MSE | MASTER OF SCIENCE IN ENGINEERING

Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Report of DeseNET Laboratory

DeSEm – Design of communicative embedded systems

Schnyder Nicolas

Delivery date: 11.01.2023

Content

1 Introduction

1.1 Objective

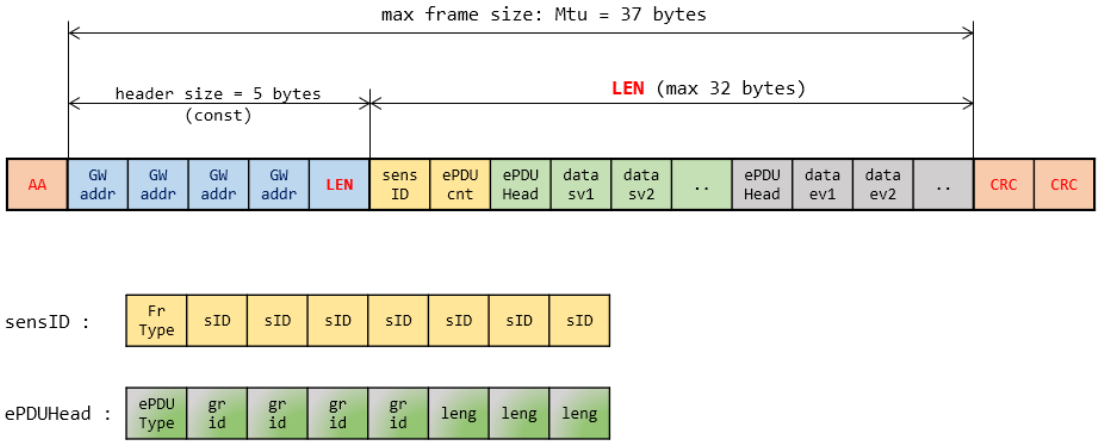
The DeseNET protocol is a real-time communication protocol, developed by the Hes-so Valais/Wallis, for a reliable and low-powered data exchange over Wi-Fi.[1]

The objective of this laboratory is to implement the necessary software requirements, to establish a functional connection between a master gateway and a slave sensor. Thereby, the sensor consists of a 3-axis accelerometer and a 4-button joystick (up, down, left, right, centre).

To facilitate the development process a "Mesh simulator" was provided, so that the real target (a NUCLEO L476RG evaluation board) can be simulated. For both, simulator and target, several test conditions were defined and conducted to ensure the proper functioning of the proposed solution.

1.2 DeseNET structure

The DeseNET protocol requires to send so-called MPDUs (Multi Protocol Data Units) in the following format:



Legend:

AA, CRC	Unused
GW addr	Gateway address (const)
LEN	Usable data length (max 32 bytes)
Sens ID	Sensor ID (const)
ePDU cnt	Number of ePDUs (embedded PDUs) in the frame
ePDU head	Header of an ePDU (type, SV group or EV ID, length)
data sv, ev	SV or EV data

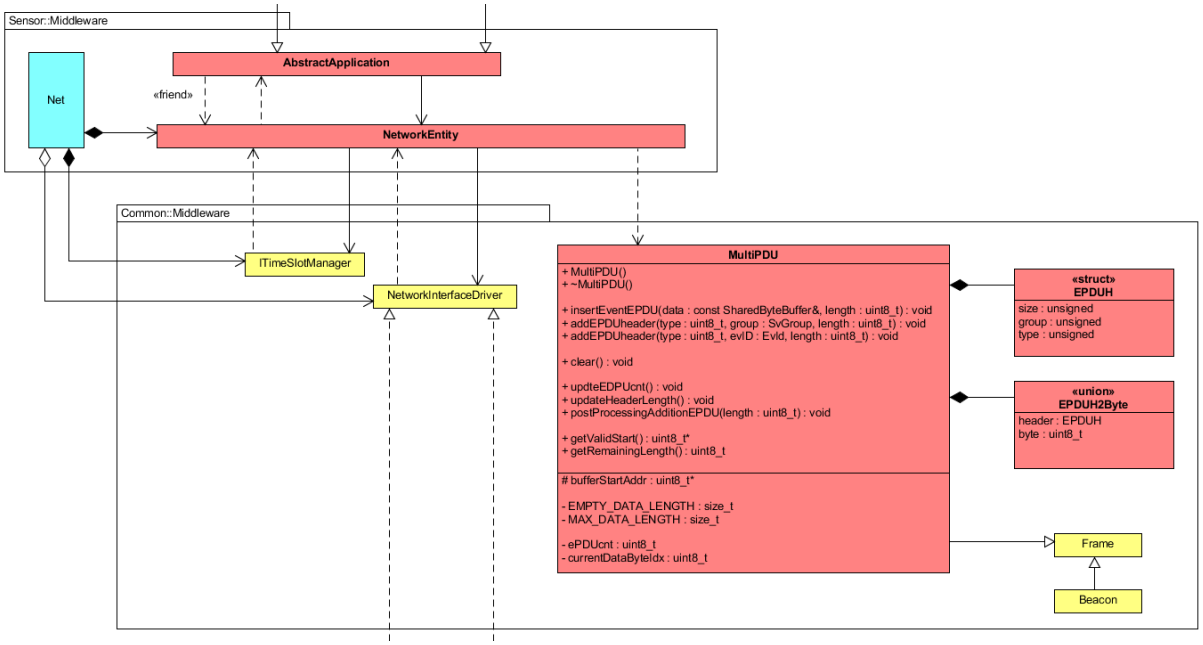
Hence, the general objective is to create and send such frames.

[1] https://moodle.msengineering.ch/pluginfile.php/228006/mod_resource/content/1/Desem2Definition.pdf (accessed 05.01.2023)

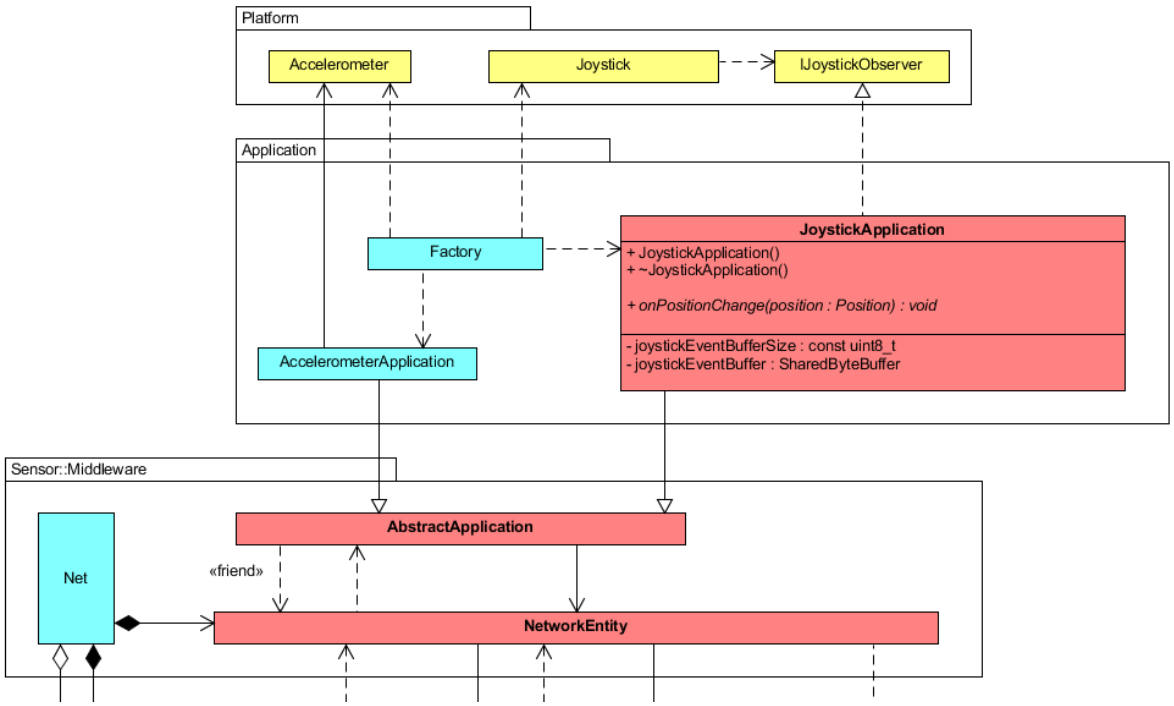
2 Architecture

The most part of the system was already working and handed-out to form the template of the work. Hence, only the self-made classes and their interaction with some existing classes are described here-below. The complete diagram can be found in the delivery folder in PDF format.

