ETALON

D3.2 On-Train Communication Systems and RF Components Report

Due date of deliverable: 31/08/2018

Actual submission date: 15/10/2018

Leader of this Deliverable: Alexander James Pane, ISMB

Reviewed: Yes

Document status		
Revision	Date	Description
1	25.06.18	First description
2	27.06.18	WSN optimization options added. Updated the acronyms tables
3	06.07.18	Added the SN-DS interface specifications. Modified the acronyms and slight changes to the section titles
4	16.07.18	Reported the document in the ETALON default template. Various modification to the document content and layout. Modified the acronyms
5	17.07.18	Updated the images
6	20.08.18	Updated the contents
7	17.09.18	Integration of the contributions
8	18.09.18	Added the simulation Section
9	26.09.2018	Final review and general control check

Project funded from the European Union’s Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/09/2017 Duration: 30 months

REPORT CONTRIBUTORS

Name	Company	Details of Contribution
Alexander James Pane	ISMB	Document generation and various updates and contributions
Alexander James Pane	ISMB	Document revision and change of template
Simone Ciccia	ISMB	Added contribution for the antenna design
David Vincent	PER	Added contribution for radio bands licensing
Veronika Nedviga and Carles Artigas	ARD	Added contributions for various aspects of the document
Paul Hyde	UNEW	Added contributions for various aspects of the document
Roberto Cafferata	SIRTI	Performed the last editing check and final revision

EXECUTIVE SUMMARY

The following document reports the solution for the wireless communication for the on-board TI (Train Integrity) check.

The deliverable covers various points:

- Review of the requirements for the communication system.
- The WSN communication structure:
    - Discovery of the wanted nodes
    - Generation of the communication backbone
- TI check through a distance measuring between WSN nodes.
- Characterization of the antenna for both communication and distance measuring
- Analysis of the system working in the railway environment.
- Simulations of the communication protocol.

The proposed solution aims at minimizing the human-machine interaction by automating the whole process, limiting the user interaction to only the initialization process, where the interaction is limited to the wireless communication initialization.

The proposed wireless communication solution can potentially be in two major states:

- Node discovery – unknown topology and searching for all peers
- Active communication protocol – all wanted nodes have been discovered and a wireless communication link has been established

During the “Node discovery” state, TI cannot be checked. Train parameters are still unknown.

During the “Active communication protocol” state, TI can be checked. Train parameters are known.

The network presents robustness features, such as SW and HW redundancy, to ensure the network functionality even in cases where one or more nodes fail. The network also presents security features that ensure various security related aspects avoiding external attacks and protecting data through encryption.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PARTICIPANTS

NO	LEGAL NAME	SHORT NAME
2	Sirti Società per Azioni	SIRTI
3	Ardanuy Ingenieria SA	ARD
6	Istituto Superiore Mario Boella Sulle Tecnologie Dell’Informazione e delle Telecomunicazioni Associazione	ISMB
7	Perpetuum Limited	PER
8	University of Newcastle upon Tyne	UNEW

1. INTRODUCTION

This document covers:

- The definition of TI (Train Integrity)
- Requirements for checking TI
- Definition of a WSN (Wireless Sensor Network) protocol for TI assessment
- Report of the employed hardware for implementing the TI assessment protocol.
- Power analysis and interfacing with an EH (Energy Harvesting) solution
- Antenna design for the WSN node
    - Antenna for communication
    - Antenna for measuring distance
- Simulations of the network

1.1 List of Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard (ISO/IEC 18033-3)
CID	Coupling ID within the train
CLD	Coupling Limit Distance
CM	Control Module device located on train engine
COM	Communication
CRC	Cyclic Redundancy Check
DS	Distance Sensor
EH	Energy Harvesting
EHS	Energy Harvesting & Storage
EU	European Union
ISR	Interrupt Service Routine
NSID	Sensor node Short ID
NTID	Sensor node ID in the train (CID concatenated with NSID)
NUID	Sensor node Unique ID
OTI	On-board Train Integrity
REST	Representational State Transfer
RF	Radio Frequency
RSSI	Received Signal Strength Index/Indicator
RTLS	Real Time Localization System
RX	Reception
SN	Sensor Node for train integrity check
STM	STMicroelectronics

TDMA	Time Division Multiplexing Access
TI	Train Integrity
TRL	Technology Readiness Level
TWR	Two-Way-Ranging
TX	Transmission
UWB	Ultra-Wide Band
WSN	Wireless Sensor Network

Table 1 - List of Acronyms

1.2 Relevant Requirements

The system requirements that the communication system and the distance measuring system should respect have been previously analysed in T2.4, proving the actual requirements in D2.2.

In the following Section the requirements for the communication and TI check systems are reported.

1.3 List of Requirements

1.3.1 Distance Sensor

1. SR_DS_1: The DS module shall be capable of measure a distance up to 15 meters (with a maximum error +/- 5 meters) to ensure that integrity is actually lost and not that the module is unable of measuring the distance.
2. SR_DS_2: The DS module must present one or more interfaces for exposing the APIs for the Microcontroller unit.
3. SR_DS_3: The DS module shall be able of performing a measurement under 1 second, in order to not introduce a delay that will confirm integrity lost due to timeout.

1.3.2 Microcontroller

1. SR_MU_1: The Microcontroller unit must present one or more interfaces suitable for communicating with the other modules.
2. SR_MU_2: The Microcontroller unit shall be capable of managing an ad-hoc communication protocol stack for a WSN
3. SR_MU_3: The Microcontroller unit shall be compliant to very low power consumption computing and capable of being powered by an EH solution.

1.3.3 Communication Module

1. SR_COM_1: The Communication module must present one or more interfaces suitable for communicating with the Microcontroller unit.
2. SR_COM_2: The Communication module must be compliant to the Sub-GHz 868 MHz communication standards of frequency, band and power.
3. SR_COM_3: The Communication module shall be capable of enhanced features for power consumption and communication reliability adapted to a WSN.

2. DESIGN AND IMPLEMENTATION OF THE COMMUNICATION SOLUTION

In the following chapter the communication solution is outlined. The solution is based on a WSN that employs a sub-GHz communication radio band; since the solution aims to be deployed within the EU (European Union), the used sub-GHz ISM (Industrial Scientific and Medical radio bands) band is the 868 MHz, recalling to the standards defined by the ETSI (European Telecommunications Standards Institute) organization.

The design comprehends two main components that will compose the network:

1. The SN (Sensor Node)

2. The CM (Control Module)

[1] The application scenario foresees the installation of four SN modules for each wagon, independently of the role of the latter (wagon, passive locomotive or active locomotive). In addition to the installation of the SN nodes, it is also expected the installation of a CM component on the leading locomotive. An overview of the described design is depicted in Figure 1.

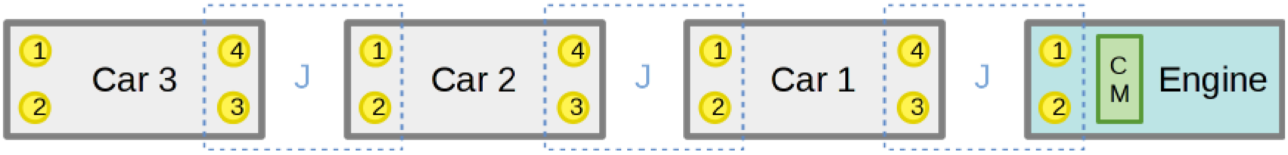


Figure 1: Train elements for integrity assessment

2.1 DEFINITION OF THE TRAIN INTEGRITY SYSTEM

The TI System is based on:

- A period collection of the TI data collected by the SN devices mounted on the wagons;
- Message forwarding from the various SN devices towards the CM mounted on the leading locomotive;
- Handling abnormal cases that may affect reliability of the TI assessment.

Furthermore, the TI System must also respect the following aspects:

- The TI System must be efficient and reliable enough for integrity assessment and for the capability of functioning with an EH solution;
- Another aspect, that the TI System must comply to, is the security of the network. The network must be resilient to leaking data and external tampering;
- The SN devices deployed on a convoy must be capable of autonomously generated and initialise the network that will then be the backbone of the TI System;

- A basic interface must be provided on the CM so that the TI can be monitored. The TI result shall be displayed on the CM display and potential alarms can be set and sent to the main on-board computer.

TI is assessed by means of checking that the distance between two facing SN devices across each coupling is less than a pre-set threshold. A measured distance above the admissible threshold is decoded as a “broken coupling” and signalled to the CM that will output a TI not confirmed or TI lost signal to the main on-board computer and display the status to the human operator(s) through the interface.

2.1.1 Sensor Node Architecture

The employed WSN-based solution employs devices denoted “Sensor Nodes”. All SN are identical to each other and is a small system based on the following components:

- Micro-controller module;
- Radio communication module;
- Distance measurement module.

The detailed hardware overview is outlined in Section 2.4, in Figure 2 it is possible to overview the high-level architecture of the Sensor Node.

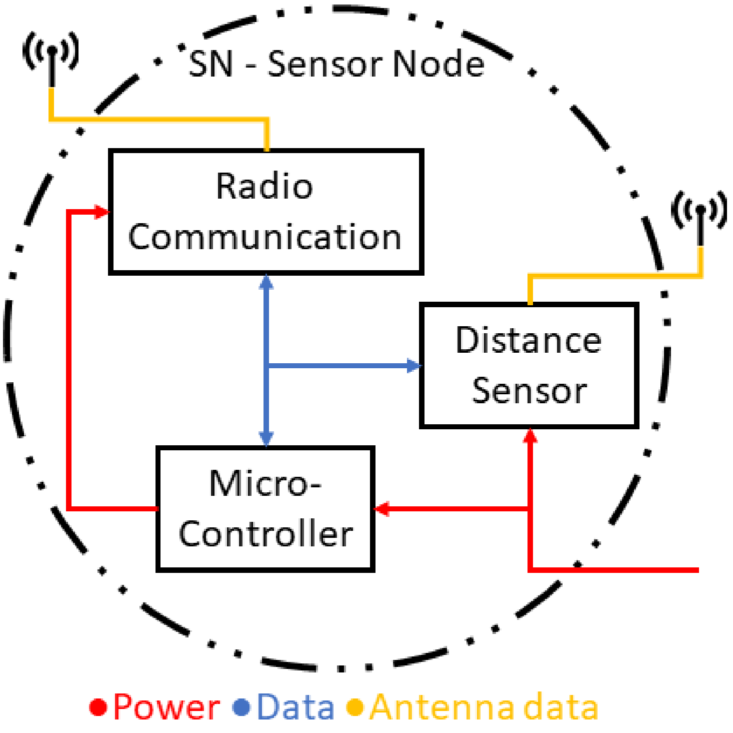


Figure 2: Sensor Node high-level architecture

2.1.2 Development Scenario

For the TI assessment, the scenario shown in Figure 1 is considered, specifically:

- One leading locomotive at one end of the train (labelled “Engine” in Figure 1);
- One CM unit located on the leading locomotive (labelled “CM” in Figure 1);
- The convoy can be composed by up to 50 wagons (labelled “Car <i>” in Figure 1);
- Each wagon (locomotive or not) mounts four SN modules (labelled “1” to “4” in Figure 1, only two sensors displayed for the “Engine”, but four mounted);
- For a N cars convoy, N-1 couplings are considered to check the TI status (labelled “J” in Figure 1).

