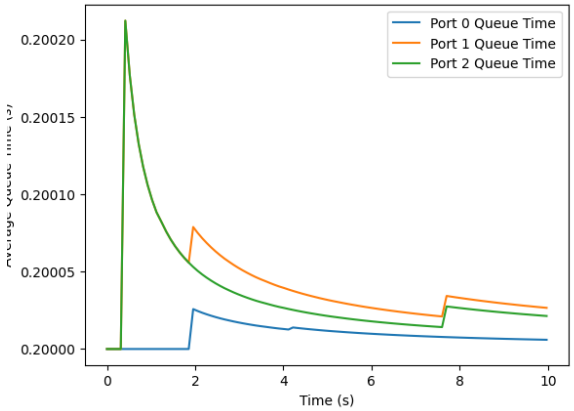

Figure 22: Queue times per port in the simulation with abr service with RR Scheduling

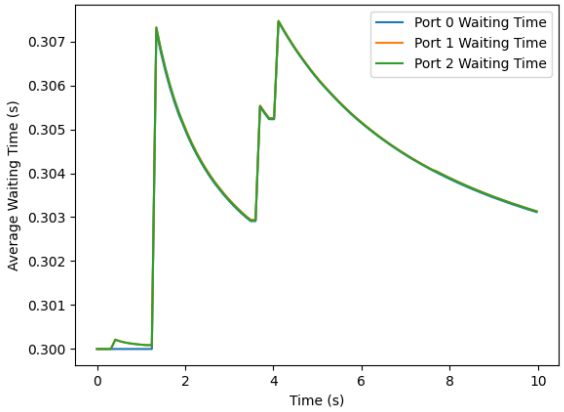

Figure 23: Waiting times per port in the simulation with abr service with RR Scheduling

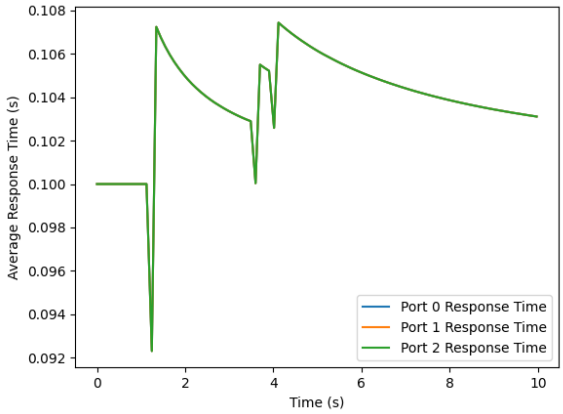

Figure 24: Response times per port in the simulation with abr service with RR Scheduling

CBR Service with RR Scheduling

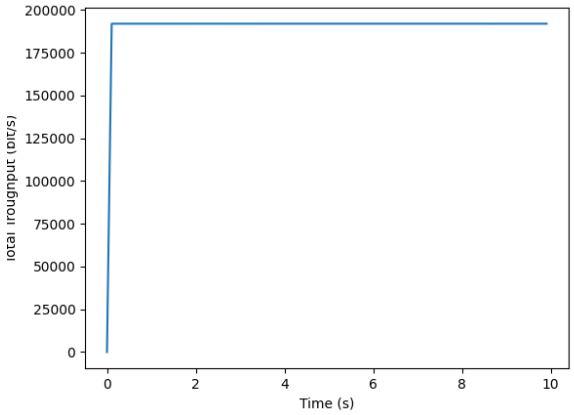

Figure 25: Overall throughput in the simulation with cbr service with RR Scheduling

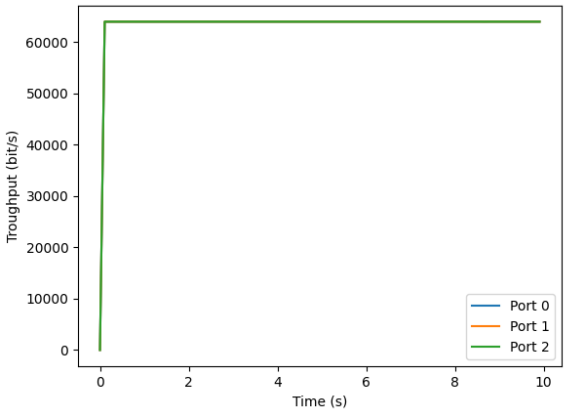

Figure 26: Throughput per port in the simulation with cbr service with RR Scheduling