2.1 Multi PDU



2.2 Joystick Application

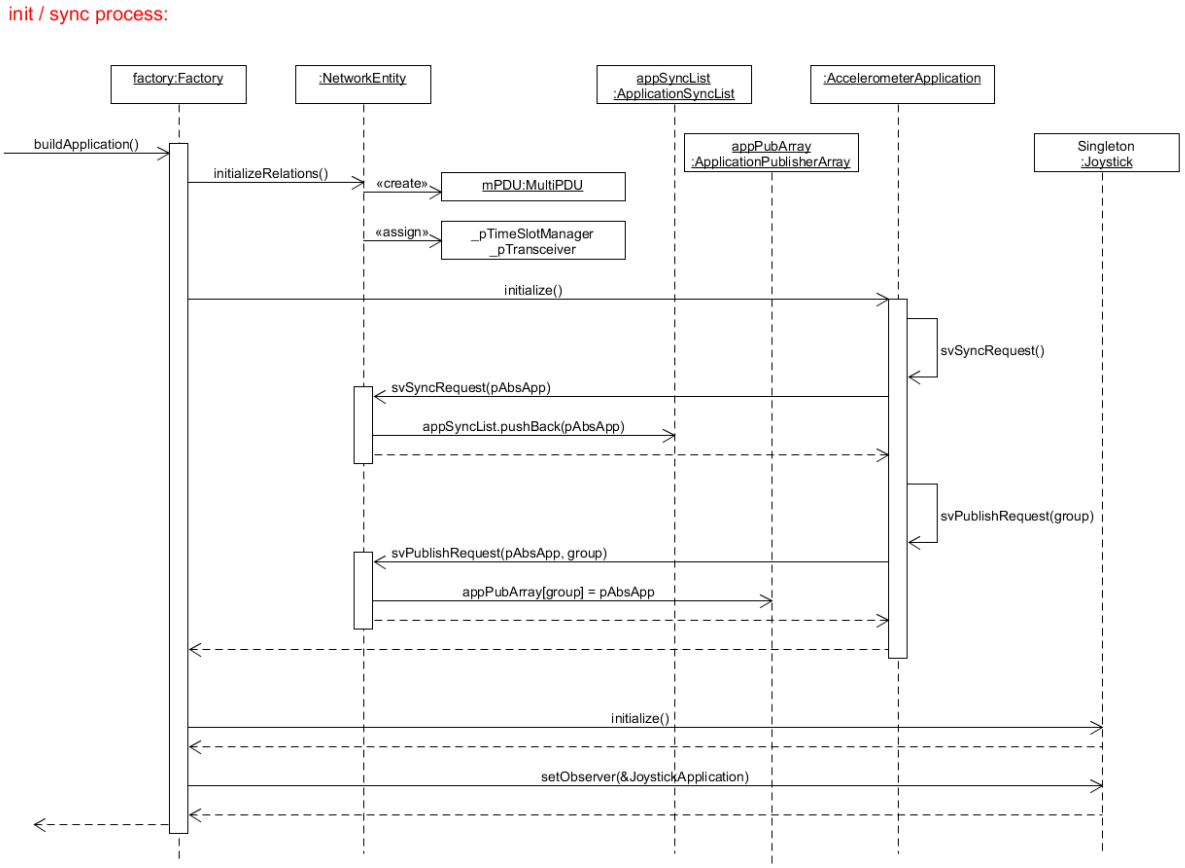


3 Design

The following sub-chapters give a brief description on how the corresponding values are acquired and transmit via the DeseNET protocol.

3.1 Initialization and synchronization

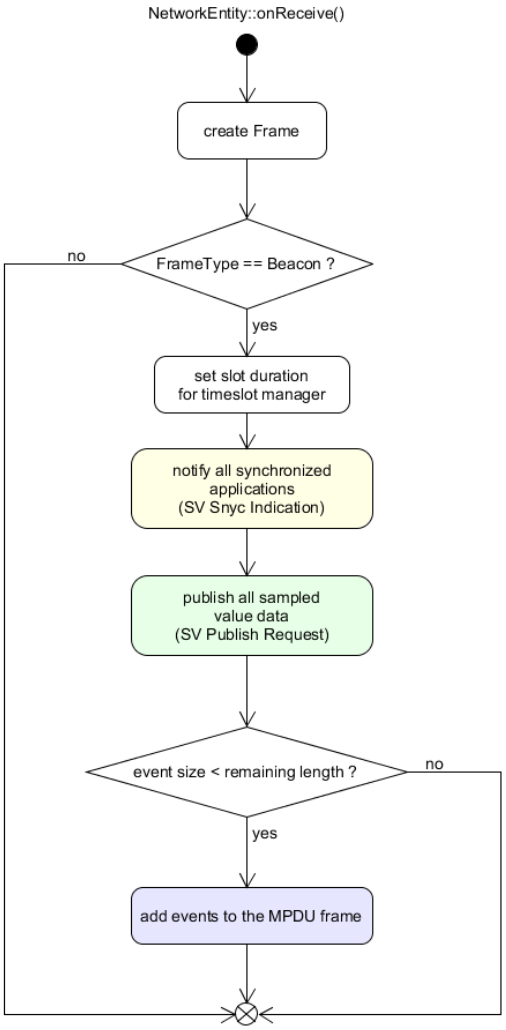
A general view of the initialization process is shown in the sequence diagram below. The corresponding diagram in PDF can be found in the delivery folder.

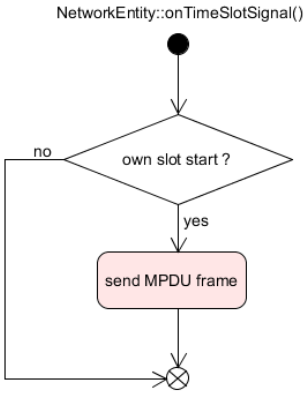


3.2 Beacon reception

The principle of the main sequence is shown in the flowcharts below. Thereby, the chart on the left describes the algorithm when a frame is received by the method “onReceive” of the NetworkEntity. The chart on the right shows the process to send the MPDU frame by the means of the “onTimeSlotSignal” method, which is called at a timer event, launched in “onReceive”.

The corresponding detailed sequence diagram in PDF can be found in the delivery folder.





4 Tests

4.1 Scenarios

To validate the implemented solution, the test scenarios in the table below were defined. Thereby, the demo sensor formed the reference for a correct behaviour and the general frame structure is like in § 1.2.

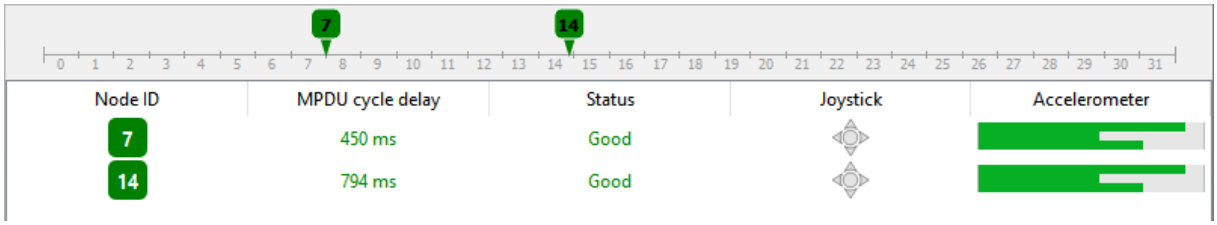
#	Test	Expected Result
1	Send single Beacon for sampled accelerometer values Objective: Test correct data and correct reception time	**Header:** LEN = 0x09 **ID + Cnt:** sensID = 0x87 (Type = 1, ID = 7) ePDUcnt = 0x01 **SV ePDU:** ePDU Head = 0x16 (Type = 0, SVgroup = 2, Length = 6) dataSV = 0xC406Ca00C703 (top left corner) **EV ePDU:** n/a **Timing:** Response time btw 400 [ms] and 450 [ms]
2	Send single Beacon for single event data Objective: Test correct data and correct reception time	**Header:** LEN = 0x0D **ID + Cnt:** sensID = 0x87 (Type = 1, ID = 7) ePDUcnt = 0x03 **SV ePDU:** n/a **EV ePDU:** ePDU Head #1 = 0xC1 (Type = 1, EV ID = 8, Length = 1) dataEV #1 = 0x20 (centre button pressed) ePDU Head #2 = 0xC1 (Type = 1, EV ID = 8, Length = 1) dataEV #2 = 0x01 (button released) **Timing:** Response time btw 400 [ms] and 450 [ms]