By leading locomotive, it is intended the locomotive that is located on one of the two ends of the train and that contributes to the train’s motion. Also, by wagon it is intended any generic wagon comprehensive of locomotives that do not cover the role of leading locomotive, thus meaning a passive locomotive or active but not located on one of the two ends of the train.

In order to be capable of assessing TI, each SN must be capable of:

- Measuring the distance between themselves and their facing node across the coupling (referencing to Figure 1, node “1” must be able of measuring the distance with node “4” and vice versa; same concept applies to nodes “2” and “3”);
- Comparing the measured distance with a reference value (Coupling Limit Distance (CLD)), if the measurement is greater than the CLD, then coupling integrity is considered not confirmed.
- Communicating through a wireless channel with the other nodes that belong to the same wagon or to other wagons.

Each SN will be associated an identifier denoted with the term NUID (Sensor Node Unique ID).

2.2 Network Formation

In this section the following procedures are outlined:

1. Discovery of all the SN that belong to the train;
2. Discovery of all the couplings between wagons and the associated SN devices;
3. Assignment of the network specific short IDs to all the SN and the couplings;
4. Configuration of the SN for proper operation in the network;
5. Network re-join after reset or power cycle events of the nodes

2.2.1 Discovery of the SN and Train Couplings

The process of discovering all the SN and the train couplings will be shortly denominated “Network Discovery”. This process is initiated by the CM located on the leading locomotive, typically under explicit human request.

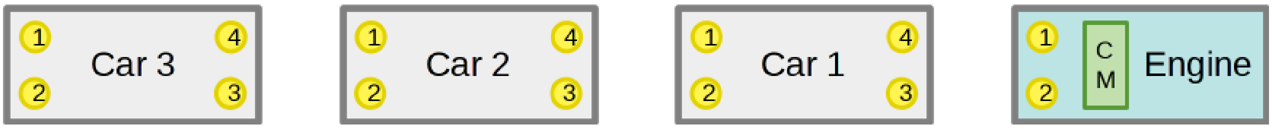


Figure 3: Train physical composition sensor node discovery

Assuming a physical train composition as the one shown in Figure 3, the purpose of the network discovery is to collect all the SN IDs that belong to the physical train and to avoid the accidental inclusion of sensor nodes that belong to nearby trains. The network discovery also includes the phases of:

- Node renumbering from NUIDs to NSIDs (Sensor Node Short ID);
- Discovery and numbering of all the train’s couplings, assigning CIDs (Coupling ID) to each of them;
- Composition of the NTID (Sensor Node Train ID) for each node, this is generated by the concatenation of the CID and NSID.

The network discovery procedure assumes that:

- Each node can detect and collect the unique ID of the facing node across a coupling (e.g. node “1” with node “4” as can be seen in Figure 3) and reject, with sufficient selectivity, network join replies from nearby nodes (e.g. reject join replies from node “3” as can be seen in Figure 3 or join request incoming from a nearby train);
- Sensor nodes on the same train element (wagon or locomotive) know the unique IDs of the other nodes on the same train element (wagon or locomotive). The CM on the locomotive knows the IDs of all the SN mounted on the locomotives itself.

A full network discovery is achieved by iteratively repeating a series of steps until the full train is covered.

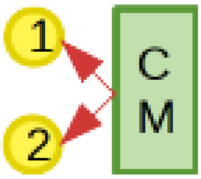


Figure 4: Initiation of network formation request by the Control Module (CM)

During the first step of the network discovery procedure, the CM sends a “start discovery” request to the NUIDs of nodes “1” and “2” located on the train’s locomotive (as displayed in Figure 3 and Figure 4). The NUIDs of the SNs of the locomotive are known by the CM and do not need to be discovered. Moreover, the CM will:

- Assign beforehand a NSID (short ID) for each unique NUID that it addresses requests to (see IDs “1” and “2” shown in Figure 4) and the number of the coupling;
- Communicate the NUID -> NSID assignment and the coupling number (CID) to the nodes together with the request for peer discovery across their coupling.

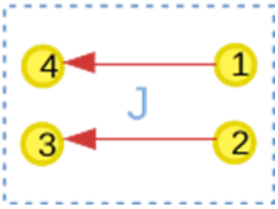


Figure 5: Direct peer discovery across coupling using the on-board distance sensors (DS)

In the next phase of the network discovery procedure, the two nodes that received the peer discovery request from the CM will start the discovery of their peers across the first train coupling. In order to correctly discover the wanted peers, the procedure uses the on-board distance sensor (DS, see Figure 5 and Figure 3). The DS sensor is directive and the sensor nodes implement distance thresholds to prevent the accidental discovery of other sensor nodes either across the same coupling (e.g. node “1” discovers node “3”), or SN that belong to nearby trains.

The DS sensor allows also to collect the peer NUID at the same time with the distance measurement. Subsequently, each of the nodes “1” and “2” will assign the NSIDs to their peers, as shown in Figure 5, and communicate both the coupling ID (CID) and these new NSIDs using direct messages towards their peers (addressed to the NUIDs of nodes “4” and “3”, respectively) through the WSN communication channel.

At the same time, the association established between the NUIDs and the coupling and junction-specific renumbered NSIDs is forwarded back to the CM over the WSN segment that has already been discovered.

The distance measurement is checked against a configurable maximum value (for example the length of a wagon), if the measurement is below this means that the distance could correspond to coupled wagons and the distance id used to generate the reference value for the integrity check for the coupling. The reference value is the CLD and represents a distance that if measured means that the vehicles can no longer be coupled, it is the sum of the initial measured value, the maximum possible change in the coupling length and an extra factor to take into account potential errors and prevent false indications. The CLD should not allow the vehicles to be separated by more than 2 meters before coupling integrity (and therefore TI) is no longer confirmed.

In the last step, sensor nodes “3” and “4” (see Figure 5) pass the network discovery request to the sensor nodes at the other end of the newly discovered train car, “Car 1” in Figure 3. As the CM on the locomotive, both sensor nodes “4” and “3” of “Car 1” know the NUIDs of nodes “1” and “2” of the same wagon. Hence, they use their NUIDs to address nodes “1” and “2” directly, as shown in Figure 6.

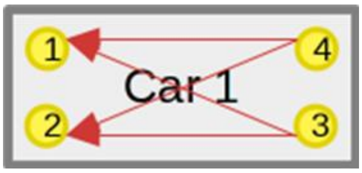


Figure 6: Propagation of network discovery request from one and to the other of a train car

For redundancy, both nodes “3” and “4” send discovery requests towards both of the nodes on the other end of the wagon. Specifically:

- node “4” sends one direct discovery request to node “1” and one direct to node “2”
- node “3” sends one direct discovery request to node “1” and one direct to node “2”
- each of the above requests use the destination node NUID and its renumbering to NSID as follows:
    - the node on the side of node “4” receives NSID “1”
    - the node on the side of node “3” receives NSID “2”

The new associations established between the NUIDs of nodes “1” and “2” and the new coupling- and junction-specific renumbered NSIDs is forwarded back to the CM over the WSN segment that was already discovered.

Once nodes “1” and “2” of “Car 1” receive the direct network discovery request(s) from nodes “3” and “4” of the same wagon, they start with the first step of the network discovery exactly how nodes “1” and “2” of the locomotive did upon receiving the same command from the “CM” (see Figure 4).

This process is repeated for all wagons until it covers all train couplings.

At the end of the train, the SN “1” and “2”, that belong to the final wagon on the far end of the train, do not face any train coupling. Hence, they will not discover peer nodes, they will redundantly report this back to SN “3” and “4” of the same wagon, as shown in Figure 7.

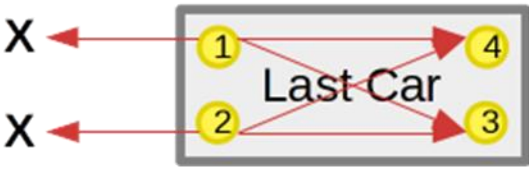


Figure 7: Last nodes of last car detect the end of train

Consequently, SN “1” and “2” of the last wagon, which do not oversee a train coupling (see Figure 7), will not be included in the newly discovered train integrity WSN.

2.2.2 Security Establishment

The information exchanged over the wireless network is protected using encryption.

The encryption mechanism is divided in two phases. The first phase has the task of distributing the symmetric key with asymmetric encryption; the second phase is the actual data encryption based on the symmetric key. The process is the following:

- Initial exchange of public keys between sensor nodes and the CM;
- Generation of a secret symmetric encryption key by the CM;
- Secure, point-to-point CM-to-node distribution of the secret symmetric key, encrypted using the public key of each target node, meaning that the asymmetric encryption is employed;
- Decryption of the secret symmetric key on each node using the node’s private key, which is the counterpart of the public key that was used by the CM to encrypt the secret symmetric key for that specific node;
- Use of the symmetric key for all subsequent communications

In the following we assume that:

- Each sensor node has a unique pair of public and private keys;
- The CM has its own pair of public and private keys;
- The CM can create a random and secret symmetric key, unique for the network;
- All nodes and the CM can use the symmetric key to encrypt and decrypt network messages.

Initial public key exchange

The initial exchange of the public keys between the SNs and the CM is done during the network discovery process described in Section 2.2, as follows:

1. The public key of the CM is initially distributed to nodes “1” and “2” located on the train’s locomotive embedded in the initial “start discovery” request packets sent by the CM to those nodes;

2. Loop for each wagon to exchange public keys:

   a) The public keys of the first two nodes “1” and “2” (located on the train engine) are sent to the CM in the packets of the network discovery procedure that propagate back to the CM the association NUID => NSID for each of the nodes “1” and “2”;

   b) As nodes “1” and “2” discover across the coupling their peers “4” and “3”, respectively, the former pair of nodes (SN “1” and “2”) transfers the CM public key to the latter pair (SN “3” and “4”) included in the network discovery packets that are described in Section 2.2;

   c) Conversely, the public keys of nodes “3” and “4” are sent to the CM in the packets of the network discovery procedure that communicate the association NUID => NSID for each of them;

   d) Further on, as nodes “3” and “4” send network discovery packets to nodes “1” and “2” towards the other end of the first wagon, the former pair of nodes will include the CM public key in the same packets so that the latter pair of nodes can store it (as in step 1 of this procedure, listed above);

3. Then steps 1.a) to 1.d) are repeated until the network of the full train is discovered and all public keys are propagated.

