UNIVERSITÀ
degli STUDI
di CATANIA

Department of Electrical, Electronic and Computer Engineering

MASTER'S DEGREE COURSE IN COMPUTER ENGINEERING

Final Report – Omnet++ Project

Extensive

Industrial Automation Networking Course

Academic Year 2021-2022

Authors:

Roggio Maria (1000037341)

Strange Joseph (1000027120)

Ventimiglia Andrea Paolo (1000039024)

Summary

1. Introduction

1.1. Machine diagram

We want to implement and simulate the following scenario of a network that contains the following characteristics:

1. 2 Switch
2. 18 End-node
3. Traffico cross-domain
    a. ADAS
    b. Infotainment

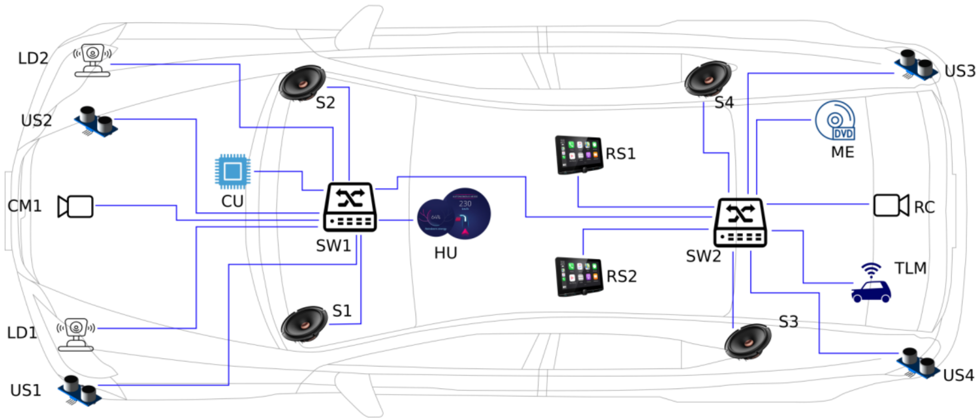


Figure 1.1 Machine Scenery

1.2. Flows

The flows that are produced by the various End-nodes are shown in the following table:

Src	Dst	Periodo	Deadline Rel.	Payload	Burst
LD1, LD2	CU	1.4 ms	1.4 ms	1300 byte	1
ME	S1, S2, S3, S4	250 us	250 us	80 byte	1
US1, US2, US3, US4	CU	100 ms	100 ms	188 byte	1
CU	HU	10ms	2 ms	1500 byte	7
CM1	HU	16.66 ms	16.66 ms	1500 byte	119
ME	RS1, RS2	33.33 ms	33.33 ms	1500 byte	119
TLM	HU, CU	625 us	625 us	600 byte	1
RC	HU	33.33 ms	33.33 ms	1500 byte	119

Table 1.1 End-Node Flows

1.3. Metrics

The metrics that are measured at the application layer are:

- End-to-end delay: $e2eDelay = RxTime - GenTime \quad per\ flusso$
- Frame Loss Ratio (FLR): $FLR = \frac{frame\_scartate}{frame\_trasmesse} \quad per\ flusso$

1.4. Scenario 3 - Variant 4

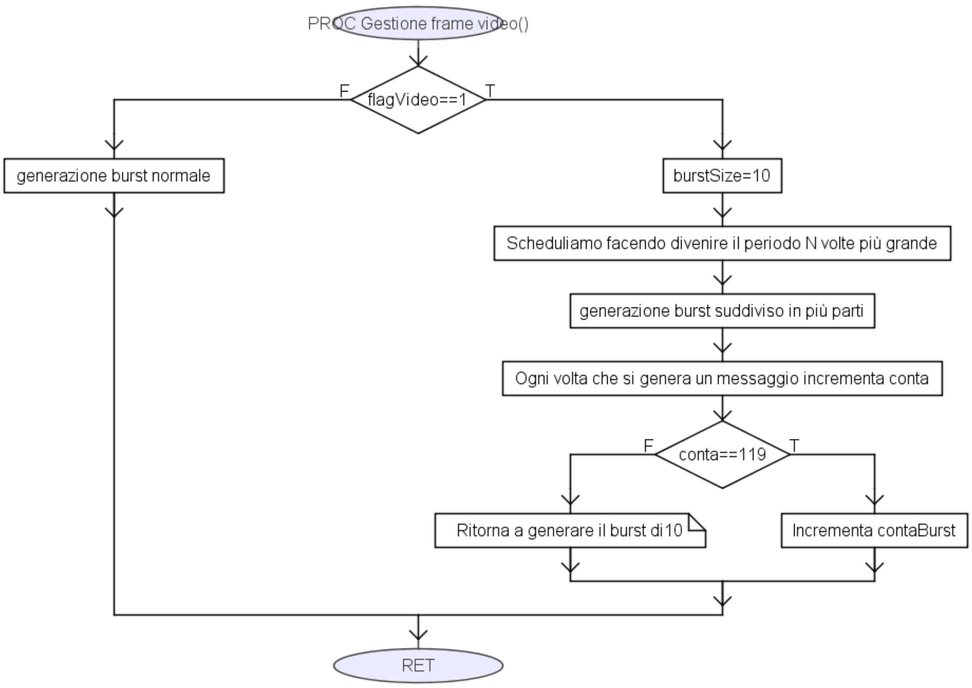
In detail, we will deal with two types of scheduling:

- a static priority (Deadline Monotonic), in which the priority will be set by relative deadline with a maximum of 8 priorities;
- a dynamic priority (EDF), in which the priority will be set by absolute deadline that will be encoded in the payload of the Ethernet frame.

The queue will be limited to 10 frames in size, in which video bursts should not be discarded.
In addition, e2eDelay and FLR will be calculated for each stream.

1.5. Scelte implementative

The first design choice was to create a limited queue that once the maximum limit was reached would discard all the frames except the video frames, as shown in the following figure:



The following block diagram has been implemented in the periodicBustApp.

- Through a flag, implemented in the omnet.ini, we verify whether the frame belongs to a video burst or not.

- If it is not a video frame, it continues to generate a burst without having to implement changes. Conversely, if the frame belongs to a video burst we check if a counter, set at the beginning of the code, is equal to 119 (critical value since 109 video frames would be discarded, thus violating the design constraint) in this case we resize the burst size to 10, dividing the single burst into several bursts and increasing the period without violating the deadline.

The second design choice was to implement a dispatcher between the node and the nic, whose task is to receive the frames of the n nodes at the same time and then forward them individually to the nic because the latter, having only one port, could not receive the n frames all at once.

The other design choices are only optimization and customization of the code.