3	Send single Beacon for multiple event data without sampled values Objective: Test correct event data without over-charging the MPDU and that the SV group subscription works	**Header:** LEN = 0x20 (max) **ID + Cnt:** sensID = 0x87 (Type = 1, ID = 7) ePDUcnt = 0x0F **SV ePDU:** n/a (unsubscribed) **EV ePDU:** ePDU Head #odd = 0xC1 (Type = 1, EV ID = 8, Length = 1) dataEV #1 = 0x20 (centre button pressed) ePDU Head #even = 0xC1 (Type = 1, EV ID = 8, Length = 1) dataEV #2 = 0x01 (button released) **Timing:** Response time btw 400 [ms] and 450 [ms]
4	Send beacons continuously for sampled accelerometer values Objective: Test correct SV data multiple times	Same as #1 (additionally check multiple slot durations) Qualitative check by moving the sensor window as well.
5	Send beacons continuously for event data Objective: Check correct data EV data by pressing each button	**Header:** LEN = 0x0C **ID + Cnt:** sensID = 0x87 (Type = 1, ID = 7) ePDUcnt = 0x02 **SV ePDU:** n/a **EV ePDU:** ePDU Head #1 = 0xC1 (Type = 1, EV ID = 8, Length = 1) dataEV #1 = 0x20 (centre button pressed) dataEV #2 = 0x04 (right button pressed) dataEV #3 = 0x10 (down button pressed) dataEV #4 = 0x02 (left button pressed) dataEV #5 = 0x08 (up button pressed) **Timing:** Response time btw 400 [ms] and 450 [ms]

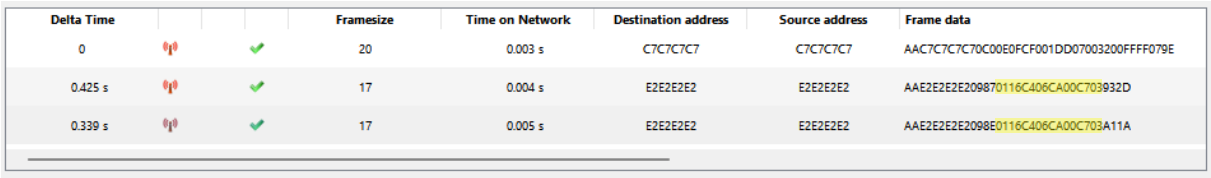
4.2 Results on Mesh simulator

#	Test	Verdict	Remark
1	Send single Beacon for sampled accelerometer values	OK	--
2	Send single Beacon for single event data	OK	--
3	Send single Beacon for multiple event data	OK	--
4	Send beacons continuously for sampled accelerometer values	OK	--
5	Send beacons continuously for event data	OK	--

As a qualitative verification, the following screenshot shows the comparison between the demo sensor (ID # 14) and the tested sensor (ID # 7). The detailed test data can be found in Appendix § 6.1.

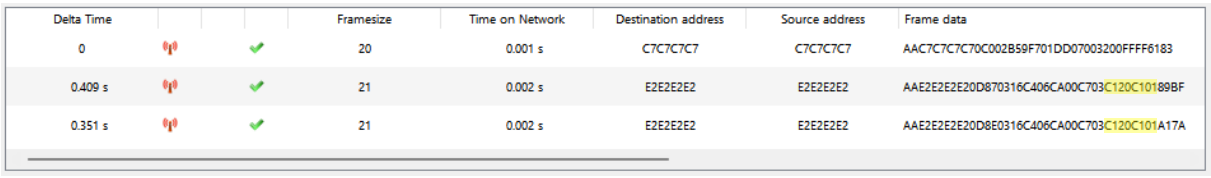


Accelerometer values comparison (top left corner)



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0			20	0.003 s	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00E0FCF001DD07003200FFFF079E
0.425 s			17	0.004 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987**0116C406CA00C703**932D
0.339 s			17	0.005 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E2098E**0116C406CA00C703**A11A

Joystick Values comparison (pressed and released centre button)



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0			20	0.001 s	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C002B59F701DD07003200FFFF6183
0.409 s			21	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20D870316C406CA00C703**C120C101**89BF
0.351 s			21	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20D8E0316C406CA00C703**C120C101**A17A

General remark

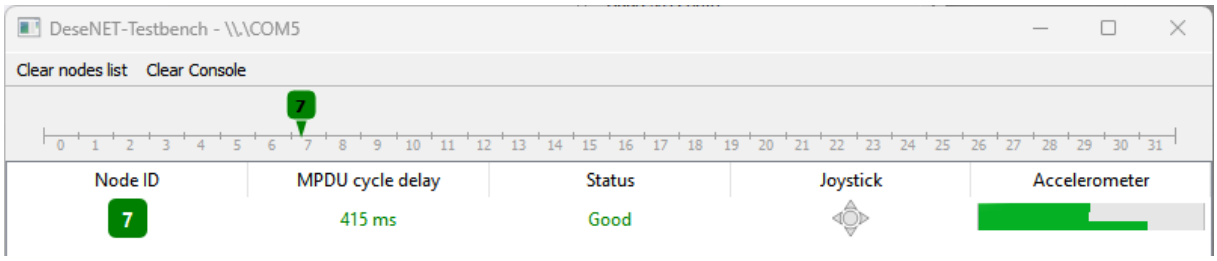
Sometimes, the sent responses of the tested sensor and the demo sensor were slightly out of range (around 10 ms too late). This can be explained by the fact that in simulation mode, the operation system of the used PC works sometimes too slow.

4.3 Results on real target

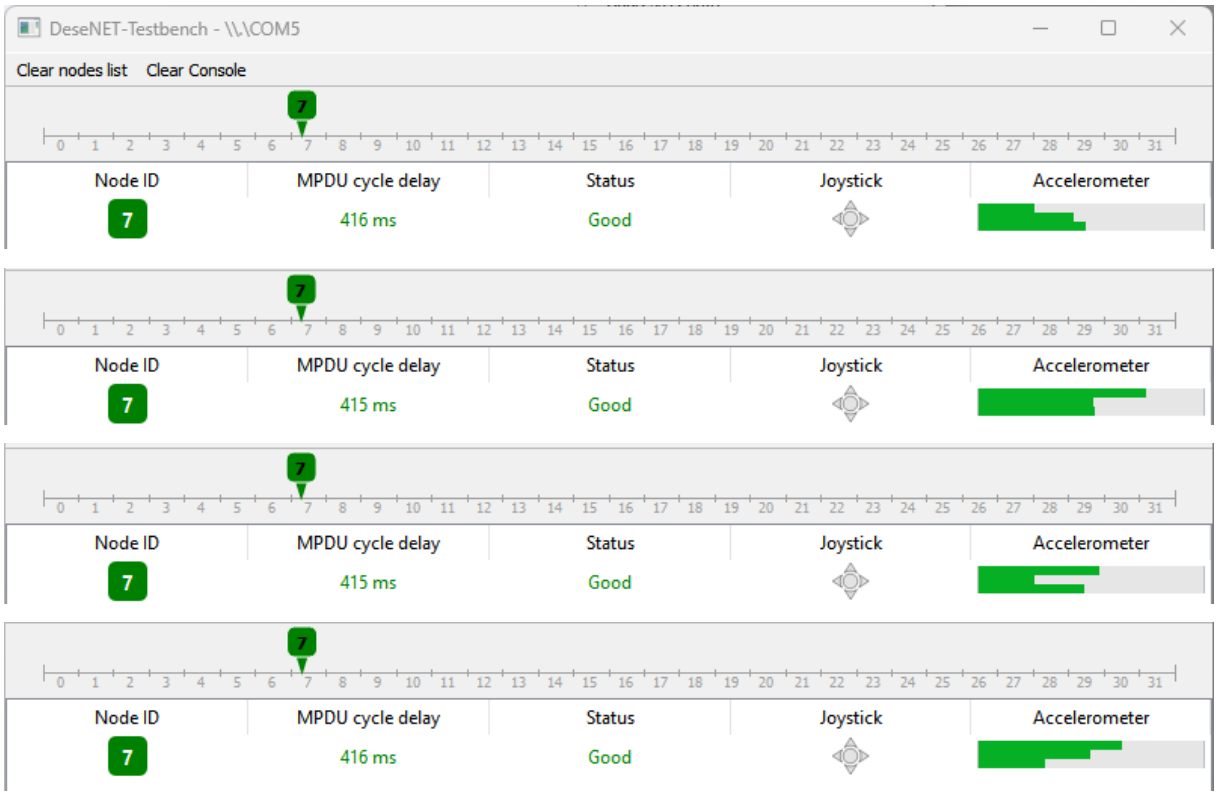
Because the sent and received frames cannot be displayed with the real target, the correct functionality is verified with the displayed icons of the testbench.