Symmetric key generation and distribution

At the end of the network discovery procedure described in Section 2.2.1 and of the public key exchanges previously described, the CM will have the public keys of all the network’s SNs, each sensor node public key associated to the node unique IDs (NUID). At the same time, each SN in the discovered network will have a copy of the public key of the CM.

The public keys allow the CM and the SNs to exchange encrypted information point-to-point.

But using asymmetric encryption using public-private keys is memory and computational expensive, and does not allow broadcasts, which are essential key points for the WSN protocol described in Section 2.3.

Hence, the CM will generate a random secret symmetric key, which allows to achieve the same level of security using less resources and also supports broadcast messages.

Once created, the new secret [1]symmetric key is then securely distributed to all sensor nodes in the network as follows:

a) The CM encrypts the symmetric key with the public key of SN “1” on the locomotive.

b) The CM sends the encrypted symmetric key to the target SN, “1” on the locomotive.

c) The target SN decrypts the symmetric key using its private key that matches the public key used by the CM to encrypt the symmetric key before sending it over the unsecure wireless channel.

d) The target SN, “1” on the locomotive, creates a node-specific and symmetric key-specific and non-zero nonce as follows:

$$(Symmetric\ key) \oplus ((CID \ll 2 \mid NSID) + 1)$$

e) The target sensor node, “1” on the locomotive, uses the symmetric key to encrypt the nonce and sends it to the CM.

f) The CM decrypts and checks the proper value of the nonce, since the CM knows the symmetric key, as well as the CID and NSID of the target node, which were used to create the nonce on the target node (“1” on the locomotive).

g) If the nonce does not check on the CM or if the CM does not receive the reply with the nonce from the node within a predefined time, it can repeat the symmetric key exchange procedure towards the target node, restarting from point a) above.

h) If the nonce check succeeds, the CM knows that the encrypted communication works fine with the node.

The CM selects the next target node and resumes the distribution procedure of the secret symmetric key, restarting from point a) above.

Use of the encrypted communication

At the end of the distribution procedure for the secret symmetric key (described in the previous Section), all sensor nodes and the CM can use the symmetric key to both encrypt and decrypt messages.

Importantly, the symmetric key allows each and any of the network participants (all sensor nodes and the CM to decrypt messages encrypted by any other network participant. At the same time, the encrypted contents of any message cannot be decrypted by nodes that do not belong to the network, even if they overhear the wireless communications of this network.

The use of encryption protects also from accidental or intentional injection of information in the network. In fact, any injected information cannot be encrypted with the encryption key used in this

[1] With the symbol ⊕ it is intended an exclusive OR operation

network (which is secretly generated during network formation and is never transmitted in clear over the air). Hence, entities outside of the network cannot properly encrypt information that can be decrypted with this network secret key, hence the network is cryptographically protected from outside injection of wrong or evil data and commands.

To achieve this level of privacy and protection, all sensor nodes and the CM of the train network should use the symmetric encryption key as soon as it is generated and distributed. Before that, network communication is protected using point-to-point communications encrypted using the public keys of the peers.

2.3 Train Integrity Assessment and Notification

Train integrity assessment is the main function of the WSN.

Train integrity is assessed by periodically measuring the distance between directly coupled train cars and checking that it is within the normal operation range, meaning that, the system checks that the measured distance does not exceed the CLD.

Train integrity is considered confirmed when the distance between any of the directly coupled train cars (including the distance between the leading locomotive and the first train car) is equal or below a predefined threshold (CLD); since there are multiple measurements, if a single measurement respects these aspects, then TI is considered confirmed. If all measurements of the corresponding coupling are above the CLD value or the measurements are invalid, then TI is considered compromised. This is an emergency case which should be notified to the CM on the leading locomotive as soon as possible. Also, if the CM does not receive confirmation of all the couplings integrity status (from at least one of the two pairs of nodes measuring the coupling integrity) the status of TI should be considered compromised.

Additionally, the train integrity procedure should prevent false positives, i.e. the procedure should not actively confirm that all train integrity conditions are satisfied (all coupling distances are below the threshold) when one or more distances are above threshold (hence train integrity is compromised) or no positive confirmation of coupling integrity has been received from at least one pair of nodes measuring the coupling. In these cases, a redundancy check must be performed, TI can be confirmed if the nodes that previously computed a measurement below the CLD confirm the measurement in the redundancy check, also confirming the neighbouring nodes are malfunctioning.

In case of failure of that leads to impossibility to check all the integrity conditions, an alarm will be produced and received by CM in less than 5 seconds.

2.3.1 Design Principles and Operation Overview

The design of the train integrity assessment and notification system satisfies several important objectives:

- Minimize positive errors while assessing train integrity, i.e. avoid asserting that train integrity is satisfied when this is not true, even in the case of TI failure;
- Perform a full assessment of train integrity within the minimum allotted time of 5 s;
- Minimize energy requirements to stay within the energy envelope provided by the EH power source device;
- Operate properly even with intermittent and unpredictable power availability, which can happen when the train is traveling at low speeds or is stationary, or when the train starts moving after a long stationary period;
- Ensure that nodes quickly discover and become operative in the current state of the network after their power-up (use few node states and quickly infer and align node state to network operation based on current network traffic);
- Minimize the WSN protocol requirements to join the network (i.e., avoid synchronous communications which require continuous and precise network-wide time synchronization).

In the following it will be described the operation of the SNs and of the WSN protocol that satisfies these objectives.

2.3.2 Assessment of the Integrity of One Coupling

Considering the train elements in Figure 1, the integrity of the train is check sequentially, one coupling at the time, as shown in Figure 8.

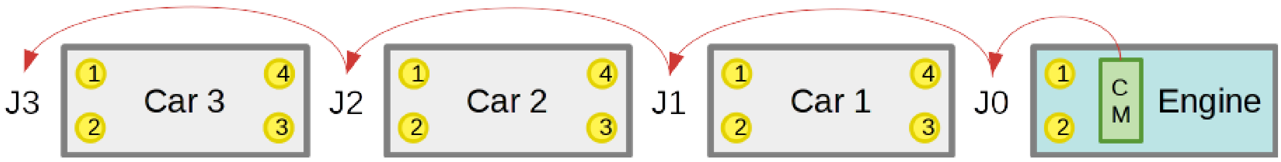


Figure 8: Sequential check of train integrity

Train integrity assessment is started by the CM or upon human direct request.

The integrity assessment starts with a request sent by the CM over the WSN to “start integrity assessment for coupling 0”. This is received by the sensor nodes assigned to coupling 0 during network formation (see Section 2.2.1). The sensor nodes of coupling 0 will start right away the integrity assessment for the coupling. The assessment includes several operations and WSN communications that are limited to the sensor nodes that belong to coupling 0, as follows:

a) Use the DS (Distance Sensor) to measure the distance between each pair of facing nodes twice, for redundancy:

- Distance between node 4 and 1;
- Distance between node 1 and 4;
- Distance between node 2 and 3;
- Distance between node 3 and 2.

b) Each measured distance is compared with the predefined integrity threshold (CLD) to assess and encode the integrity of the coupling or possible error states;

c) Each node communicates its assessment of the integrity of the coupling to the other sensor nodes that belong to the same coupling;

d) Each of the other nodes acquire the assessments sent by their peers and acknowledges it by reporting them back. This way each assessment is repeated four times in the communications of the sensor nodes that belong to the coupling, which ensures a good level of redundancy for the WSN network communications;

e) Once all nodes have completed their measurements and communicated them to their peers, all sensor nodes belonging to coupling “0” will forward the results to the nodes that belong to the next coupling, e.g. “1”, together with the request to start its integrity assessment, and the procedure starts for coupling “1” from point a) above.

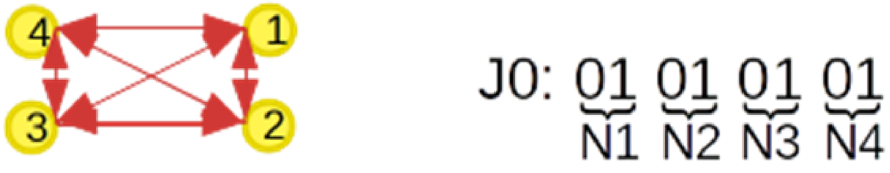


Figure 9: Coupling status byte filled integrity assessment encodings

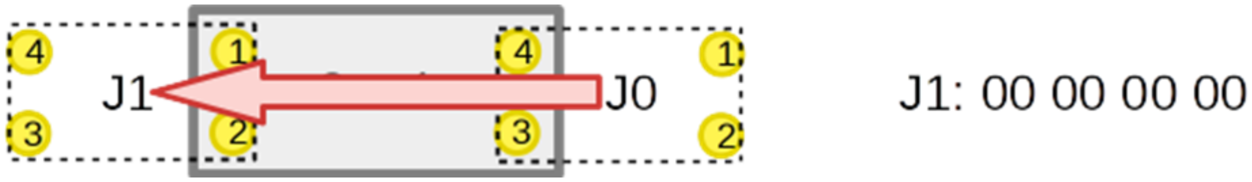


Figure 10: Transfer of integrity assessment request to next coupling

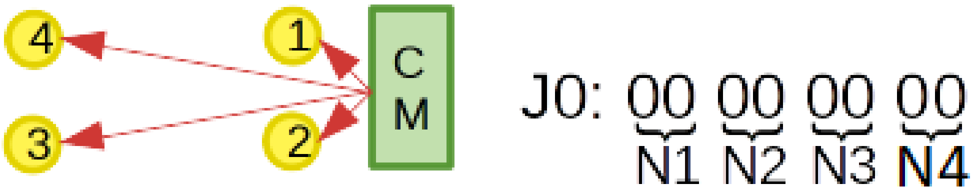


Figure 11: Empty coupling status byte before integrity assessment

Each sensor node encodes the state of its coupling on two bits, hence the assessments of all four sensor nodes belonging to a coupling are encoded in a byte.

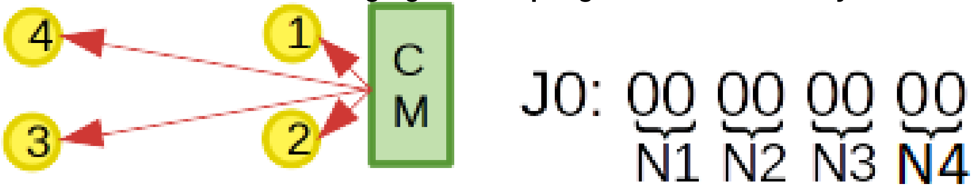


Figure 11 shows the value of the byte that will receive the encodings of the coupling integrity of the four sensor nodes at the begin of the assessment, while Figure 9 shows an example of the coupling integrity encodings collected by the four nodes.