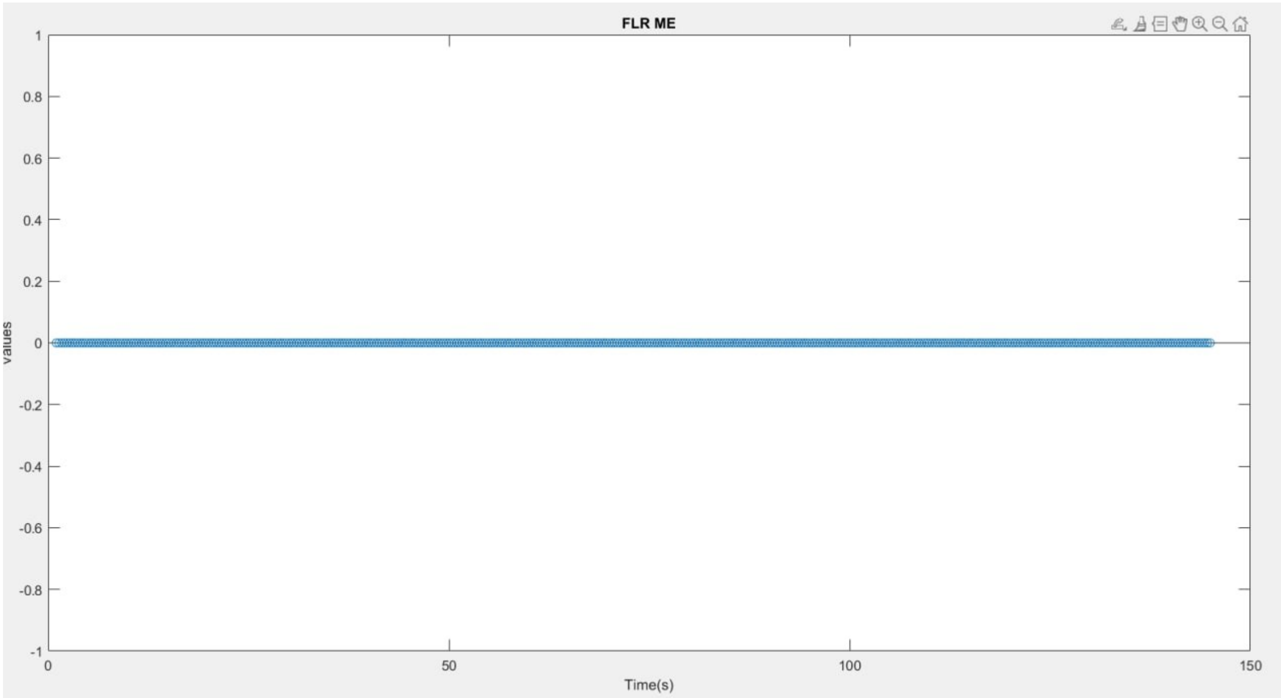
2. Static Priority

2.1. Specific

- Using Deadline Monotonic (max priority 8)
- Queue limited to 10 frames
- No video frame loss

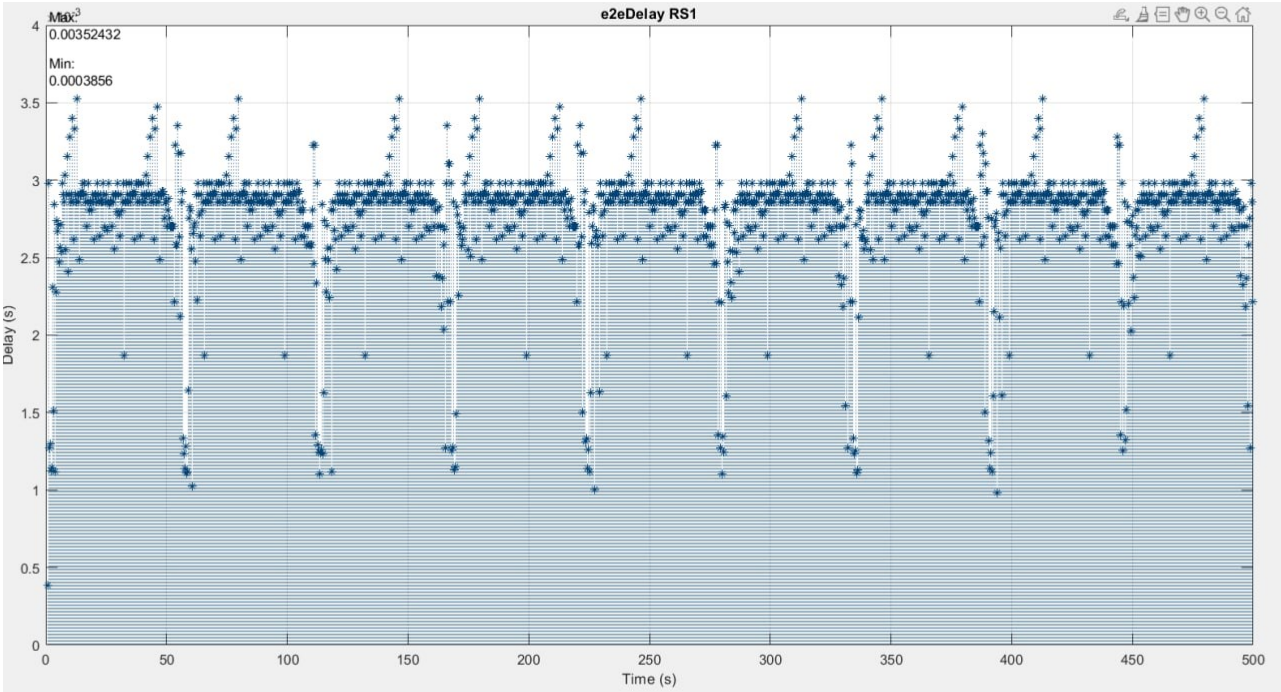
2.2. Results

The main task of the analysis to be defined is the study of the video traffic generated by the ME node. What we could see is that FLR (Frame Loss Ratio) is always 0, the result is what we hoped for since there must be no discarded frames:



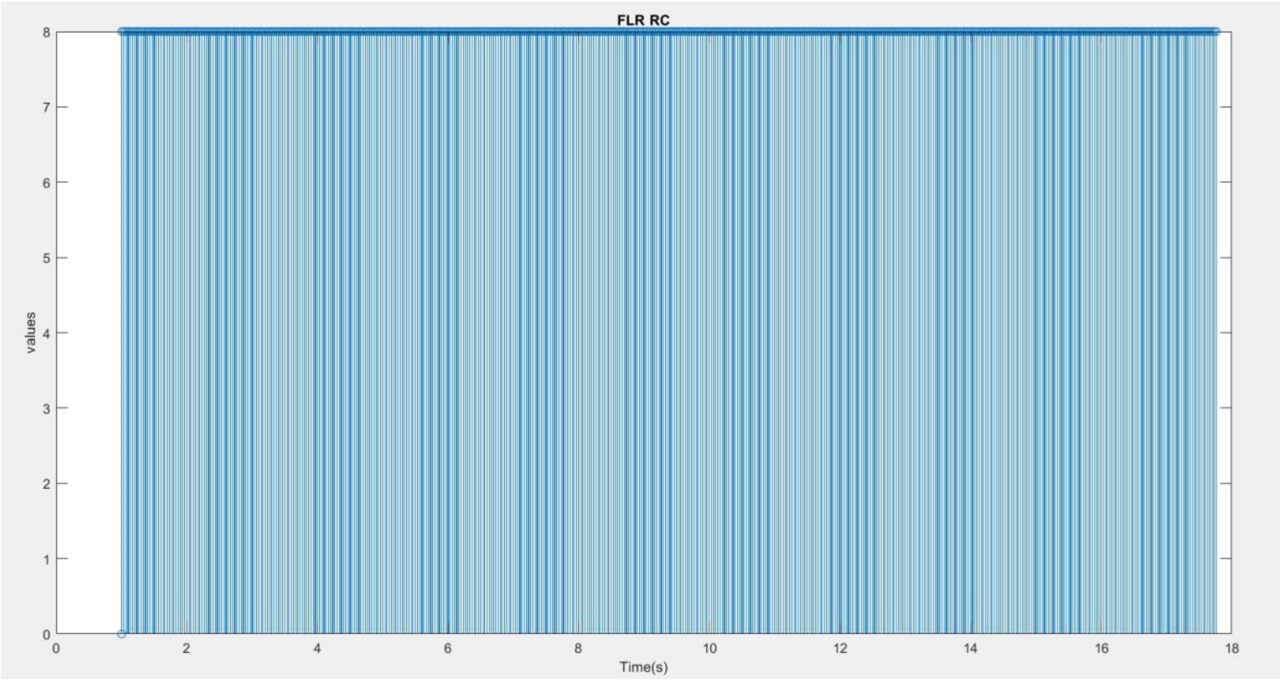
As can be seen from the graph above, the FLR is always constant at 0 even if there are multiple flows.

Another result obtained is related to the end-to-end delay, of the RS1 node that receives from ME. The end-to-end delay usually fluctuates no more than 0.5 seconds. This is a great result since the frames don't wait long to be transmitted:

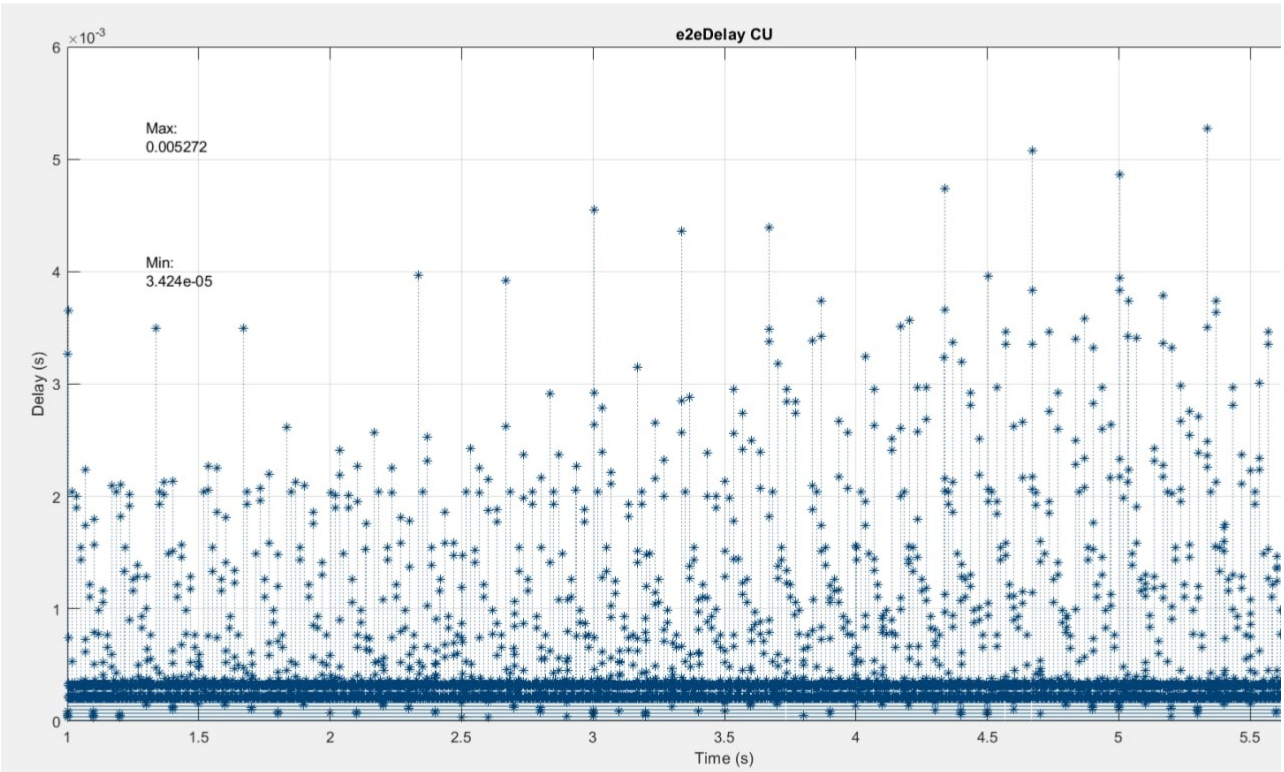


2.3. Valori end-to-end delay e FLR

The results obtained in the previous paragraph, concerning the ME node, were obtained by not having discarded frames and therefore not by losing packets. This consideration cannot be made for the other nodes present in the network, since no mechanism has been implemented for them to safeguard and avoid the loss of frames. In fact, the FLR turns out to be non-zero as can be seen from the graph below concerning the trend of the RC node:



For completeness, we also report the graph of the end-to-end delay of the CU node:



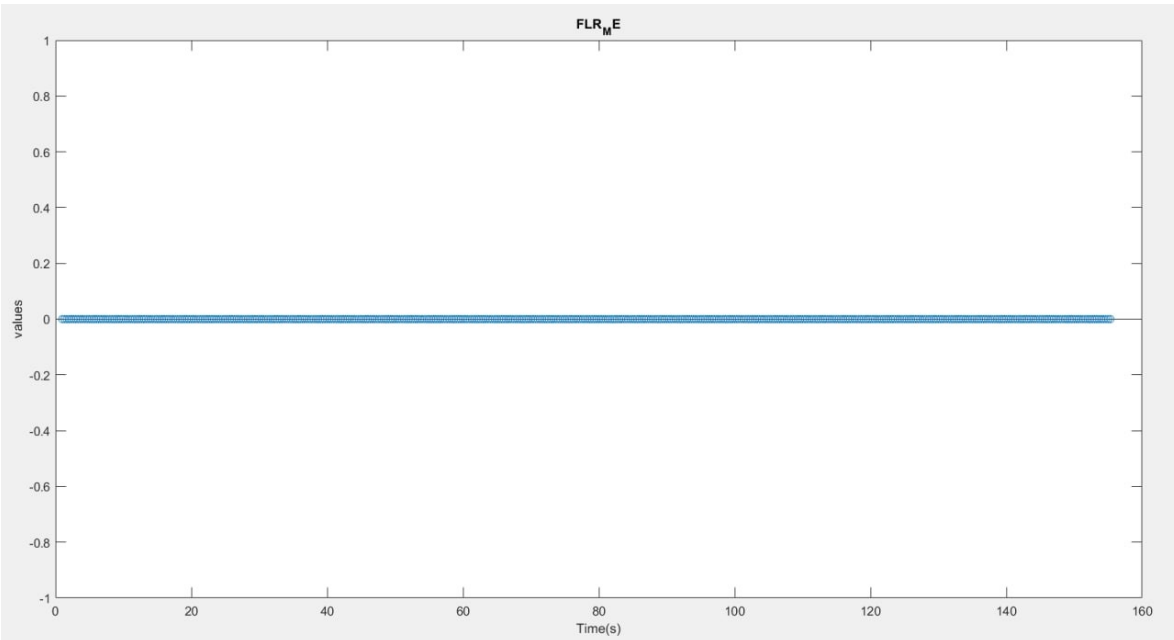
3. Dynamic Prioritization

3.1. Specific

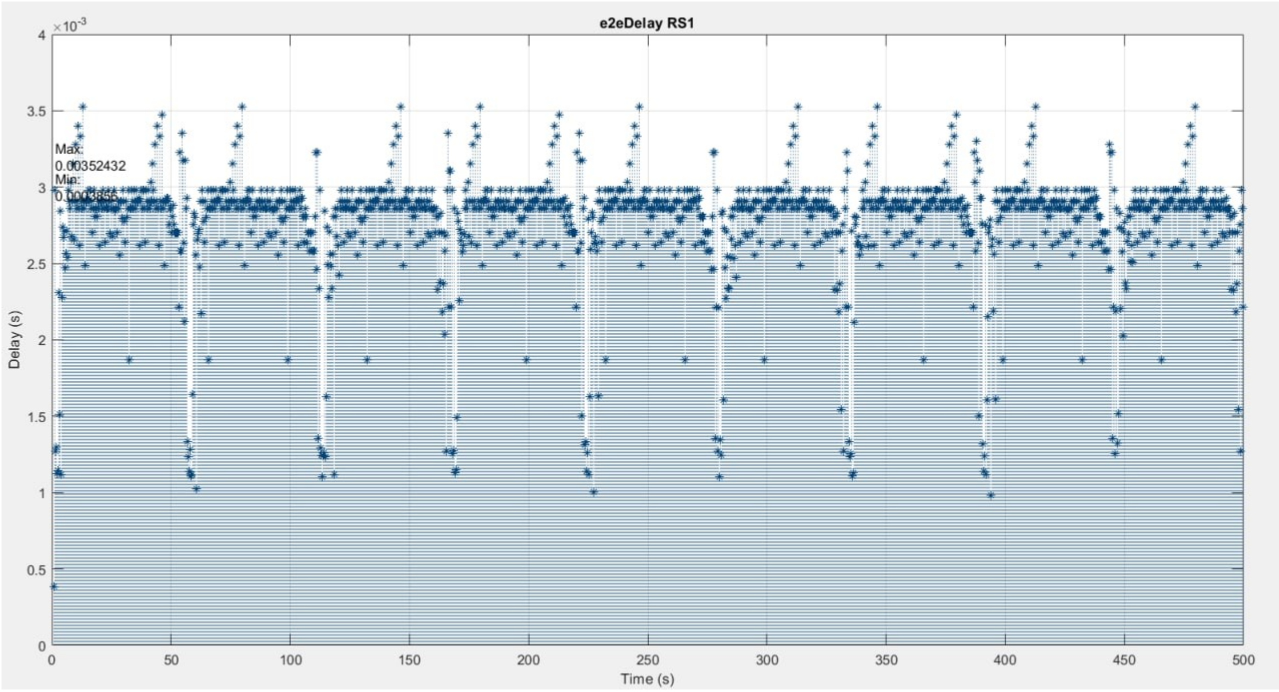
- Using Erliest Deadline First-EDF
- Queue limited to 10 frames
- No video frame loss
- Absolute Deadlines encoded in the Ethernet frame payload