Accelerometer SV data

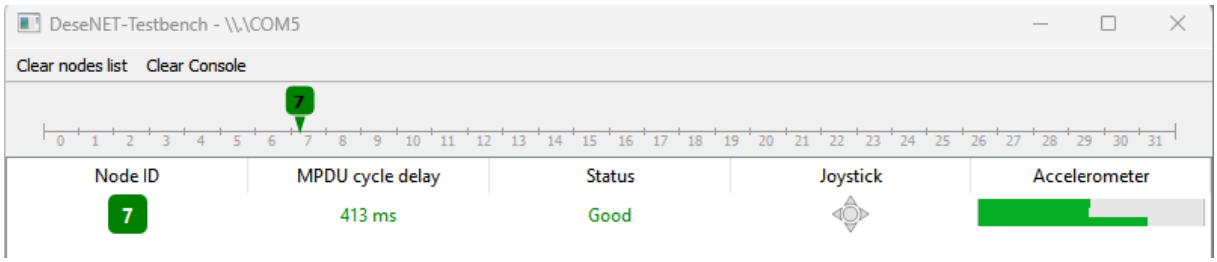
Nucleo board still on table → OK



Moving board in different positions → OK



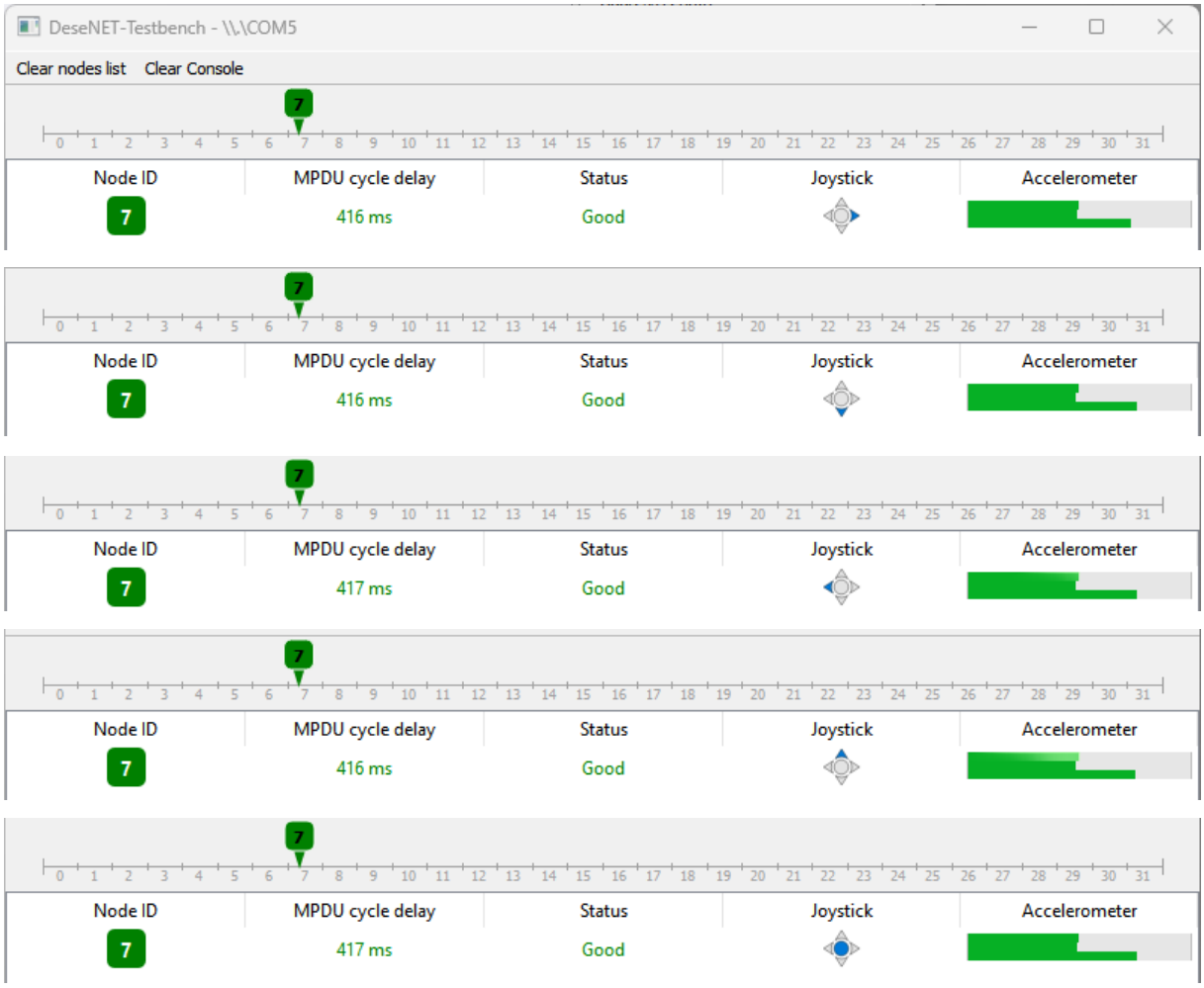
Unsubscribe accelerometer SV group (#2) and moving board → OK



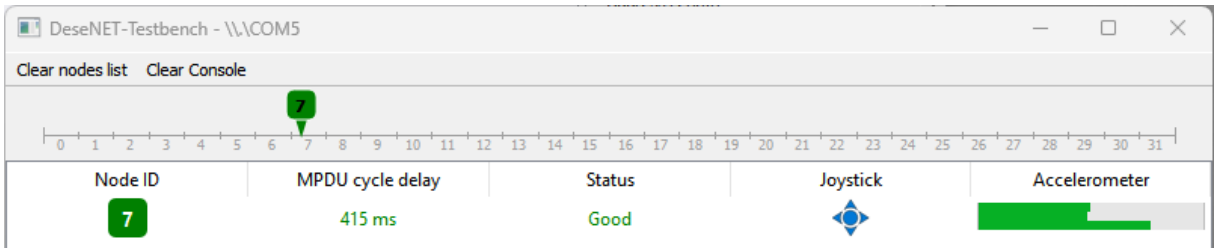
⟹ Accelerometer values freeze due to unsubscription