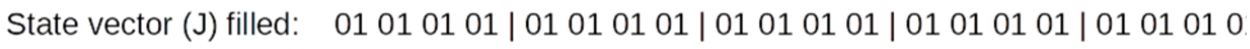
Once the integrity assessment request is forwarded to the nodes belonging to the next junction, 1, those sensor nodes will fill their assessments in a new byte, as shown in Figure 10.

Thus, the train integrity assessment procedure consists of two main operations:

1. Assessment and record of integrity of one coupling;
2. Forward propagation (from train engine to the end of train) of recorded integrity status to the nodes of the next coupling.

2.3.3 Collection of Full Train Integrity Data

The bytes holding the assessments of all train coupling are organized in a vector, where each coupling status byte is indexed by the coupling number. At the end, the vector holds the encodings of all the couplings of the train and can be used to determine the train integrity, as shown in Figure 12.



State vector (J) filled: 01 01 01 01 | 01 01 01 01 | 01 01 01 01 | 01 01 01 01 | 01 01 01 0

Figure 12: Train integrity state vector

2.3.4 Integrity Data Propagation to the Control Module

Once the assessment of the integrity of all couplings and the forward propagation of the integrity encodings reaches the end of the train, the train integrity state vector (see Figure 12) is sent back to the CM on the locomotive, along the same WSN network as follows:

a) Each SN belonging to the last train coupling will send the TI state vector back to the nodes belonging to the previous coupling, ensuring a good redundancy;

b) Once the sensor nodes of the previous coupling receive the train integrity state vector, they will forward it to the coupling before them, towards the train engine;

c) Point b) above is repeated until the train integrity state vector reaches the CM on the locomotive;

d) The CM decodes the TI state vector and displays the result.

2.3.5 Node States and Processing Flow

As mentioned in the design objectives, the operation of the SN has a few states and makes extensive use of REST operation while limiting to minimum sequential operations.

The state and processing diagram is shown in Appendix 6.

Most of the time, the sensor node stays in the Idle state. It can leave this state if one of the following events are detected:

- A WSN packet is received which is addressed to the coupling to which this node belongs, or to the next or previous coupling;
- The result of the distance measurement across the coupling is ready;
- A transmission (TX) was requested;

- A timeout ensued.

Following it is presented in detail the SN operation in each of the listed cases.

Packet Addressed to the Coupling of the Node

The sources of packets addressed to the coupling of the node can be:

- A node from the previous coupling (closer to the CM than the coupling of this node);
- Another node belonging to the same coupling;
- A node from the next coupling (farther away from the CM than the coupling of this node).

Packets received from a SN from the previous coupling (closer to the CM than the coupling of this node, we will call them FWD) contain requests to start the integrity assessment together with the results of the assessment of all previous couplings, from the CM down to the coupling of this node. Upon receiving this request, the node performs several initializations:

- Sets a timer for the maximum time allowed to assess the integrity of the coupling;
- Sets a timer for the maximum time allowed to receive the distance measurement from the DS;
- Sends the distance measurement request to the DS;
- Stores the request ID to filter out redundant requests;
- Stores the train integrity state vector received with this request;
- Clears and sets various internal flags.

Packets received from SNs that belong to the same coupling (we will call them INJ) carry measurement data stored in the coupling status byte. These data can be either:

- produced by the SNs themselves, from the measurement of the coupling distance and the subsequent assessment of the integrity of the coupling;
- replica of data received from other SNs belonging to the same coupling, which are acknowledging the reception of other integrity assessments of the coupling.

Either way, for the SNs that may have not received any of the FWD packets (hence nodes that are not aware that a measurement cycle started for their coupling), the INJ packets act also as a start of the coupling integrity assessment. I.e., besides sharing other SNs assessments, the INJ packets have also the function of FWD packets, when the latter were not received.

At the end of the processing of the integrity assessment of the coupling, the SNs belonging to the coupling send FWD requests to the SNs belonging to the next coupling.

This assessment procedure is repeated for all train couplings. Hence, the integrity of the couplings is sequentially assessed one at a time. The integrity assessment results for each coupling are stored in the coupling status byte (see Figure 9) and all the coupling status bytes are collected in the train integrity state vector (see Figure 12).

Packets received from a SN from the next coupling (further away from the CM than the coupling of this node, we will call them BWD) contain requests to forward back to the CM the TI vector, which now has collected the integrity assessments for all couplings. This packet is generated at the end of the train, after the SNs assess the integrity of the last train coupling.

SNs that receive BWD packets just forward their contents (basically the integrity state vector) to the previous coupling (i.e. the coupling closer to the CM on the leading locomotive).

It should be noted that while a SN is busy processing an integrity assessment request it will ignore other assessment requests. A SN starts processing a request when it receives either an FWD or an INJ packet, and finishes processing the integrity assessment request when it has sent all FWD requests directed to the SNs belonging to the next coupling.

Completed Measurement Results

The measurement of the distance of the coupling is requested by each SN to the DS sensor that it is attached to when the node receives the first FWD or INJ packet for a new integrity assessment request.

While the DS performs the measurement, the node returns to the idle state to conserve energy and to be ready to process other internal (e.g., timeouts) or external (e.g., RX packets) events.

When the DS finishes the measurement, it will notify the node, which will start processing it as soon as possible. The SN will:

- Set the “measurement done” flag to prevent re-requesting a new measurement that would be triggered by a subsequent RX packet;
- Assess coupling integrity by comparing the distance measured with the predefined threshold (CLD);
- Store the coupling integrity assessment in the position allocated to the SN in the coupling status byte (see Figure 9);

- Request sending a TX packet with the new measurement;
- Set an acknowledge timeout timer and an acknowledge counter.

Timeouts

The timeouts set by the SN are for:

- DS measurement ready (DS_TO);
- Acknowledge of integrity assessment data produced by the SN itself or repetitions of FWD and BWD packets (ACK_TO);
- Full assessment of coupling integrity (J_TO, collect integrity assessment from all sensor nodes assigned to the coupling, see Figure 9);
- BWD packet not received (BWD_TO).

If the DS measurement is not ready by the expiration of DS_TO (or if the DS reports back an error), the SN will fill a code for “error” as its assessment of junction integrity. Then it will proceed with the propagation of this assessment the same way as it would do for successful assessments of junction integrity.

The acknowledge timeout (ACK_TO) and the acknowledge counter are used to ensure that an integrity assessment is reported at least four times with INJ packets, as follows:

- First time it is send by the node that performed the assessment;
- The other three times come from the acknowledgment sent by each of the other three SN belonging to the coupling.

Nevertheless, the protocol considers the SN that produced the integrity assessment “responsible” for repeating the measurement notification four times, even if the other nodes do not acknowledge it for some reason (e.g., due to node faults or RF interference).

Hence, each time the originating SN receives an acknowledge from one of the peer nodes, it will reset its acknowledge timer and decrease the acknowledge counter.

If the ACK_TO timer expires, this means that an acknowledge from one of the other nodes was not sent. Consequently, the node will repeat the TX with the measurement to compensate for the lack of external acknowledge and maintain a proper redundancy level for the WSN protocol.

The J_TO will typically expire when one or more of the SNs of the coupling do not operate properly (e.g., they do not have enough energy). Upon J_TO expiration, the SN will fill with error code the missing assessments from the nodes that did not respond, then it will proceed to send FWD packets.

The TX of FWD packets is handled similarly to other data produced by the SN, such as integrity assessment data. The SN will set ACK_TO and counter and will behave exactly as when sending integrity assessment data described above.

BWD_TO is set after the coupling processing finished. It means that the SN expects to be asked to forward a backward packet towards the CM related to the current request ID before BWD_TO expires.

If BWD_TO expires, the SN will:

- Fill errors in its copy of train integrity state vector (see Figure 11) for the couplings beyond its coupling;
- Send the vector towards the CM using a BWD packet.

This behaviour is meant to cover the cases when all nodes in the next train coupling are not working properly, e.g., because of node or power failures.

Transmission Requested

A SN requests transmission either upon internal or external events.

Internal events that lead to transmissions are:

- New coupling integrity assessment ready;
- Expiration of some timeout.

Before issuing a TX request, the node will:

- Set up the data that will be included in the packet;
- Set the packet type;
- Start the TX handler.

The TX handler will assemble the packet contents using the available data, based on the packet type, then will attempt to send the packet according to WSN MAC.

2.3.6 WSN MAC

The WSN MAC is kept simple to facilitate node re-joining the network and not rely on inter-node or network-wide precise time coordination, which can be difficult to achieve in the wide temperature range expected during the operation of the train or with asynchronous power failures and restores that follow the energy harvesting performance of individual nodes.

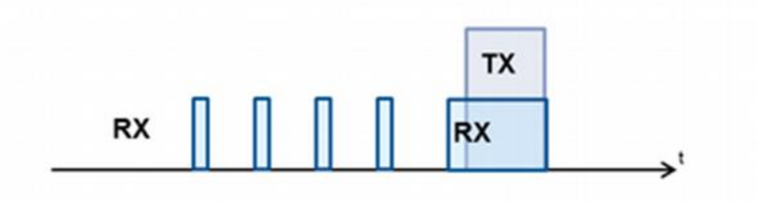


Figure 13: Asynchronous WSN MAC using RX sniff operation

The integrity assessment WSN uses a typical asynchronous MAC. This requires the nodes to be always ready to receive a packet, hence allows the nodes to TX at any time they do not find a busy communication channel (listen before talking).

To reduce the RX energy consumption, the radios of the SNs will sample periodically the communication channel looking for RF activity. If RF intensity in the channel (RSSI) is above a predefined threshold, the RX continues attempting to receive a full packet, as shown in Figure 13.

To improve the energy efficiency, the packets should be preceded by a preamble, which carries no data and whose length determines the maximum spacing between two RX sniff samplings (hence the RX energy).

The MAC does not provide retransmissions in case of conflicts or if the radio space is busy.

2.3.7 Network Re-Join After Node Reset (Power Cycle)

The asynchronous MAC allows any node that was previously configured as described in Section 2.2 to re-join the network with no overhead, just by simply starting to sniff the channel as shown in Figure 13.

2.3.8 WSN Optimizations

One important WSN objective is to transfer the TI assessment request from the CM, on the leading locomotive, and the assessment results for all train couplings back to the CM in the allotted time, of 5 seconds.

Another WSN objective is to reduce the WSN-related energy consumption of each SN of the network.

With these broad objectives in mind, in the following it will be discussed the several WSN optimization options.

Packet preamble length

The STMicroelectronics S2-LP radio chip allows much flexibility in the composition of packets. Table 2 shows the elements of the basic packet.

Table 2: Elements of the basic S2-LP radio packets

Preamble	Sync	Length	Address	Payload	CRC	Postamble
0:2046 bits	0:32 bits	0:2 bytes	0:1 bytes	0:65535 bytes	0:4 bytes	0:510 bits

The WSN packet preamble is used to allow the PLL of the radio RX device to synchronize with the clock of the radio TX device that sends a packet to properly receive the packet symbols. Hence, the packet preamble needs to have a minimum length to ensure enough time for the receiver PLL to stably synchronize.