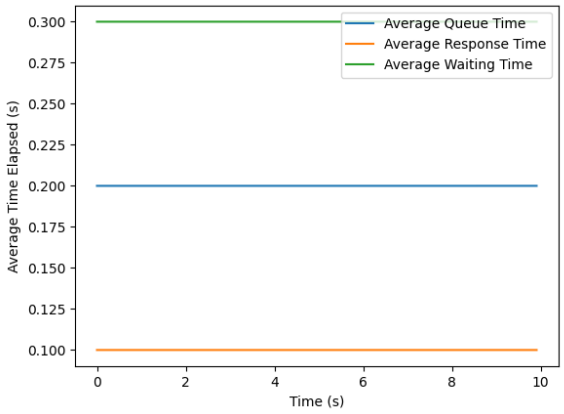

Figure 27: Overall datagram statistics in the simulation with cbr service with RR Scheduling

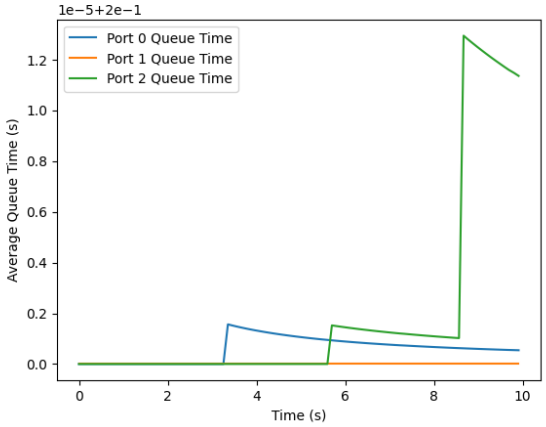

Figure 28: Queue times per port in the simulation with cbr service with RR Scheduling

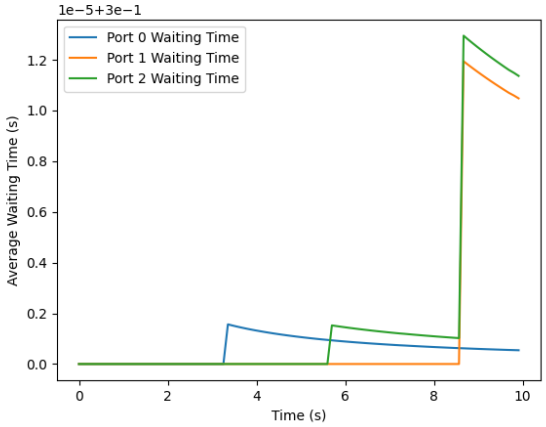

Figure 29: Waiting times per port in the simulation with cbr service with RR Scheduling

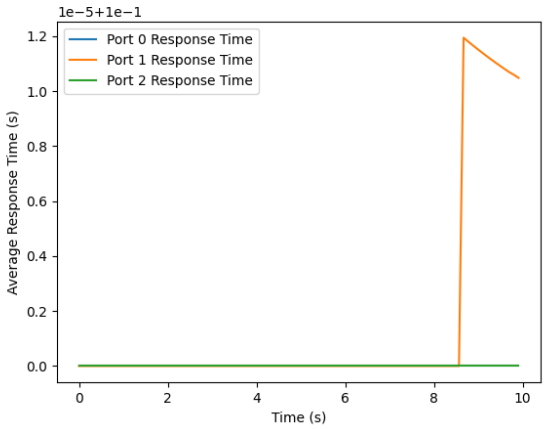

Figure 30: Response times per port in the simulation with cbr service with RR Scheduling

VBR Service with RR Scheduling

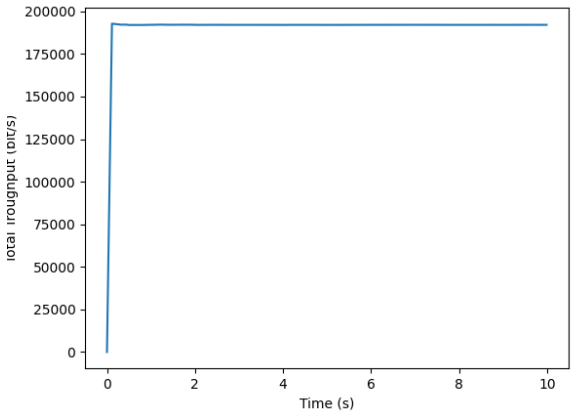

Figure 31: Overall throughput in the simulation with vbr service with RR Scheduling