Joystick EV data

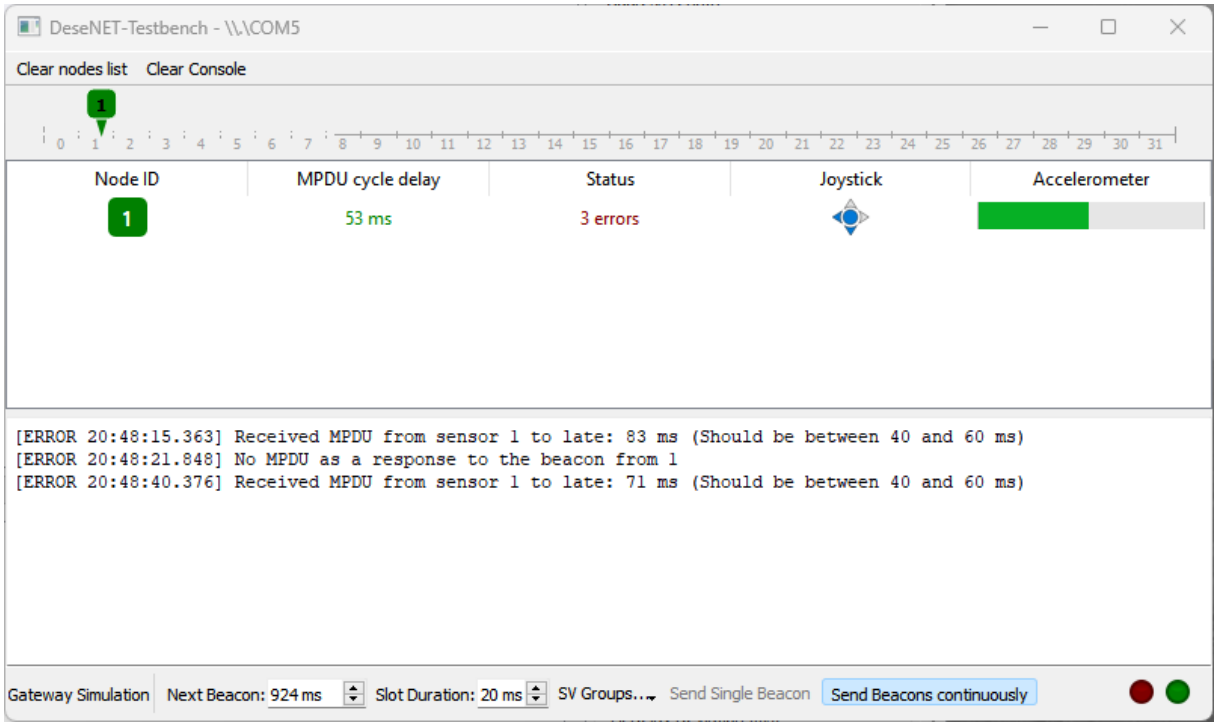
Single button check → OK



Multiple button presses→ OK



Checking time limit → OK



⟹ Even by spamming the button, the time slot was respected for most of the time

General Remark

On the real system, the error message “No MPDU as a response to the beacon from 7” shows up sometimes. This error is due to the air as a transmission medium, which is perturbed by other devices.

The MPDU frame was never over-charged, hence it showed always coherent data with a delay of maximal 2 beacon cycles.

5 Conclusion

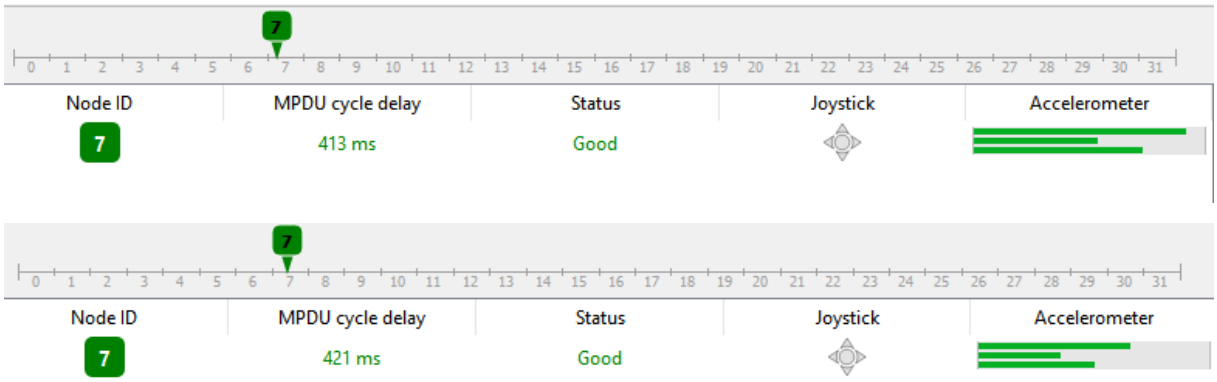
The embedded software solution, proposed in this document, showed a solid functioning. It fulfilled the requirements of the test scenarios in the simulated environment, as well as on the real target.

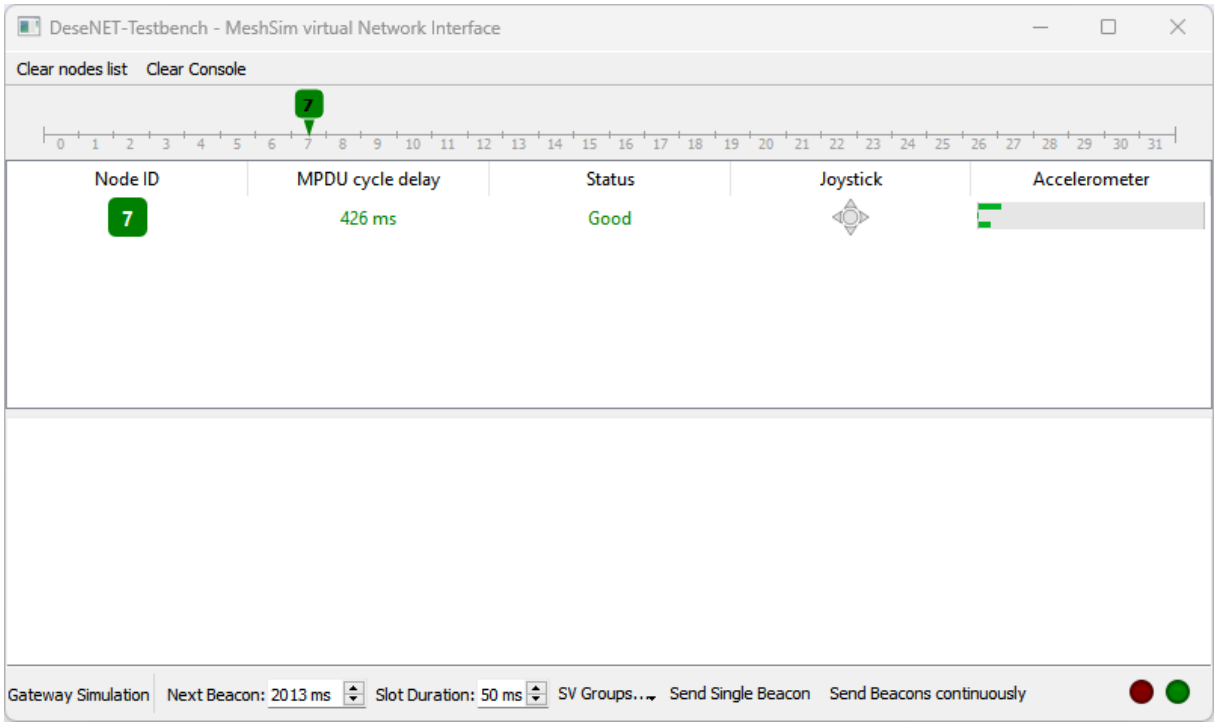
Nevertheless, some small issues could have been observed during the tests under real conditions. This was due to the perturbations in the air. However, the Nucleo board never lagged, froze or got stuck. This shows that a satisfying solution has been developed.