As show in Figure 13 and Figure 14, in sniff mode the radio device samples periodically the communication channel looking for radio activity. If there is no RF signal, the receiver returns to sleep. Otherwise, as shown in Figure 14, the receiver will stay in RX mode and will attempt to receive a full packet.

To avoid losing packets, the sampling period in sniff mode should be less than the preamble length minus the minimum preamble length. This ensures that when the receiver will “sniff” the TX signal in the communication channel there will be at least the minimum preamble length left to be sent by the transmitter, which is important because this ensures the proper PLL synchronization and proper packet reception.

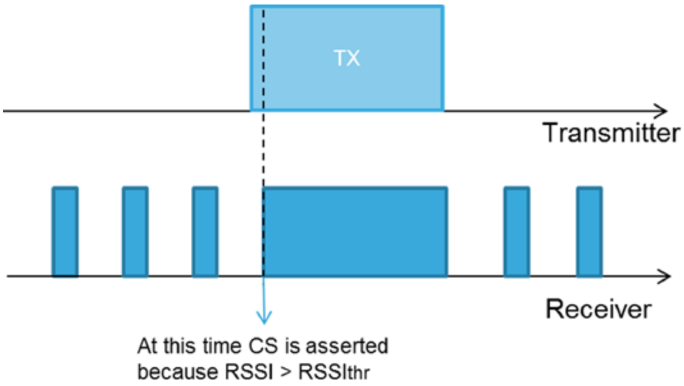


Figure 14: Use of packet preamble in the sniff-mode RX

The S2-LP radio allows to extend the length of the preamble beyond the minimum length. This allows the receivers that operate in sniff mode (See Section 2.3.6) to reduce the sniff frequency, hence their energy consumption in RX listening mode.

At the same time, a longer packet preamble means a longer overall packet duration. This increases both the energy consumption of the transmitter and the occupation of the radio communication channel.

With these considerations in mind, the best length of the packet preamble will be explored that on the one hand minimizes the energy consumption of the nodes, and on the other hand does not increase too much the occupation of the communication channel.

Packet Size

As shown in Table 2, the payload can significantly increase the packet length besides the preamble.

We can consider the reduction of the packet payload on the forward path, see Section 2.3.2 and Section 2.3.3.

The current implementation of the WSN protocol uses the same length packets for both forward and backward (see Section 0) paths, because fixed length packet communications are generally less susceptible to receive ill-formed packets.

However, we can see that on the forward path there is no need to propagate the train integrity state vector. The SNs can store the integrity assessment results and fill them in the integrity state vector during the backpropagation.

This way we can reduce the payload of most packets by around 50 bytes (the length of the integrity state vector), specifically for 18 out of the typically 26 packets exchanged by a node during one integrity assessment cycle (see Section 2.4).

Hence, packet size is another important aspect to optimize during simulations, together with packet preamble.

In fact, it is possible to approximately calculate the maximum duration of a packet. It is known that the maximum duration of a TI assessment is 5 s and in the worst case there are 50 couplings to assess. This leaves a maximum of 100 ms per coupling, which includes integrity assessment and all related communications on the forward and backward path. Considering that the WSN protocol typically needs 26 packet exchanges per coupling, it is possible to calculate the maximum duration of a packet to 100 ms / 26 packets = 3.8 ms.

This duration can be used as baseline to calculate the maximum packet length (which includes the preamble, payload, sync, CRC, etc.). For the maximum RF bit rate of 250 kbit/s, in 3.8 ms we can transfer at most 950 bits, or about 118 bytes.

These results and simulation results will be used to optimize channel allocation between the forward and backward path, as well as the size of the preamble vs packet payload.

Redundancy Level

The WSN protocol includes communication redundancy to ensure adequate data and command propagation in case of variable and adverse propagation conditions, or external RF interference.

Redundancy is embedded in the WSN protocol by design, and includes reception acknowledges and retransmissions, as well as repeated transmissions (see Sections 2.3.2, 2.3.3, 0).

The necessary amount of redundancy will be determined using extensive WSN protocol simulations, both with and without injection of errors. Lower redundancy reduces the occupation of the communication channel, which may be used to improve the trade-off between TX and RX energy by varying the size of the packet preambles.

2.4 HARDWARE COMPONENTS

In order to comply with all the required functionalities, each SN device must include the following hardware modules:

- A computational unit
- An 868 MHz radio chip
- A distance measurer device

The following components have been chosen to cover the outlined functionalities:

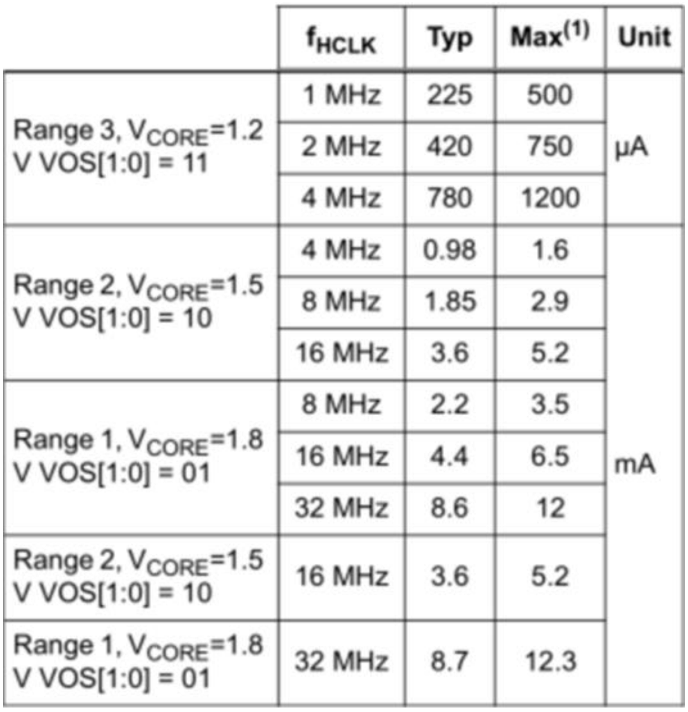
- NUCLEO-L152RE board of STMicroelectronics
- S2-LP radio chip for the 868 MHz communication
- DWM1001 UWB (Ultra-Wide Band) distance measurer

For this purpose, STMicroelectronics provides an evaluation kit (STEVAL-FKI868V1) that couples a S2-LP radio chip with a NUCLEO-L152RE. The evaluation kit provides an all in one hardware solution for Sub-GHz communication based on very low power consumption.

2.4.1 Microcontroller - NUCLEO-L152RE

The NUCLEO-L152RE mounts a STM32L152RE MCU, based on an ARM M3 microcontroller. The microcontroller provides various power operation modes, these can be viewed in Figure 15.

The ARM M3 processor provides adequate resources for this application, such as 80 KB of SRAM, 16 KB of EEPROM and 512 KB of Flash memory. Various interfaces are also provided, such as UART, SPI and I^{2}C.



	f_{HCLK}	Typ	Max$^{(1)}$	Unit
Range 3, V_{CORE}=1.2 V VOS[1:0] = 11	1 MHz	225	500	µA
	2 MHz	420	750	
	4 MHz	780	1200	
Range 2, V_{CORE}=1.5 V VOS[1:0] = 10	4 MHz	0.98	1.6	mA
	8 MHz	1.85	2.9	
	16 MHz	3.6	5.2	
Range 1, V_{CORE}=1.8 V VOS[1:0] = 01	8 MHz	2.2	3.5	
	16 MHz	4.4	6.5	
	32 MHz	8.6	12	
Range 2, V_{CORE}=1.5 V VOS[1:0] = 10	16 MHz	3.6	5.2	
Range 1, V_{CORE}=1.8 V VOS[1:0] = 01	32 MHz	8.7	12.3	

Figure 15: Power modes of the STM32L151 ARM M3 microcontroller

2.4.2 Sub-GHz Communication – S2-LP

The S2-LP radio chip features a very high reception sensitivity on the 868 MHz band (up to -140 dBm) and high transmission rates (up to 250 kbit/s). The radio chip reports a consume of current around 1 mA in sniff mode. The chip also provides antenna diversity, that can potentially increase signal robustness and reliability.

The high Rx sensitivity coupled with antenna diversity should help achieving reliable connectivity with RF propagation in the tight and multipath-prone space below the train, between the train cars and the tracks.

The high bit rate should help achieving the tight timing of a full train integrity assessment of 50 coupling in less than 5 seconds.

The good TX and RX energy efficiency help reducing the average SN energy consumption, reducing the load on the EH power supply.

The S2-LP radio also allows to set long packet preambles, which will be turned for optimal balance between the RX current consumption and RF channel occupation.

The typical power consumption data of the S2-LP radio chip are:

- 20 mA for TX
- 8.6 mA for RX

2.4.3 Distance Sensor – DWM1001

The DWM1001 is a UWB and BLE-ready module that offers RTLS (Real Time Localization System) based in Decawave’s DWM1001 IC and Nordic Semiconductor nRF52832 SoC. The device can be configured in two options:

- Anchor – reference device for measurement;
- Tag – device that requires the measurement.

The high-level block diagram of the device is depicted in Figure 16.

By default, the device has an accelerometer and two integrated antennas for both UWB and BLE. It is also provided with an embedded firmware that enables:

- Accurate UWB-based RTLS
- Data encryption for network connectivity

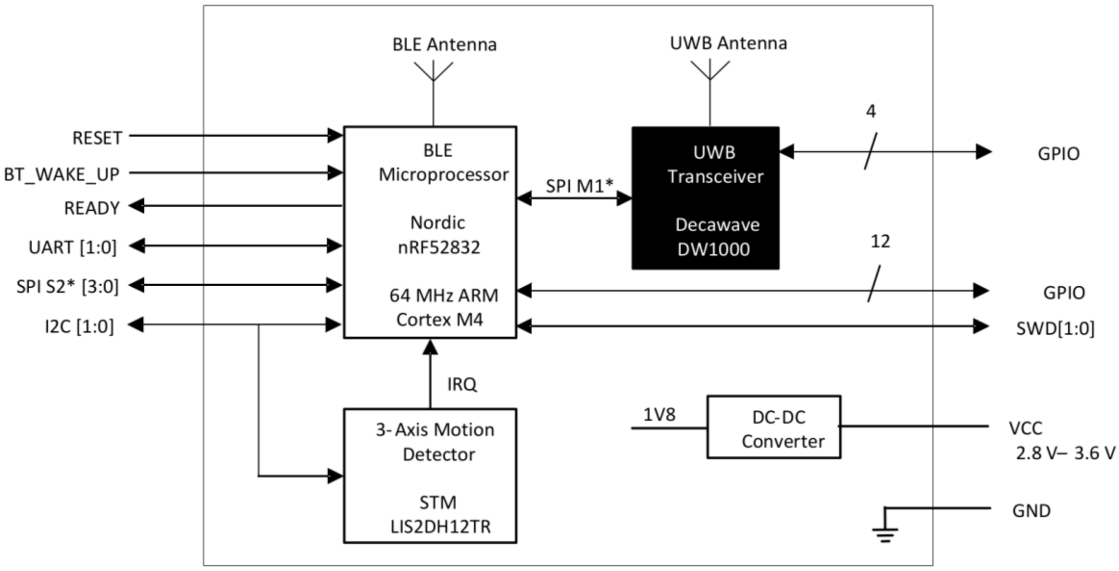
The device employs a TWR (Two-Way-Ranging) RTLS with up to thousands of tags. The same module can assume both tag and anchor roles by just toggling the settings. The BLE interface provides an easy to use interface through which the user can easily interact.