3.2. Results

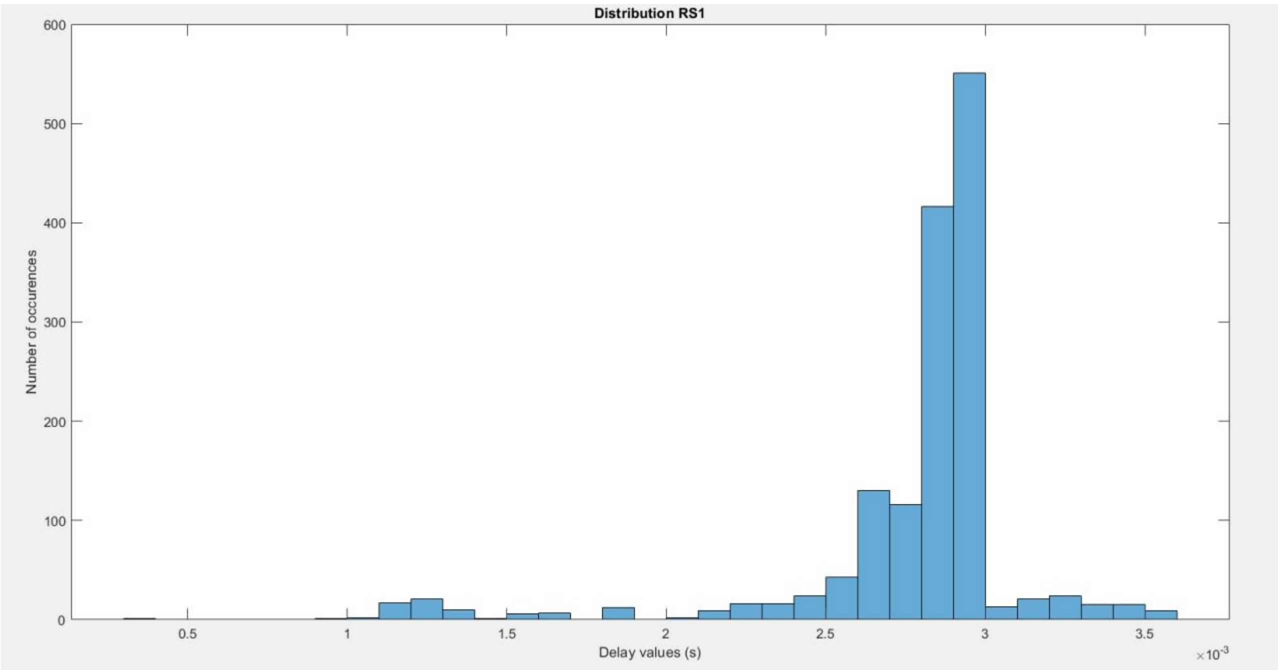
Compared to the previous case, where the priority is static, with dynamic priority it was possible to see that the FLR remained unchanged, since the mechanism adopted previously does not need to be modified as it does not depend on priority. We can see it from the following graph:



Another node taken into consideration, in addition to the ME, is RS1. The values of the end-to-end delay and its distribution relative to the previously mentioned node will be shown below:

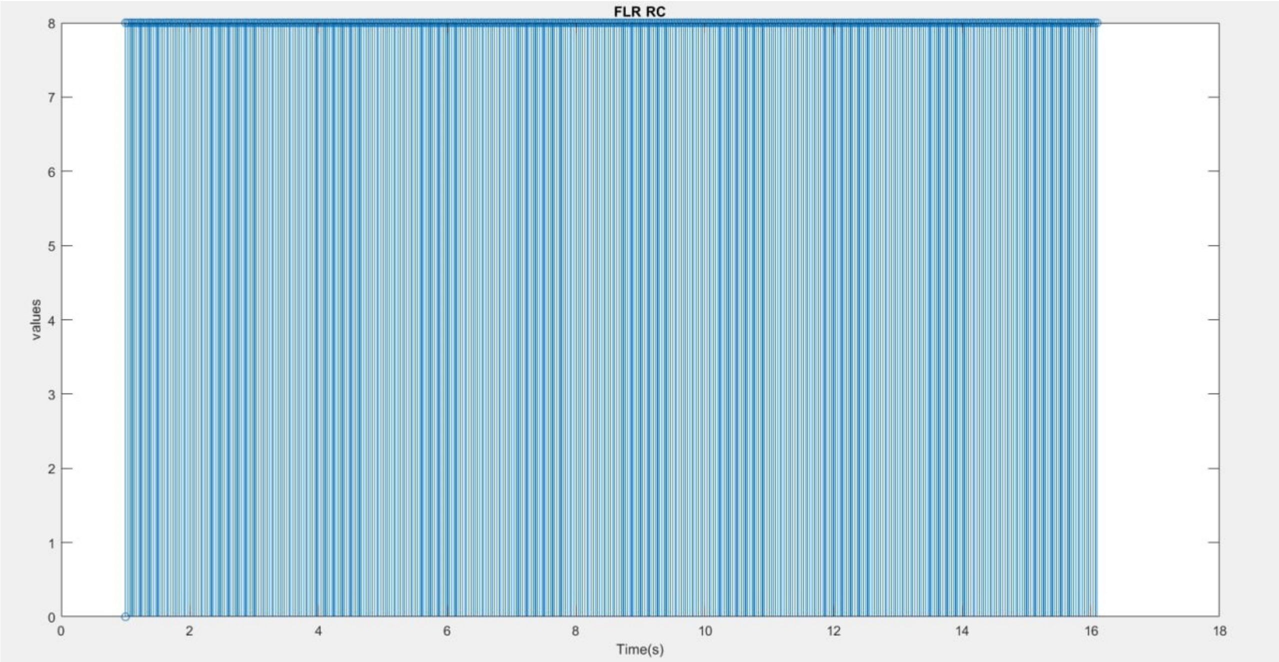


As shown in the upper left of the figure, there are the minimum and maximum points.