6 Appendix

6.1 Frame observation of Mesh simulator test series

6.1.1 Test 1: SV accelerometer data with single beacon

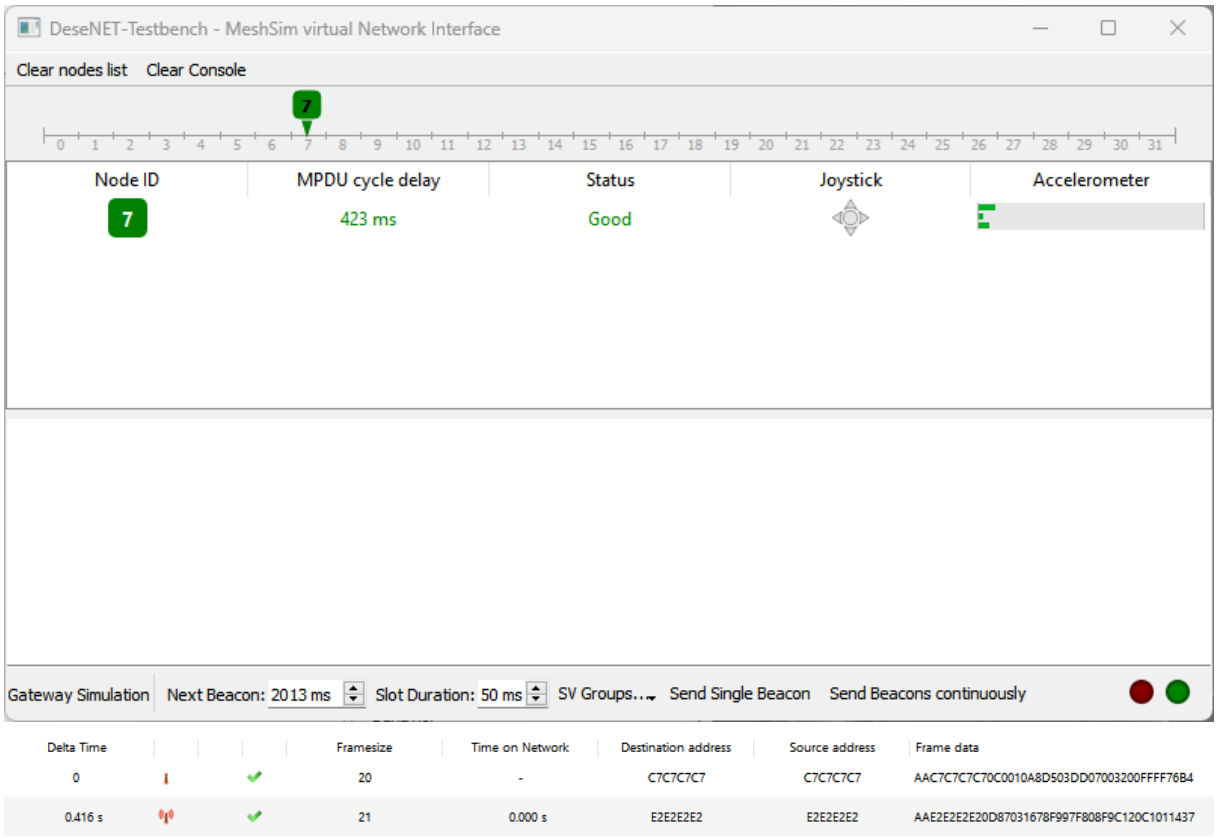




Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0003B48204DD07003200FFFF9EBE
0.406 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116C406CA00C703932D
154.134 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00AE0F8504DD07003200FFFF71F2
0.409 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116AA02DDFD43003ED2
30.438 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C002E888504DD07003200FFFF85EA
0.411 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116D6F950F813F93488

6.1.2 Test 2: Single EV joystick data with single beacon

Press + Release centre button

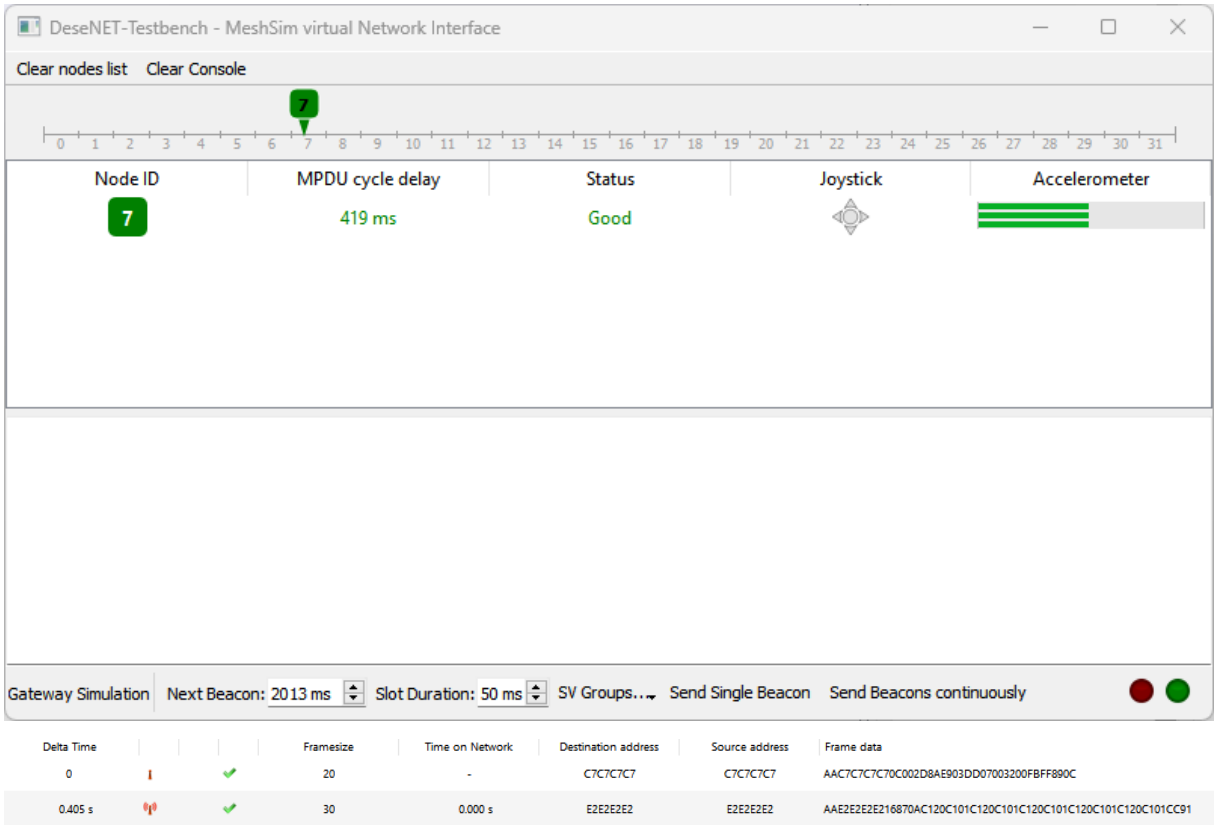


6.1.3 Test 3: Multiple EV joystick data with single beacon (SV group 2 disabled)

```
MAX_EVELM_NBR = 30            // maximal event number to avoid spam
CUTOFF_EVELM_NBR = 4          // number of elements to be cut after having reached MAX_EVELM_NBR
```

Press the centre button 5 times

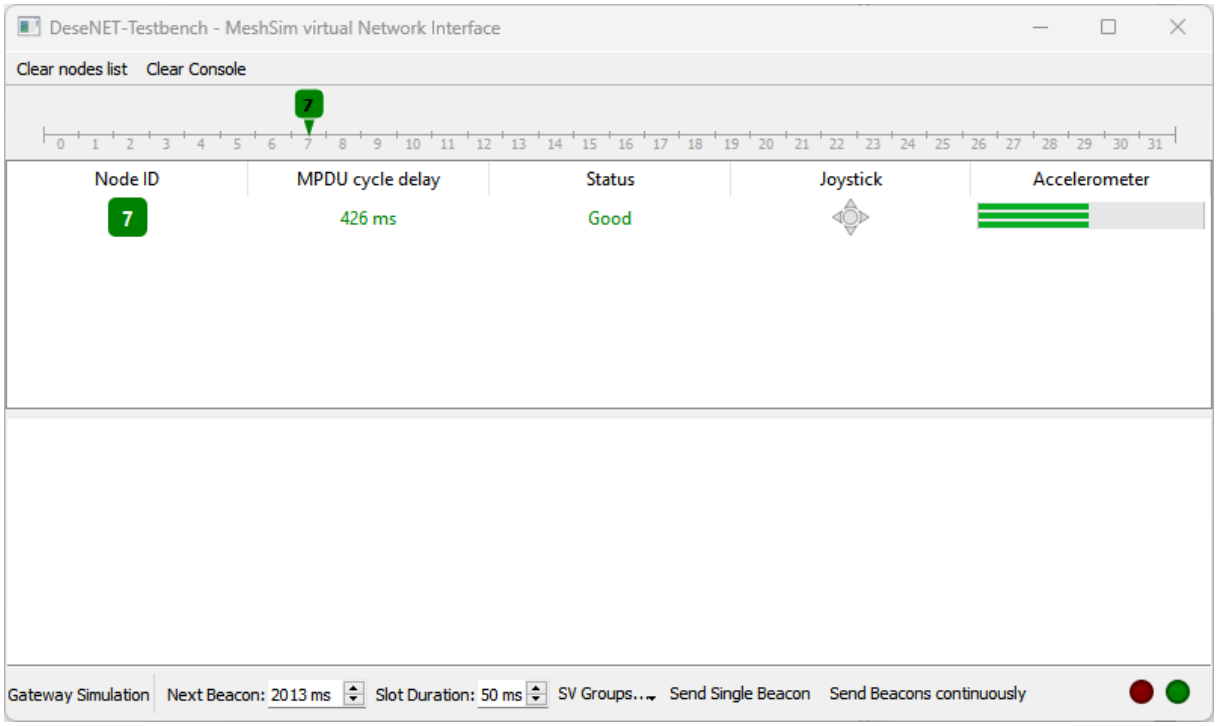
⟹ The captured data is sufficient to take place into one single buffer



Press the centre button 15 times

⟹ The captured data is sufficient to take place into exactly two buffers.