Apart from the Bluetooth interface, other ways of interfacing with the DWM1001 are available, like UART and SPI. This high-level architecture is displayed in Figure 17, where in ETALON application the “user host device” role is covered by the NUCLEO micro-controller unit. These two interfaces, plus the BLE interface, are provided with APIs by the firmware. The device also provides a low power hardware and software architecture for minimizing energy consumption or to employ the solution where energy consumption is critical.

The device is ideal for localization, but it can also be used to measure a single distance between two devices, one configured as tag and the other as anchor. The device has a ranging accuracy within

10 cm and a 60 m line-of-sight typical range. It can be power using a power supply source between 2.8 V and 3.6 V.

The DWM1001 UWB module is compliant with the IEEE 802.15.4-2011 standard. The device provides a UWB channel 5 (6.5 GHz) printed PCB omnidirectional antenna, although an ad-hoc antenna is being designed for the ETALON application prospective, this aspect is discussed more in depth in Section 5.



*SPI M1 is nRF52 SPI master 1, SPI S2 is SPI slave 2

Figure 16: DWM1001 High Level Block Diagram

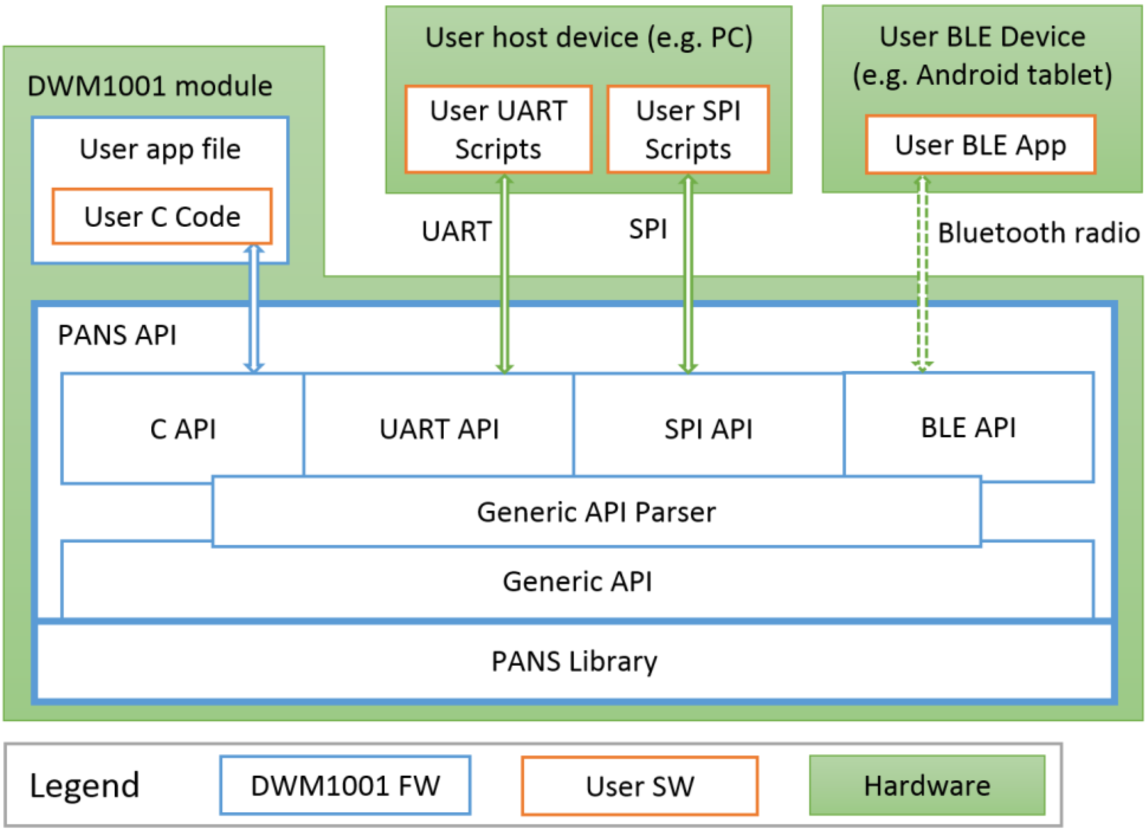


Figure 17: High-Level Architecture of DWM1001 Firmware vs. User Software

2.4.4 Control Module – Raspberry Pi 3 Model B+

The CM will be implemented on a Raspberry Pi 3 Model B+ board running Linux operating system and using a 7” Touch Screen colour display.

The Human Interface of the CM will include:

- A means to specify the movement state of the train (moving or stationary);
- A means to request the network formation and display the result;
- A means to request a TI request and display its result.

2.5.1 WSN packets transactions for each SN

The typical RF activity for a single SN requires the exchange of 26 packets (TX or RX), as follows:

- 18 packets (TX or RX) on the forward propagation (meaning from the CM towards the end of the train, see Section 2.3) as follows:
    - 5 TX packets:
        - 1 to transmit the note integrity assessment;
        - 3 to acknowledge the integrity assessments of the other three nodes;
        - 1 to forward the integrity assessment request to the nodes belonging to the next coupling.
    - 13 RX packets:
        - 3 to receive the integrity assessments of the other three nodes;
        - 3 to receive the acknowledgements of the other three nodes to the integrity assessment sent by this node;
        - 4 requests to start integrity assessment received from the nodes in the previous coupling (closer to the CM);
        - 3 requests to start the integrity assessment directed by the other nodes of this coupling to the nodes in the next coupling (further away from the CM).
- 8 packets (TX or RX) on the backward path (from the tail of the train up towards the CM):
    - 1 TX packet:
        - 1 forward of the TI status vector to the previous coupling (closer to the CM);
    - 7 RX packets:
        - 4 carrying the TI status vector from the previous coupling (further away from the CM);
        - 3 forwards of the TI status vector to the previous coupling (closer to the CM) sent by the other nodes in this coupling.

The WSN does not require other packets, e.g. for network maintenance.

Besides the outlined packets, that are mandatory for the proper operation of the WSN, the nodes may overhear (thus meaning RX) some packets not addressed to them, from the mandatory exchanges between the nodes belonging to other train couplings, as shown in Figure 18.

The extent of the overhearing area is highly dependent on the actual RF propagation conditions, which can vary significantly.

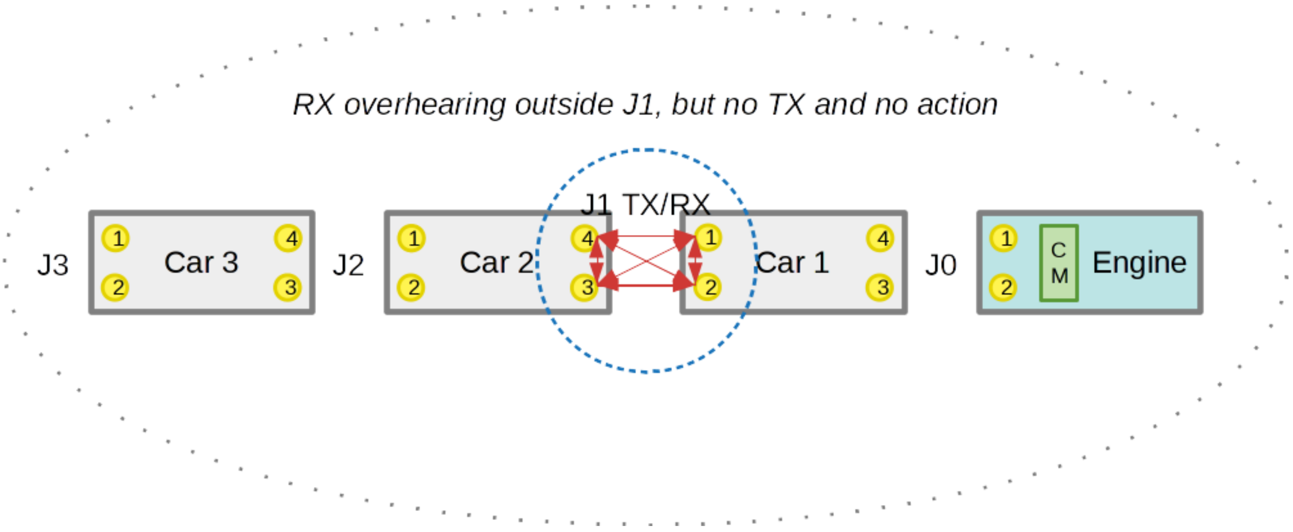


Figure 18: Variable RX overhearing range of nodes

2.5.2 Distance Sensor Interface Specification

The DS is implemented using Ultra-Wide Band (UWB) communication modules that are capable to accurately measure the physical distance separating the UWB communicating nodes.

Each SN of the WSN includes a UWB-based DS, which is capable to assess the physical distance to the UWB-based DS of other nearby WSN sensor nodes.

DS ranging of nearby peers is used to detect breaches of train integrity as follows. If the reported distance to the peer UWB-based DS nodes across a coupling (e.g., between the WSN SNs “1” and “4” or “2” and “3” in Figure 9) is above a pre-defined CLD, then the WSN SNs assume that the coupling mechanical integrity is lost and report the status to the CM on the leading locomotive using the WSN communication protocol.

Hence, the WSN of SNs periodically assesses the integrity of the whole train by periodical assessments of the integrity of each coupling of the train, as described above. For this purpose, each WSN SN needs to:

- interface with the attached DS using a local communication bus;
- retrieve the distance to the facing DS across a train coupling;
- compare the distance to a given threshold;
- assess the integrity of the coupling as follows:
  - if the distance is below the threshold the coupling is intact;
  - if the distance is above the threshold, coupling integrity is compromised;
  - if no valid distance measurement is given, coupling integrity is compromised.
- store and report the result of coupling integrity assessment using the WSN protocol.