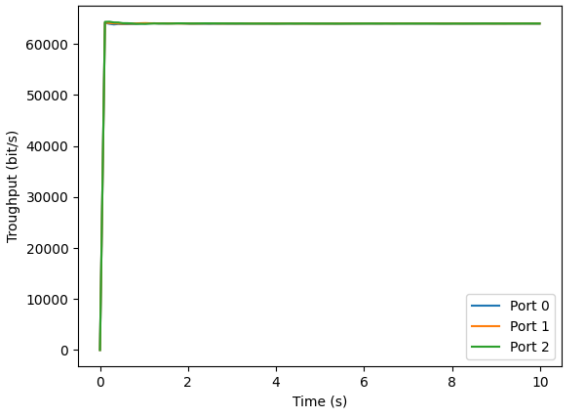

Figure 32: Throughput per port in the simulation with vbr service with RR Scheduling

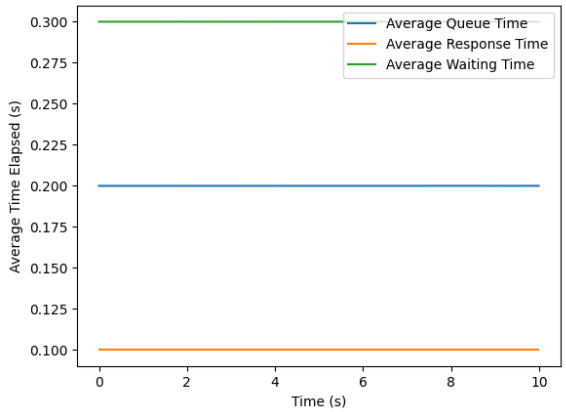

Figure 33: Overall datagram statistics in the simulation with vbr service with RR Scheduling

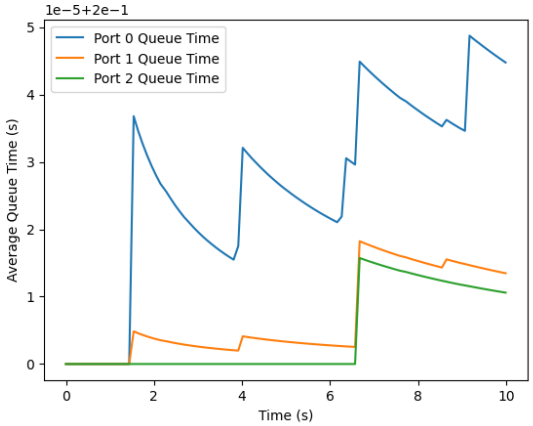

Figure 34: Queue times per port in the simulation with vbr service with RR Scheduling

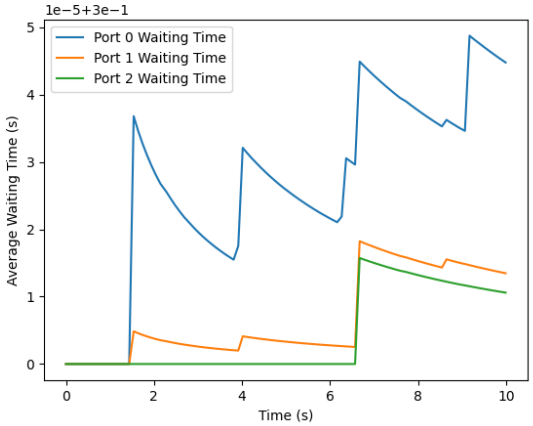

Figure 35: Waiting times per port in the simulation with vbr service with RR Scheduling

Simulation Results continued on next page...