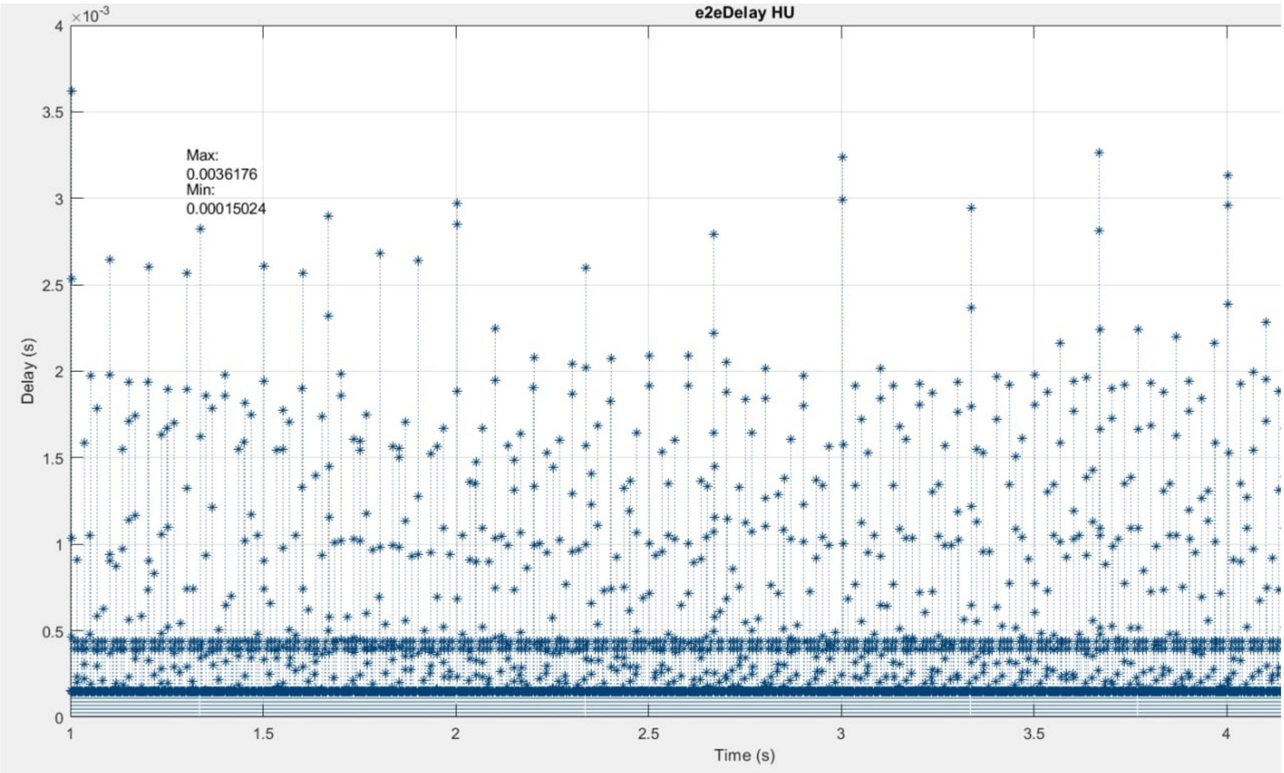
3.3. Valori end-to-end delay e FLR

The results obtained in the previous paragraph, concerning the ME node, were obtained by not having discarded frames and therefore not by losing packets. This consideration cannot be made for the other nodes present in the network, since no mechanism has been implemented for them to safeguard and avoid the loss of frames. In fact, the FLR turns out to be non-zero as can be seen from the graph below concerning the trend of the RC node:



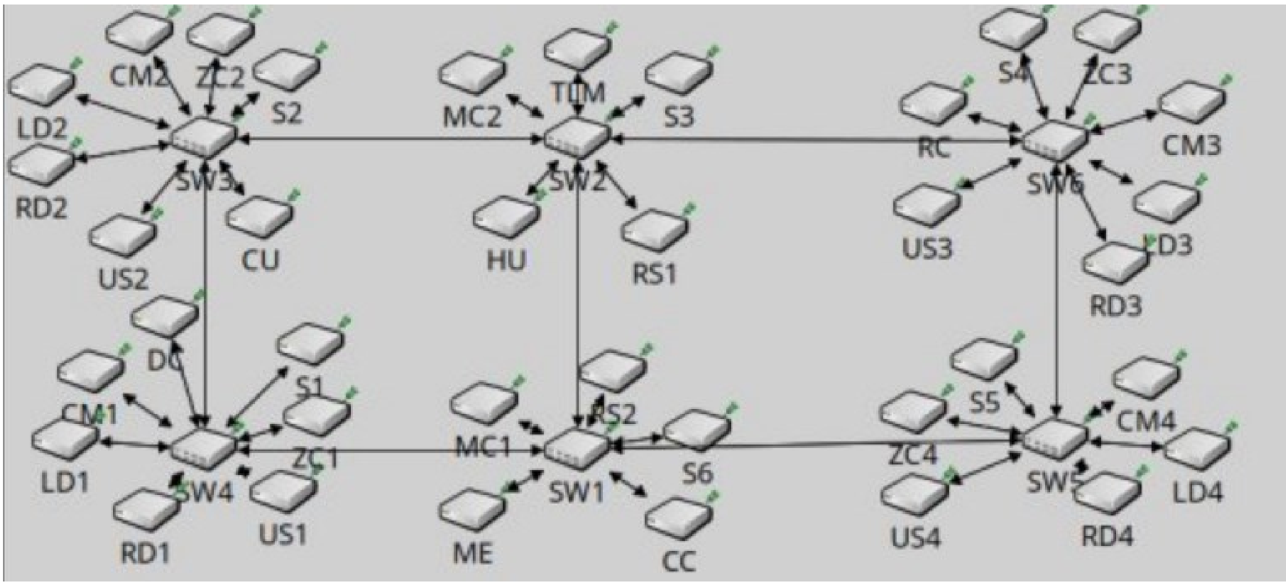
As can be seen from the graph, the RC node periodically loses a certain number of frames that cause the FLR to increase dramatically.

For the sake of completeness, we also plot the end-to-end delay of another node in the HU network as a comparison between its values and those of the ME:



4. Extension

4.1. Scenario 2 – Variant 1



4.2. Simulation

Eseguire simulazioni al variare della priorità assegnata ad ogni flusso da 0 a 7.

Priorità statica scelta casualmente per ogni flusso a inizio simulazione.
Stessa priorità per tutta la simulazione
Più simulazioni possibili con seed diversi (minimo 5000 simulazioni.)

Durata simulazione calcolata per collezionare almeno 5000 risultati per ogni flusso.

File csv risultante per ogni flusso:
NomeFlusso; Src; Dst; periodo; deadline relativa; payload; burst; sw1; sw2; priorità; DMR; Max(e2eDelay); Avg(e2eDelay); numRxFrames;

4.3. Flows

Nome	Src	Dst	Periodo	Deadline Rel.	Payload	Burst
LiDAR	LD1, LD2, LD3, LD4	CC	1.4 ms	1.4 ms	1248 byte	1
Audio	ME	S1, S2, S3, S4	250 us	250 us	80 byte	1
VideoInfot.	ME	RS1, RS2	33.33 ms	33.33 ms	1500 byte	119
Ultrasonic	US1, US2, US3, US4	CC	100 ms	2 ms	188 byte	1
Camera	CM1, CM2, CM3, CM4	CC	16.66 ms	16.66 ms	1500 byte	119
HeartBeat	CM1, CM2, CM3, CM4, MC1, MC2, RD1, RD2, RD3, RD4, LD1, LD2, LD3, LD4, US1, US2, US3, US4, ZC1, ZC2, ZC3, ZC4, TLM, CU	CC	10ms	10ms	64 byte	1
MirrorCamera	MC1, MC2	CC	16.66ms	16.66ms	1500 byte	119
DriverCamera	DC	CC	16.66ms	16.66ms	1500 byte	119
RearCamera	RC	CC	16.66ms	16.66ms	1500 byte	119
Radar	RD1, RD2, RD3, RD4	CC	2500us	2000us	625 byte	1

Nome	Src	Dst	Periodo	Deadline Rel.	Payload	Burst
AdasSensors1	ZC1, ZC2, ZC3, ZC4	CC	10 ms	1 ms	1000 byte	10
AdasSensors2	ZC1, ZC2, ZC3, ZC4	CC	20 ms	1.5 ms	1000 byte	10
TelemMap	TLM	CC	6ms	6ms	1500 byte	1
Telematics	TLM	HU, RS1, RS2	625us	625us	600 byte	1
Control1	CC, CU	CC, CU	200ms	200ms	64 byte	1
Control2	CC, CU	CC, CU	5ms	200ms	64 byte	1
Control3	CC, CU	CC, CU	50ms	200ms	64 byte	1
Control4	CC, CU	CC, CU	100ms	200ms	64 byte	1
Control5	CC, CU	CC, CU	500ms	200ms	64 byte	1
Control6	CC, CU	CC, CU	1ms	200ms	64 byte	1
DACAM	CC	HU	166us	166us	1500 byte	1

4.4. Results

The implementation of additional nodes and switches in the initial design caused frame loss to skyrocket. This is because the mechanisms used previously were designed for that type of network, making the system inflexible.