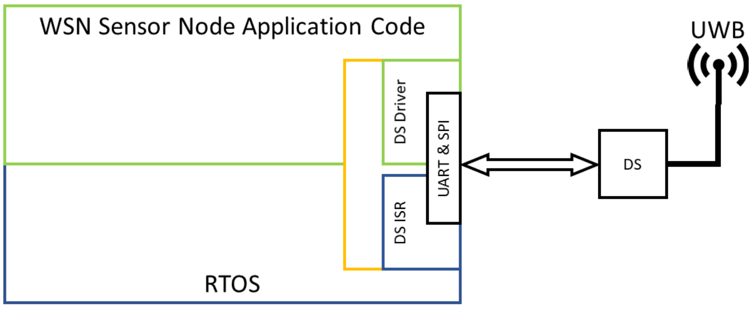


Figure 19: Interface between the DS driver and the SN

In the following it is provided the design and implementation guidelines for the hardware, software, messages and protocol interface of the DS driver and interface between the SN and the DS driver (SN-DS).

As shown in Figure 19, the DS hardware sensor measures the physical distance to its peer(s) using an UWB network. It interfaces with the SN using a serial connection (e.g., UART and SPI), which is handled by code running in the RTOS (Real-Time Operating System) of the SN, such as ISR(s) (Interrupt Service Routine) and the DS driver implemented as RTOS task(s). At the same time, the DS driver code provides a mechanism to interface with the WSN sensor node application code.

More specifically:

- DS-specific logic and operation should be fully included in the DS driver – proper DS operation should not depend on specific interaction with the host SN for initialization, peer discovery, distance measurement or other DS functions;
- The DS driver should:
    - Be implemented as a task(s) and ISR(s) of the operating system used by the SN;
    - Minimize critical resource consumption (execution time (both ISR and normal tasks), data and code space);
    - Avoid as much as possible unwanted interferences with the operation of the SN;
- Hardware communication between the DS and the SN where the DS driver is implemented using a serial line (e.g. UART and SPI);
- All messages between the DS and its driver implemented on the SN should include reliable error checking (e.g. CRC), which should be used to reject any corrupted message;
- SN-DS communication (between the DS and the SN) should:
    - Be kept minimal and use a suitable simple mechanism, preferably synchronous (function calls and call-backs);
    - Adhere to REST design principles as much as possible, compatible with the REST design of the WSN protocol – mostly query-reply, independent on previous message exchanges.

- SN-DS communication should include:
    - DS initialization (and optional turn on) command:
        - Is sent by the SN to the DS driver:
            - Whenever the SN initializes itself (e.g. at power up);
            - During the network discovery.

        - During initialization, the DS should re-discover its DS peer across the coupling (DS should forget its previous peer if any);
        - The DS driver reply should include:
            - The initialization result (OK or error);
            - DS ID;
            - Peer DS ID (across the coupling or none if no peer was detected).

        - The DS and the driver may measure the distance:

            - Upon request from the host SN, in the allowed time;

            - Periodically, independent on the SN requests and provide the latest reading as reply to the SN request. In this case, the period od DS measurements should be designed such a way to always provide fresh enough readings as reply to the SN requests to satisfy the specifications of the project on the overall detection time of TI failures;

    - DS turn off command (optional):

        - It can be sent by the SN to the DS driver when the TI assessment is no longer required (e.g. the train reaches its destination and becomes stationary or its cars are moving elsewhere).

3. POWER CONSUMPTION ANALYSIS

For analysing the system energy and power consumption, the system has been divided in two main components:

- Microcontroller and Communication module
- Distance Sensor

The two components are separately analysed regarding both the power and energy consumption.

3.1 Power Consumption of the Microcontroller and Communication Module

The power consumption of the microcontroller and the radio chip are analysed as a whole, since the producer provides these components as a single kit.

3.1.1 Microcontroller Power Consumption

This component has the least impact on the power consumption. As outlined in Section 2.4.1, specifically in Figure 15, the device is optimised for an extreme low-power consumption processing. This can potentially reach a current consumption of about 225 µA; this value can be achieved by slowing down the clock frequency to the minimum frequency of 1 MHz and supplying a minimal 1.2 V to the ARM core.

This configuration can achieve an extremely low power consumption that drains a power in the range of µW (values around 300 µW). At maximum clock frequency and voltage supply, the device can reach some mW of power consumption (around the value of 15 mW).

At this stage, the best trade-off between power consumption and processing power has not been defined, leaving the device in the default configuration.

The most resource intensive processes for this device are the security routines. Message encryption and decryption procedures potentially require a lot of computational effort, especially for asymmetric encryption. For this purpose, some light weighted security libraries have been chosen, in order to have a robust encryption and not present a high impact on power consumption.

3.1.2 Radio Communication Power Consumption

As already outlined in Section 2.4.2, the communication model is based on the S2-LP sub-1GHz transceiver. This chip is designed for ultra-low power designs, giving the possibility of having small current consumptions during transmission and reception below 10 mA.

The consumption model regarding the already described protocol, is particularly challenging and not easily predictable. This issue is cause by the overhearing feature, this introduces a random factor that triggers RX current consumption due to the overhearing. The device presents a current consumption of 1 mA in sniff mode.

As previously outlined and from the simulation results, it is possible to have an estimate of the radio chip power consumption for a TI check procedure without taking into account the overhearing consumption. A SN requires the transition of 26 packets (6 TX and 20 RX) of the procedure, having a typical 10 mA consumption for TX and a 7 mA consumption for RX; also from the simulations it is possible to estimate a radio transition delay in the worst case of 3 ms.

First of all, it is possible to compute the average current consumption for the protocol as:

$$avg\ packet\ current = \frac{TX\ packets * TX\ current\ consumption + RX\ packets * RX\ current\ consumption}{total\ number\ of\ packets} = 7.7mA$$

Knowing the transition delay, it is possible to compute the required average electric charge of a packet as:

$$packet\ avg\ electric\ charge = transition\ delay * avg\ packet\ current = 6.4167 * 10^{-6}mAh$$

Having the value for a single packet, it is possible to estimate the employed electric charge for the whole TI check procedure for a single node:

$$avg\ electric\ charge = number\ of\ packets * packet\ avg\ electric\ charge = 1.6683 * 10^{-4}mAh$$

With this data, it is possible to estimate the energy consumption and the average power consumption of a SN for TI check procedure (having a supply voltage of 3.6 V):

$$SN\ energy\ consumption\ for\ TI\ check = 2.1622 * 10^{-3}J$$

$$SN\ avg\ power\ consumption\ for\ TI\ check = 27.72mW$$

The radio chip has also presents a sleep state for minimizing the current consumption, reaching a value of 700 nA.

With this data, it is possible to estimate the energy and power consumption of the radio chip. The data are reported in , considering the chip in sleep mode the remaining time. As previously explained, the calculations do not take into account the RX overhearing consumption, since this effect is random and extremely variable with regards to the environment.

Table 3: Energy and power radio consumption data for various TI intervals

TI interval check (s)	Electric charge used (mAh)	Average current consumption (mA)	Energy consumption @3.6V (J)	Average power consumption @3.6V (W)
1	$1.67013 * 10^{-4}$	$6.01245 * 10^{-1}$	$2.16448 * 10^{-3}$	$2.16448 * 10^{-3}$
3	$1.67402 * 10^{-4}$	$2.00882 * 10^{-1}$	$2.16952 * 10^{-3}$	$7.23174 * 10^{-4}$
5	$1.6779 * 10^{-4}$	$1.20809 * 10^{-1}$	$2.17456 * 10^{-3}$	$4.34913 * 10^{-4}$
10	$1.68763 * 10^{-4}$	$6.07545 * 10^{-2}$	$2.18716 * 10^{-3}$	$2.18716 * 10^{-4}$
30	$1.72652 * 10^{-4}$	$2.07182 * 10^{-2}$	$2.23756 * 10^{-3}$	$7.45854 * 10^{-5}$
60	$1.78485 * 10^{-4}$	$1.07091 * 10^{-2}$	$2.31316 * 10^{-3}$	$3.85527 * 10^{-5}$

3.2 POWER CONSUMPTION OF THE DISTANCE SENSOR

The Distance Sensor is based on UWB technology. This technology-based ranging uses very brief radio impulses at a very high frequency (with a central band of 6.5 GHz).

Most of the time the module will remain in a dormant state, in which the current consumption is extremely low (in the order of tens of µA). The device can be powered with a voltage in the range of 2.8-3.6 V, resulting in a power consumption that is below 100 µW in dormant state.

To start a ranging procedure, the device firstly performs a wake-up procedure and subsequently transmits a radio impulse and receives one or more impulses (depending on the number of nodes that respond) as the result of the measurement.

Once the ranging has been performed, the device returns in the dormant mode until the next ranging.

In Figure 20 it is shown the current consumption of the module in details. In the first phase the module is the dormant very low power consumption mode (12 µA current consumption). Once the time between ranges is elapsed, the devices firstly wakes up, performing a spike of current consumption and stabling around 6 mA. Before starting the ranging, the device remains for a brief period in idle mode, with a current consumption of 13 mA. The module will now perform two ranging transmission pulses, in between which it will receive some response impulses depending on the quantity of anchor devices that reply (3 anchors in Figure 20). These impulses are instants during which the module will have the highest power consumption, with peaks over 150 mA.

In order to have a better view of the actual energy consumption during these phases it is possible to approximate and integrate the current consumption over time. In Figure 21 the power consumption is reported by approximating to linear steps each impulse or in general any difference in current consumption.

The calculated values are reported in Table 4, where the consumed electric charge is calculated by approximating the impulse to a perfect rectangular impulse during which it has a constant current consumption.