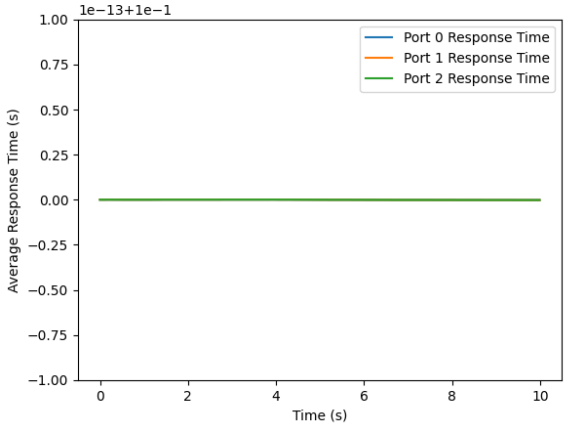

Figure 36: Response times per port in the simulation with vbr service with RR Scheduling

ABR Service with WFQ Scheduling

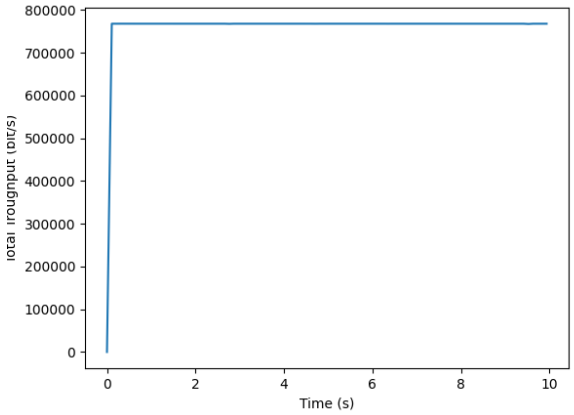

Figure 37: Overall throughput in the simulation with abr service with wfq Scheduling

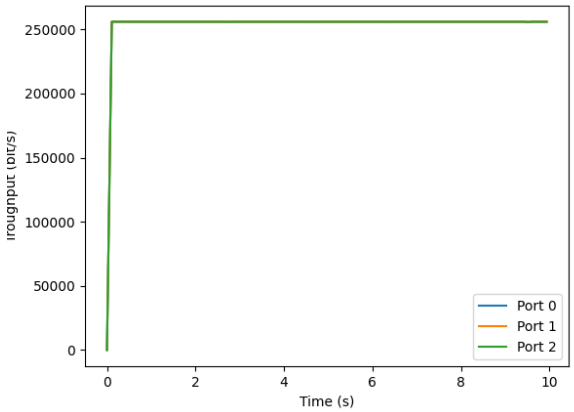

Figure 38: Throughput per port in the simulation with abr service with wfq Scheduling

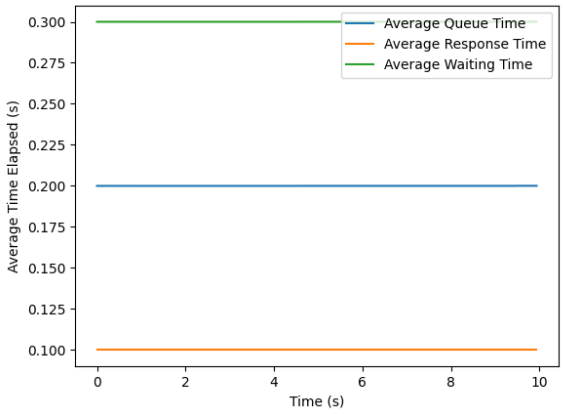

Figure 39: Overall datagram statistics in the simulation with abr service with wfq Scheduling

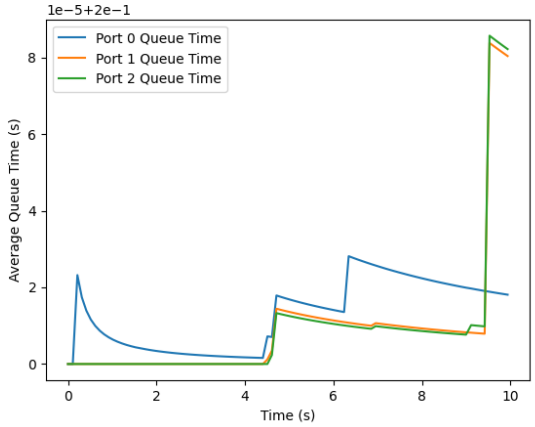

Figure 40: Queue times per port in the simulation with abr service with wfq Scheduling