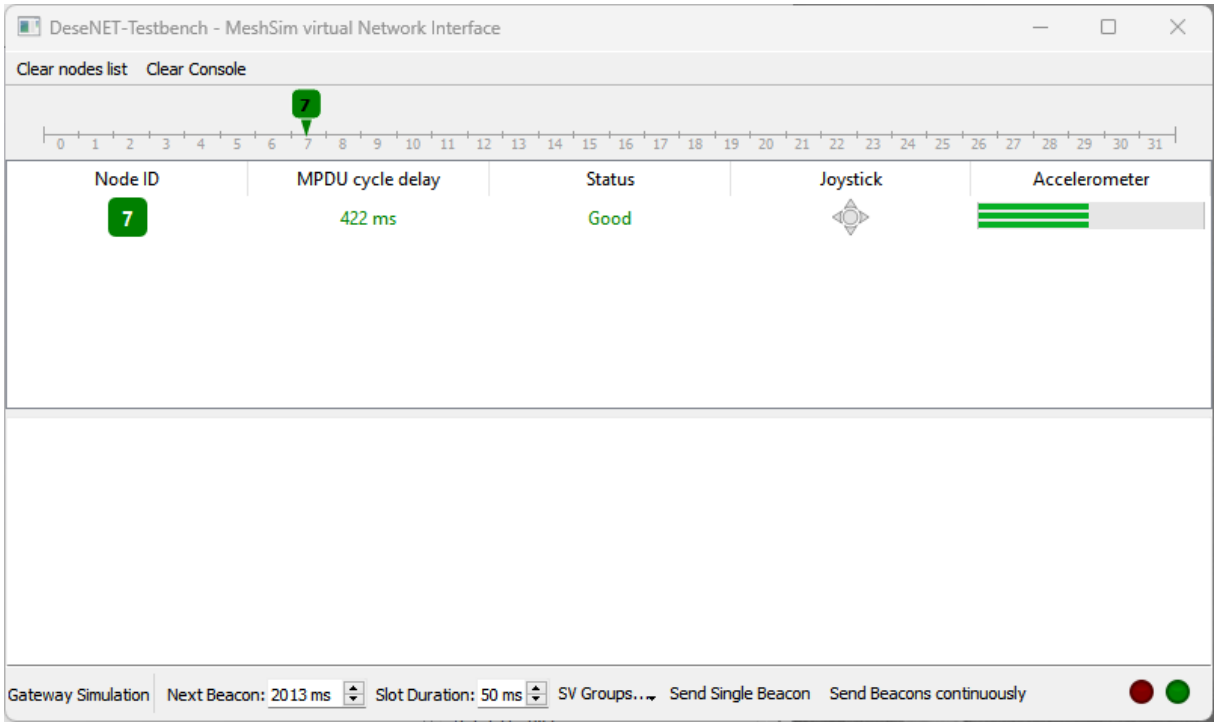
Because of press and release: 2 * 15 Events = 30 Events = MAX_EVELM_NBR



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
43.707 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0089EDEE03DD07003200FBFF7711
0.412 s			40	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E220870FC104C101C104C101C104C101C104C101C104C101C104C101C104C101C104AEEE
4.121 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C003DFFEE03DD07003200FBFFA28C
0.412 s			40	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E220870FC101C104C101C104C101C104C101C104C101C104C101C104C101C104C101D518
3.359 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00F80DEF03DD07003200FBFF6E8F
0.414 s			10	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E2028700D7F8

Press the centre button 16 times

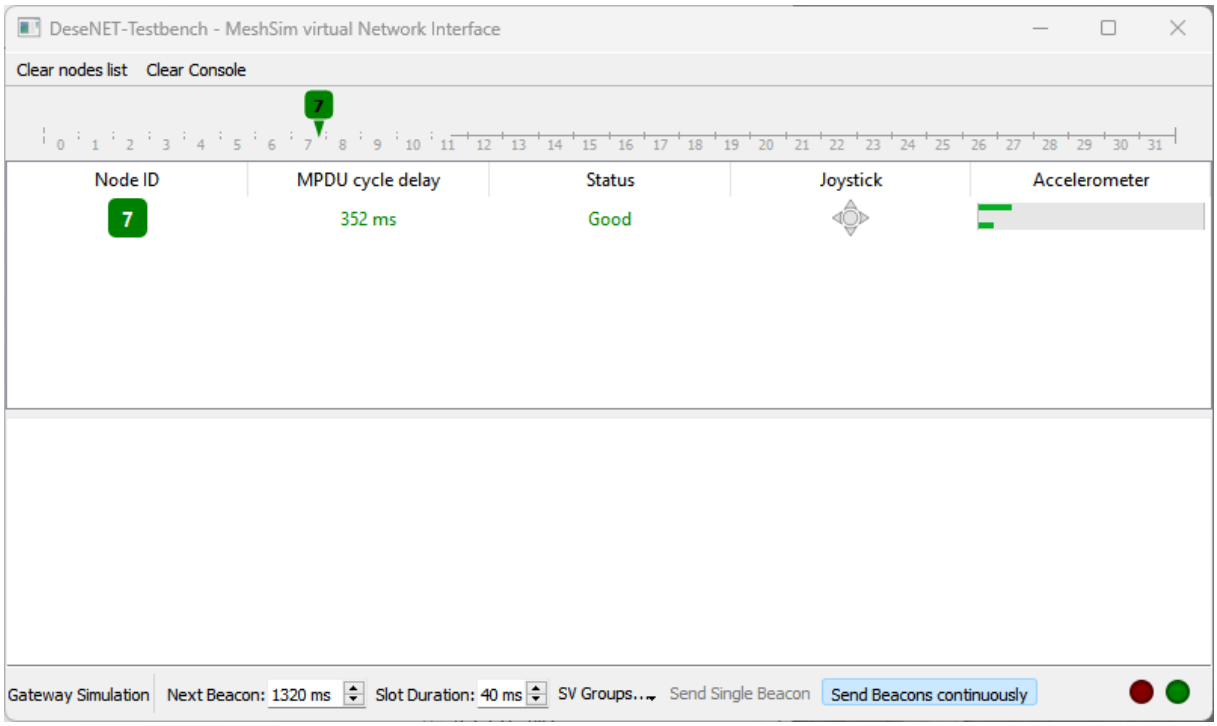
⟹ The captured data is sufficient to take place into exactly two buffers and the “spam delimiter” resizes the event list down to 26 elements (30 Events – CUTOFF_EVELM_NBR)



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0027F2F503DD07003200FBFF678A
0.409 s			40	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E220870FC120C101C120C101C120C101C120C101C120C101C120C101C120039D
2.667 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C002AFEF503DD07003200FBFF0951
0.408 s			36	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E21C870DC101C120C101C120C101C120C101C120C101C120C101C120C101ACB2
2.804 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00B50AF603DD07003200FBFF1DB2
0.409 s			10	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E2028700D7F8