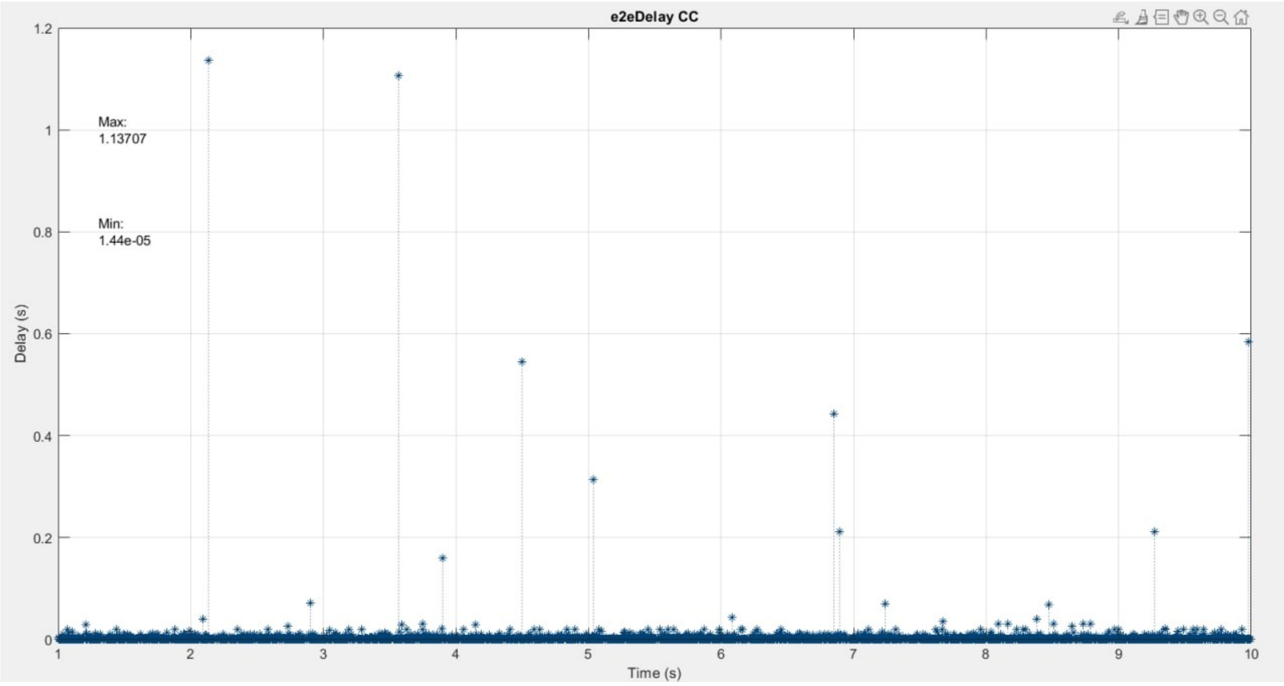
the results obtained show FLRs almost always greater than 1, even in the case of nodes that transmit video frames. This is for the reasons explained above.

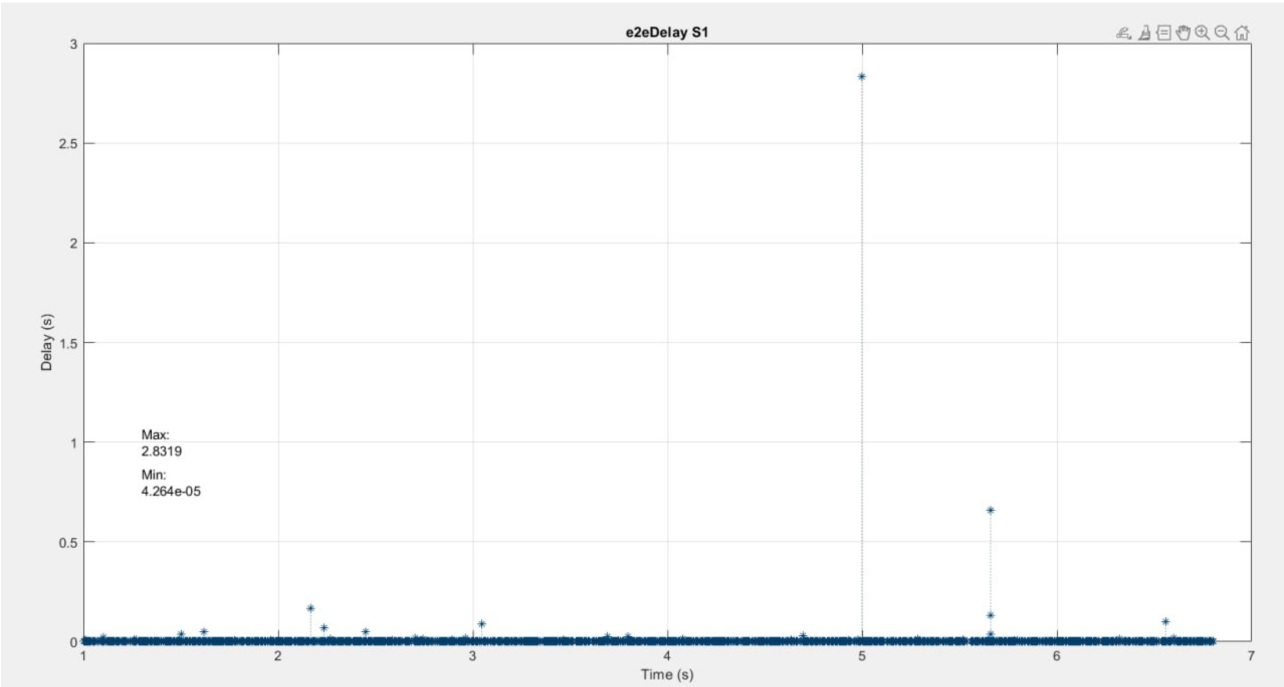
After several attempts and appropriate changes to the code, to reset the FLR, the considerations that can be extracted from the results are:

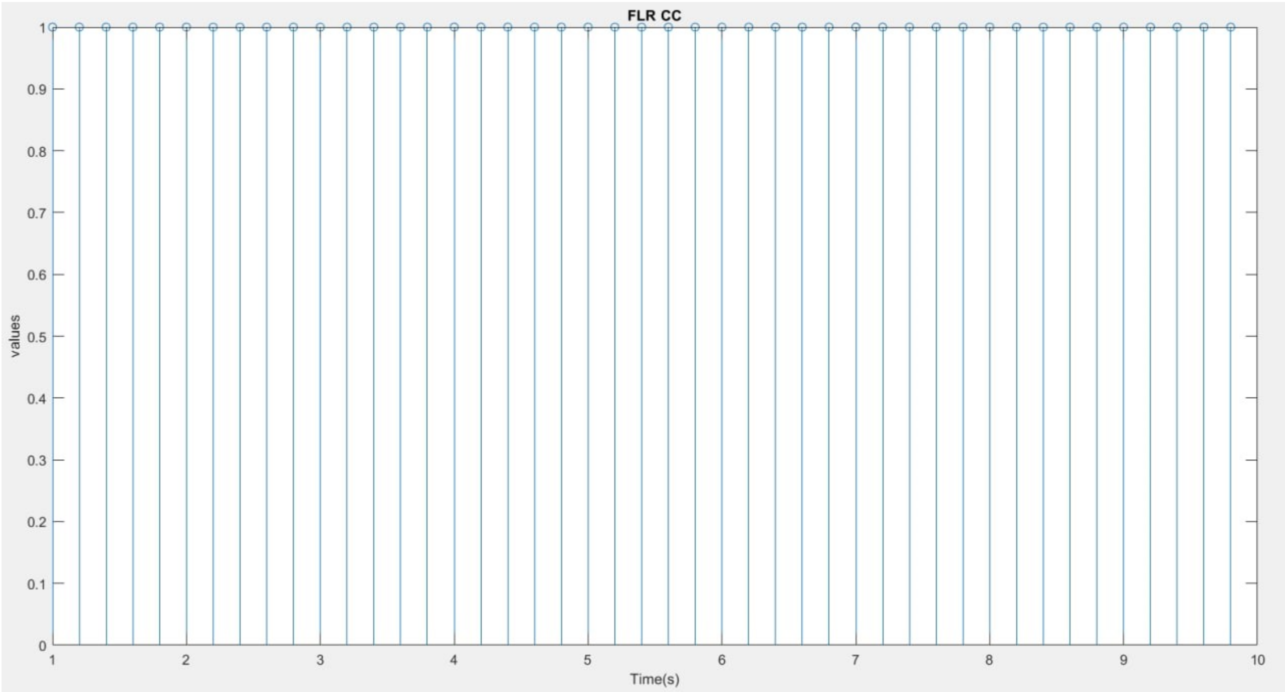
that the approaches used and studied were only able to minimize the FLR, without ever being able to obtain a value of lost frames equal to zero or so small compared to the transmitted frames to obtain a zero FLR.

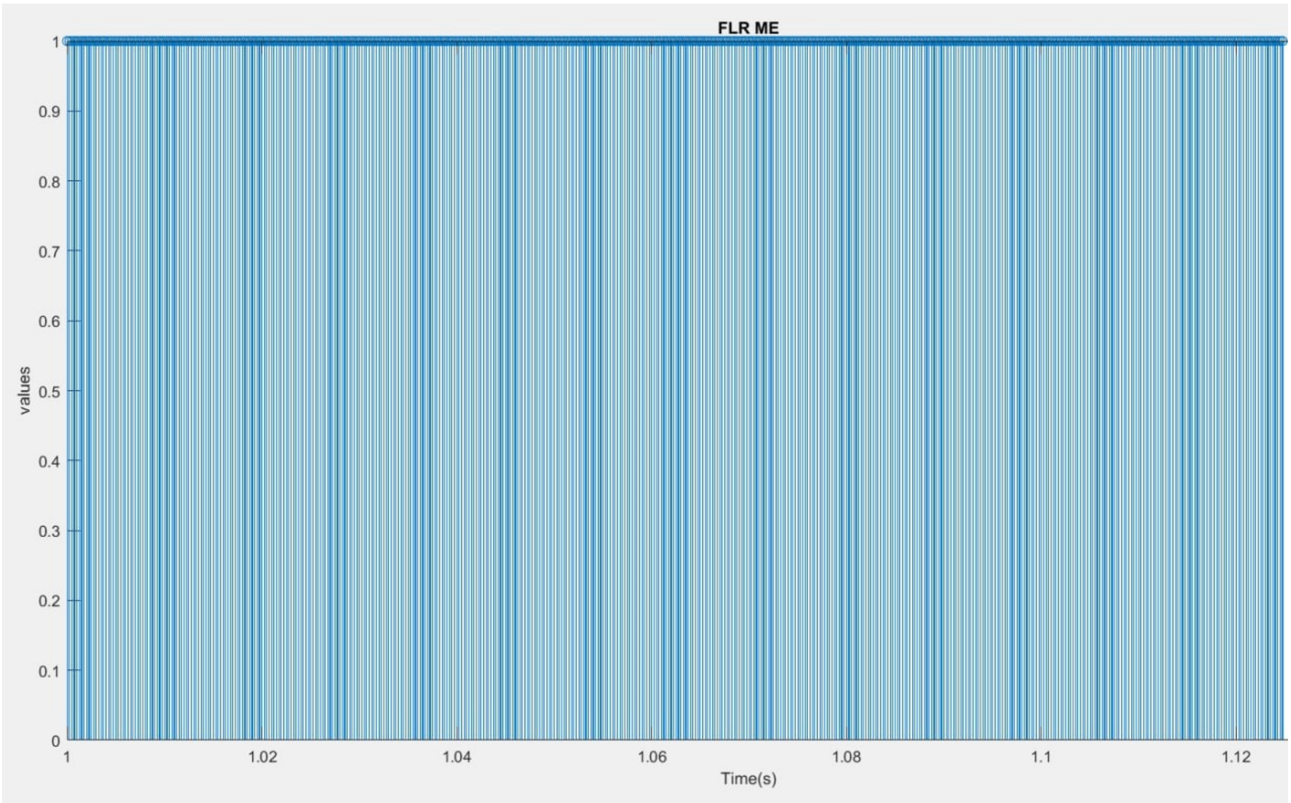
The results obtained from the end-to-end delay vary from node to node, compared to the trend seen in the initial project which returned similar trends for the various nodes.

4.5. Graphs






FLR CC
values
Time(s)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
9
10


FLR ME
values
Time(s)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
1.02
1.04
1.06
1.08
1.1
1.12