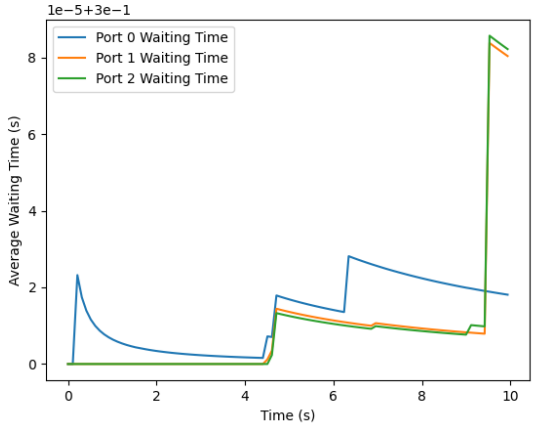

Figure 41: Waiting times per port in the simulation with abr service with wfq Scheduling

Simulation Results continued on next page...

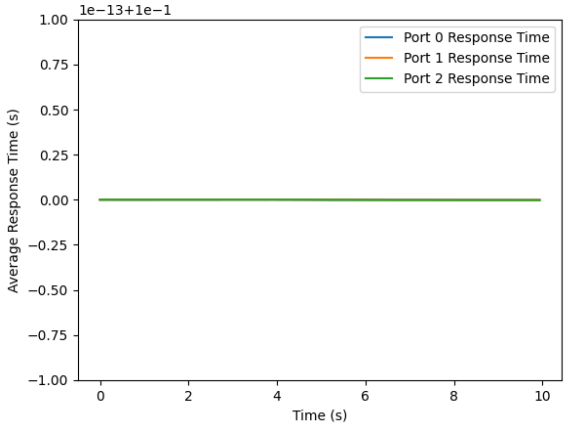

Figure 42: Response times per port in the simulation with abr service with wfq Scheduling

CBR Service with WFQ Scheduling

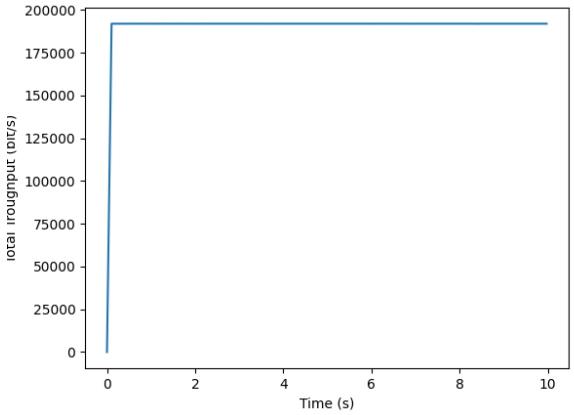

Figure 43: Overall throughput in the simulation with cbr service with wfq Scheduling

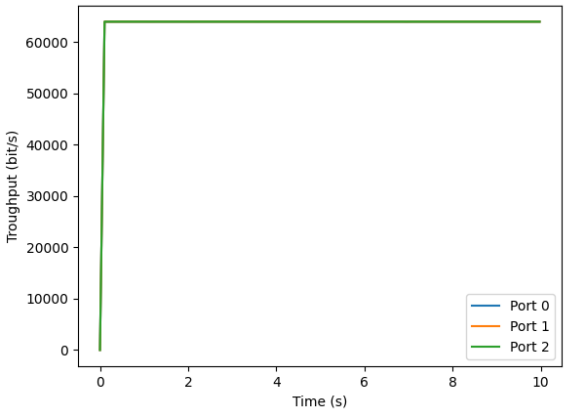

Figure 44: Throughput per port in the simulation with cbr service with wfq Scheduling

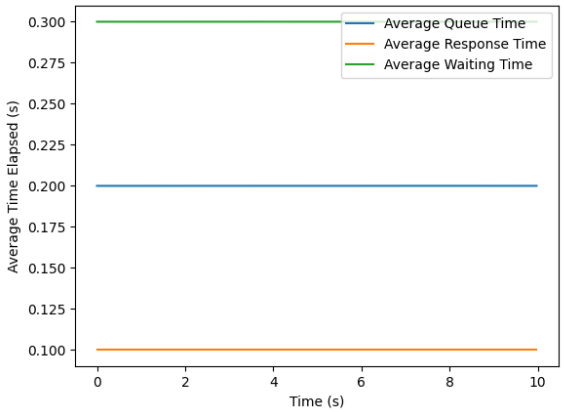

Figure 45: Overall datagram statistics in the simulation with cbr service with wfq Scheduling