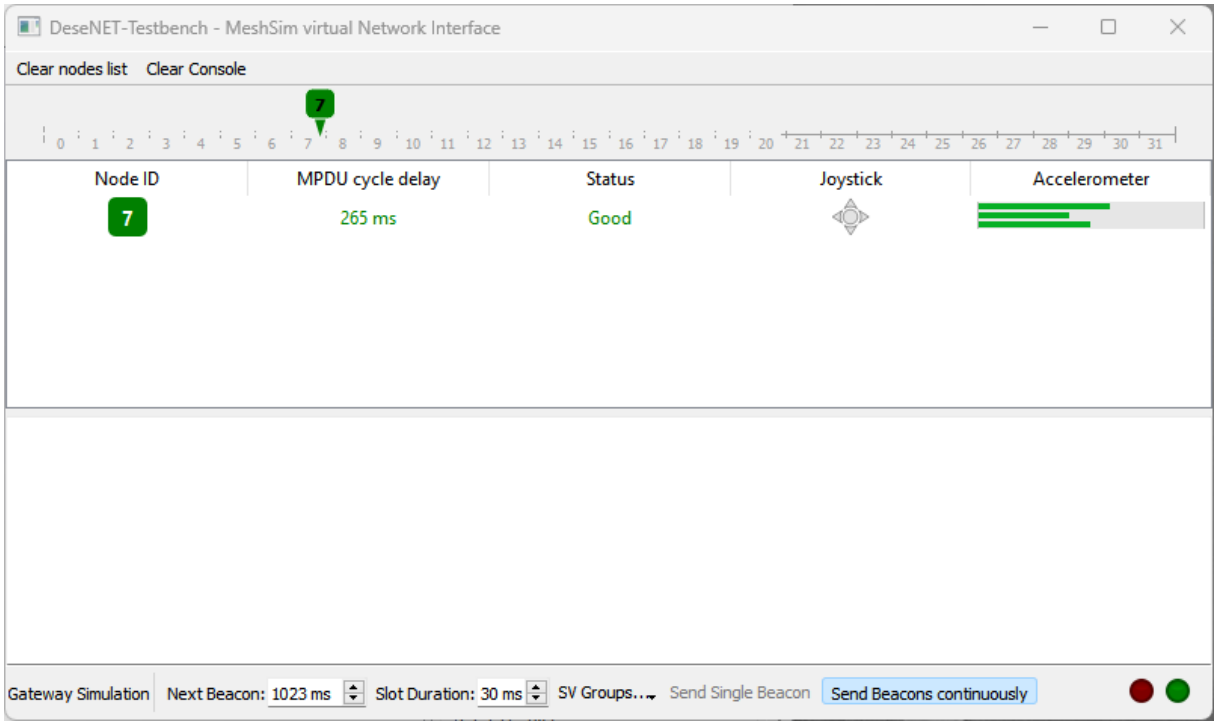
6.1.4 Test 4: SV accelerometer data with multiple beacons

Time Slot Duration: 40 ms



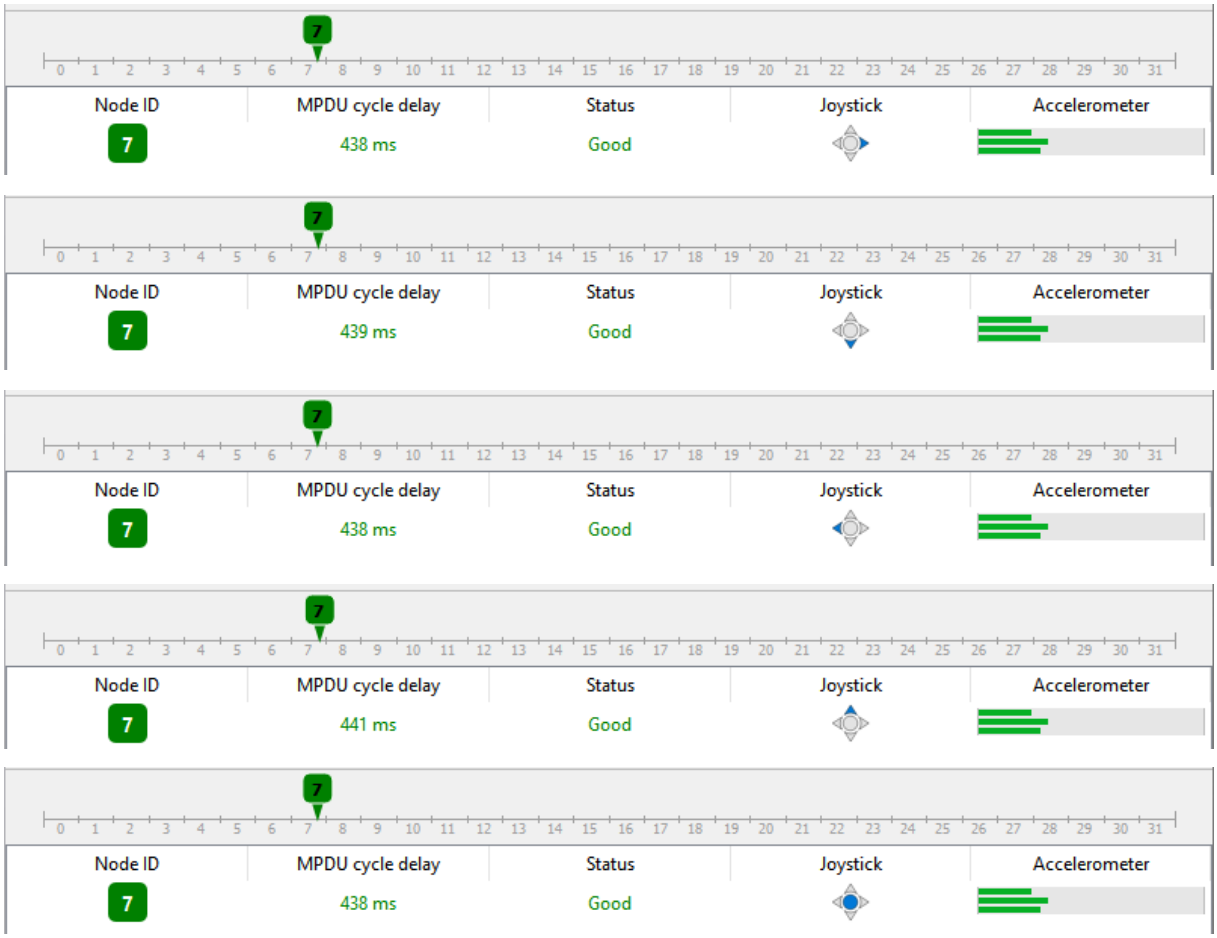
Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00430A0E042805002800FFFFFA02
0.327 s		✔	17	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116A5FA04F855F95574
0.995 s		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C006E0F0E042805002800FFFF335D
0.330 s		✔	17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116A5FA04F855F95574
0.994 s		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C009A140E042805002800FFFF6ECB
0.335 s		✔	17	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E209870116E9FEC3FCD6FDA458
0.988 s		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00C5190E042805002800FFFF3B55
0.339 s		✔	17	0.003 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E2098701168FFF50FAF0FC4BE6
0.985 s		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00F11E0E042805002800FFFF323A
0.339 s		✔	17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011652FC4AFBCEFB3BE8
0.984 s		✔	20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C001C240E042805002800FFFF7740
0.337 s		✔	17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E2098701163D0330FEB600BC84

Time Slot Duration: 30 ms



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
0.758 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0081D10904FF03001E00FFFFF71B
0.253 s			17	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00
0.773 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0083D50904FF03001E00FFFFECA5
0.253 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00
0.772 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0084D90904FF03001E00FFFF95F6
0.253 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00
0.773 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0087DD0904FF03001E00FFFFDBD9
0.254 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00
0.772 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C0088E10904FF03001E00FFFF60C7
0.254 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00
0.775 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C008DE50904FF03001E00FFFFDB9F
0.253 s			17	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20987011686019FFE12009D00

6.1.5 Test 5: EV joystick data with multiple beacons



Delta Time			Framesize	Time on Network	Destination address	Source address	Frame data
1.610 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C000E6A1304DD07003200FFFFFF4A
0.408 s			19	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20B870216A7FB9FFA23FBC104322D
1.606 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00ED711304DD07003200FFFF737F
0.412 s			21	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20D870316A7FB9FFA23FBC101C110FFEB
1.603 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00CD791304DD07003200FFFFF8DC
0.414 s			19	0.002 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20B870216A7FB9FFA23FBC1016580
1.602 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C00AD811304DD07003200FFFF3310
0.414 s			19	0.000 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20B870216A7FB9FFA23FBC102571B
1.602 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C008D891304DD07003200FFFFB8B3
0.416 s			21	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20D870316A7FB9FFA23FBC101C1086322
1.599 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C006C911304DD07003200FFFFEBCD
0.422 s			21	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20D870316A7FB9FFA23FBC101C120CE68
1.594 s			20	-	C7C7C7C7	C7C7C7C7	AAC7C7C7C70C004C991304DD07003200FFFF606E
0.420 s			19	0.001 s	E2E2E2E2	E2E2E2E2	AAE2E2E2E20B870216A7FB9FFA23FBC1016580