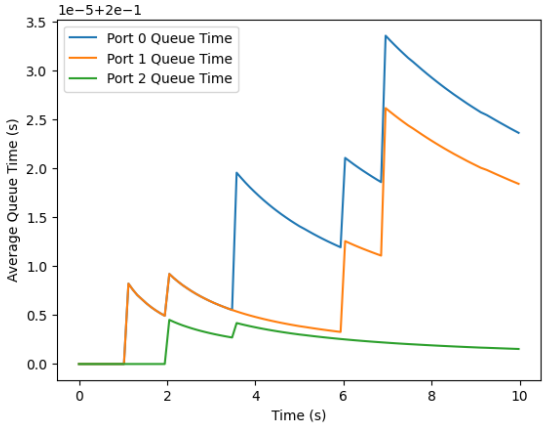

Figure 46: Queue times per port in the simulation with cbr service with wfq Scheduling

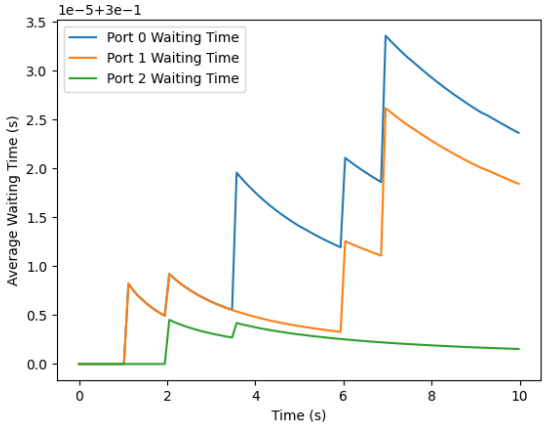

Figure 47: Waiting times per port in the simulation with cbr service with wfq Scheduling

Simulation Results continued on next page...

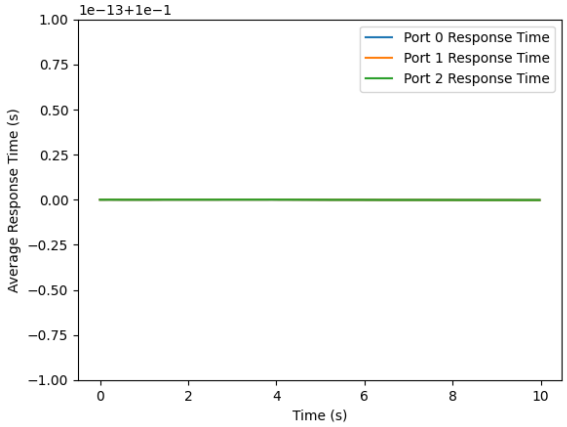

Figure 48: Response times per port in the simulation with cbr service with wfq Scheduling

VBR Service with WFQ Scheduling

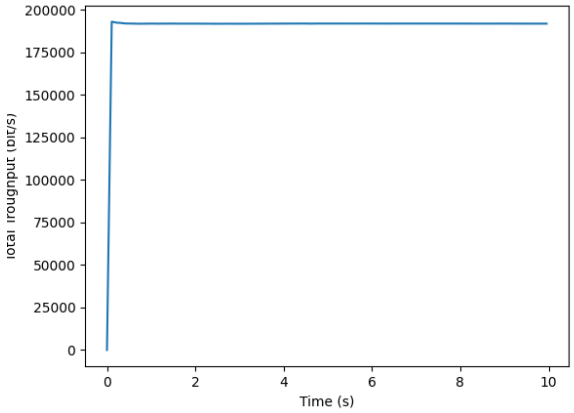

Figure 49: Overall throughput in the simulation with vbr service with wfq Scheduling

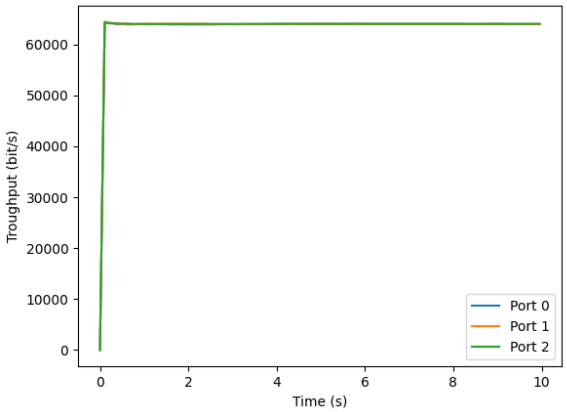

Figure 50: Throughput per port in the simulation with vbr service with wfq Scheduling

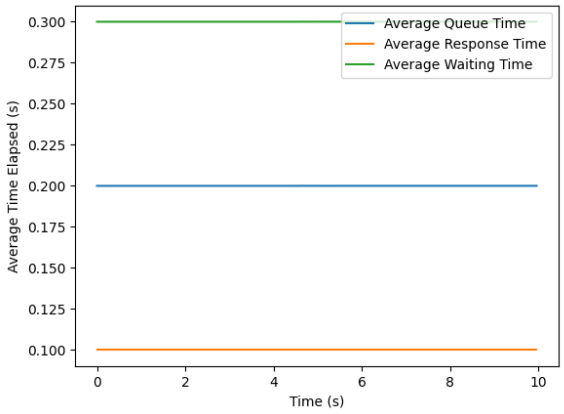

Figure 51: Overall datagram statistics in the simulation with vbr service with wfq Scheduling

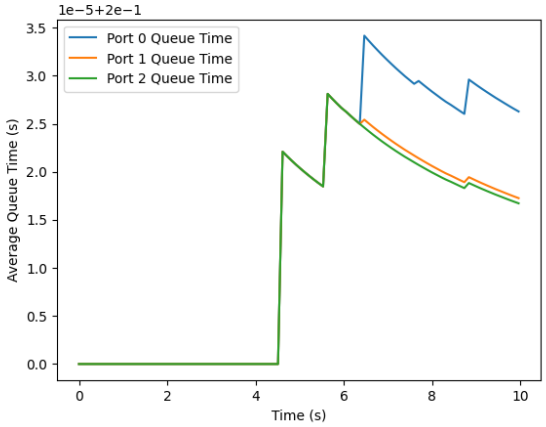

Figure 52: Queue times per port in the simulation with vbr service with wfq Scheduling

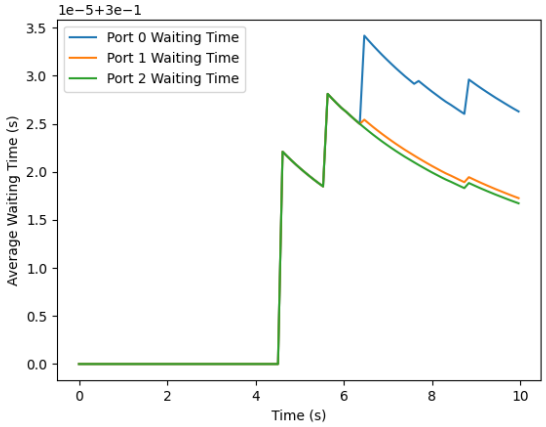

Figure 53: Waiting times per port in the simulation with vbr service with wfq Scheduling

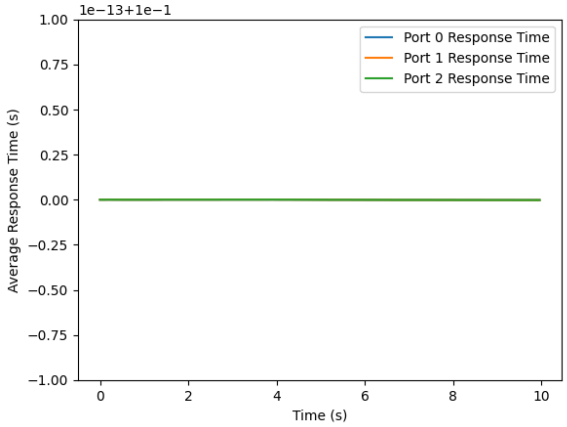

Figure 54: Response times per port in the simulation with vbr service with wfq Scheduling

Problems with Simulation

I have written the code for the simulation hastily, and several bugs need to be fixed, I would also make some major refactoring to the code. The code can be written in OOP standards with more robust strategies.

I also believe that I got the meaning of response time wrong. I would refactor it with the correct knowledge.