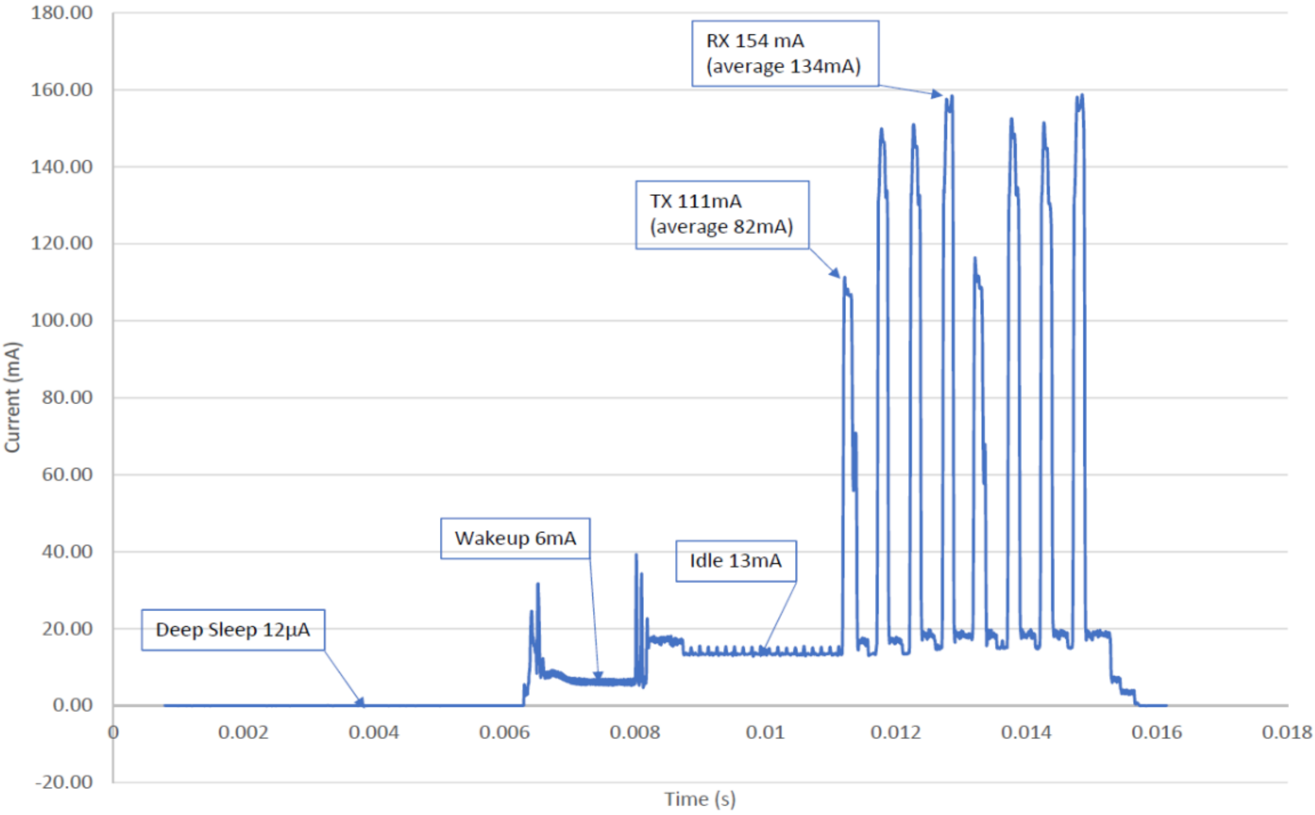


Figure 20: DWM1001 power consumption graph

In Table 4 the data reported in red are referred to the ranging pulses, where there is the highest power consumption; while the remaining data is referred to the other states of wakeup and idle. The reported calculations only take in consideration the period during which the device is active and performs the ranging, while it is omitted the deep sleep consumption, thus meaning that the reported consumption is referred specifically to the raging phase.

The time required for the device to activate and perform the measurement and finally going back in deep sleep sums up to approximately 9ms. The timing partially depends on the quantity of nodes that answer the ranging request, thus resulting in a briefer ranging time for fewer answering nodes and vice versa.

From the computed data it possible to calculate the major power requirements for a ranging action performed by the node. In Table 5 the following power and energy data are reported:

- Total time required for the device to wake up, perform the ranging and return in deep sleep
- The total electric charge consumed for the process
- The average current consumption for the process
- Total energy consumption for the process (with a power supply of 3.6V)
- The average power consumption for the process (with a power supply of 3.6V)
- The power consumption peak for the process (with a power supply of 3.6V).

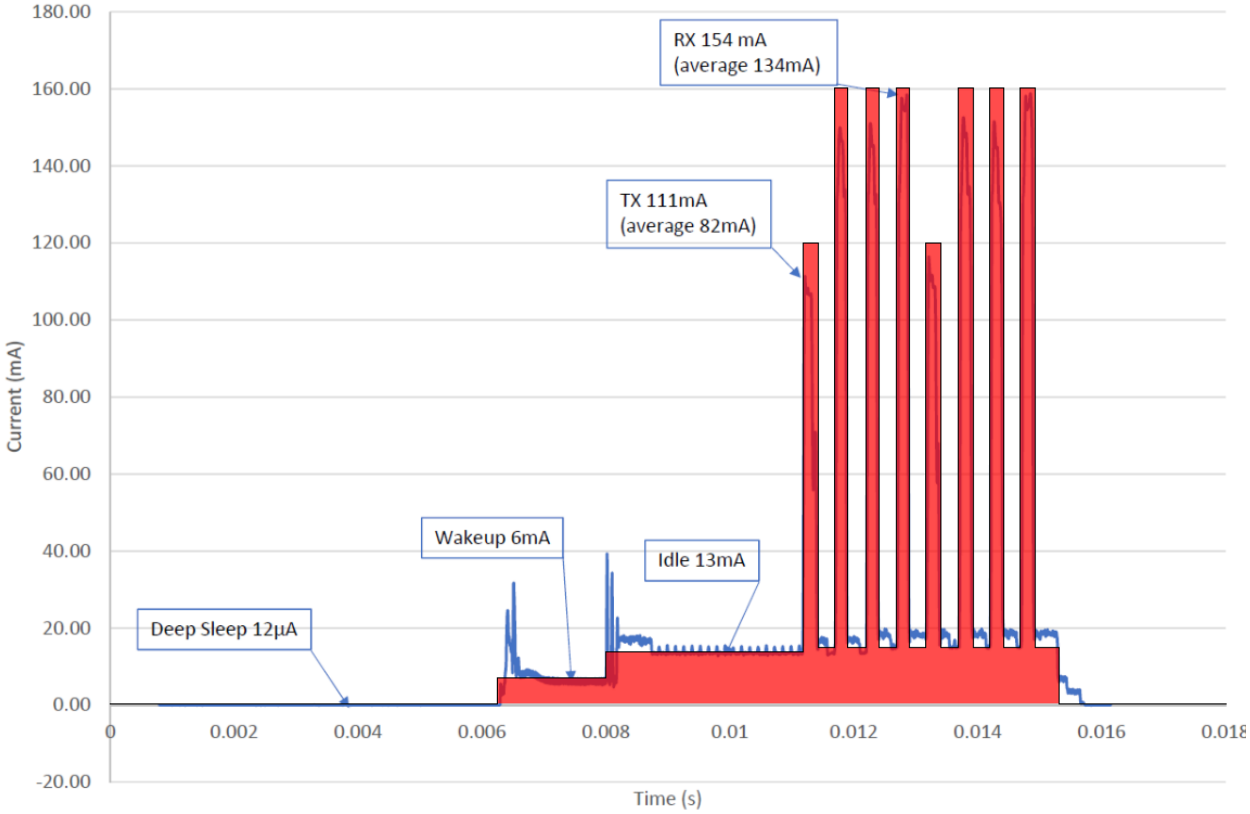


Figure 21: DWM1001 power consumption approximation graph

Since TI aims at checking integrity at defined intervals (1s, 5s, 30s, etc.), only a single ranging process needs to be performed within the interval, the remaining time the device remains in the deep sleep mode.

In

Table 6 the energy and power data are reported taking in account various TI check periods (specifically for the intervals of 1s, 3s, 6s, 10s, 30s and 60s). It is noticeable that the average current consumption is inversely proportional to the TI check period, meaning that for larger intervals the average current consumed for the TI check decreases. The same behaviour is reflected for the average power consumption.

Table 4: Calculation of the consumed electric charge for ranging through approximation

Step	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Time (s)	$1.73646 * 10^{-3}$	$3.17475 * 10^{-3}$	$2.4556 * 10^{-4}$	$2.2802 * 10^{-4}$	$2.1048 * 10^{-4}$	$2.631 * 10^{-4}$	$2.1048 * 10^{-4}$	$2.4556 * 10^{-4}$	$2.1048 * 10^{-4}$	$2.2802 * 10^{-4}$	$2.4556 * 10^{-4}$	$2.4556 * 10^{-4}$	$2.4556 * 10^{-4}$	$2.2802 * 10^{-4}$	$2.2802 * 10^{-4}$	$2.2802 * 10^{-4}$	$2.4556 * 10^{-4}$	$3.6834 * 10^{-4}$
Current (A)	$6.66675 * 10^{-3}$	$1.30668 * 10^{-2}$	$1.2 * 10^{-1}$	$1.41335 - 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$	$1.2 * 10^{-1}$	$1.41135 * 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$	$1.6 * 10^{-1}$	$1.41335 * 10^{-2}$
Used electric charge (mAh)	$3.21571 * 10^{-6}$	$1.15233 * 10^{-5}$	$8.18533 * 10^{-6}$	$8.952 * 10^{-7}$	$9.35467 * 10^{-6}$	$1.03292 * 10^{-6}$	$9.35467 * 10^{-6}$	$9.64062 * 10^{-7}$	$9.35467 * 10^{-6}$	$8.952 * 10^{-7}$	$8.18533 * 10^{-6}$	$9.62698 * 10^{-7}$	$1.09138 * 10^{-5}$	$8.952 * 10^{-7}$	$1.01342 * 10^{-5}$	$8.952 * 10^{-7}$	$1.09138 * 10^{-5}$	$1.44609 * 10^{-6}$

Table 5: Energy and power data during a ranging procedure

Total amount of time for ranging (s)	Total electric charge used (mAh)	Average current consumption (mA)	Total energy consumption @3.6V (J)	Average power consumption @3.6V (W)	Peak power consumptio n @3.6V (W)
$\mathbf{8.78755 * 10^{-3}}$	$9.92403 * 10^{-5}$	40.6558	$1.28645 * 10^{-3}$	$1.46361 * 10^{-1}$	$5.76 * 10^{-1}$

Table 6: Energy and power ranging consumption data for various TI intervals

TI interval check (s)	Electric charge used (mAh)	Average current consumption (mA)	Energy consumption @3.6V (J)	Average power consumption @3.6V (W)
1	$1.02544 * 10^{-4}$	$3.6916 * 10^{-1}$	$1.32897 * 10^{-3}$	$1.32897 * 10^{-3}$
3	$1.09211 * 10^{-4}$	$1.31053 * 10^{-1}$	$1.41537 * 10^{-3}$	$4.71792 * 10^{-4}$
5	$1.15878 * 10^{-4}$	$8.34321 * 10^{-2}$	$1.50178 * 10^{-3}$	$3.00356 * 10^{-4}$
10	$1.32544 * 10^{-4}$	$4.7716 * 10^{-2}$	$1.71777 * 10^{-3}$	$1.71777 * 10^{-4}$
30	$1.99211 * 10^{-4}$	$2.39053 * 10^{-2}$	$2.58177 * 10^{-3}$	$8.60592 * 10^{-5}$
60	$2.99211 * 10^{-4}$	$1.79527 * 10^{-2}$	$3.87777 * 10^{-3}$	$6.46296 * 10^{-5}$

Regarding the Electric charge and the energy consumption, these parameters are directly proportional with the TI check interval. This is due to constant consumption during deep sleep mode, but since the consumption in this state is minimal, the energy and electric charge consumption are minimal.

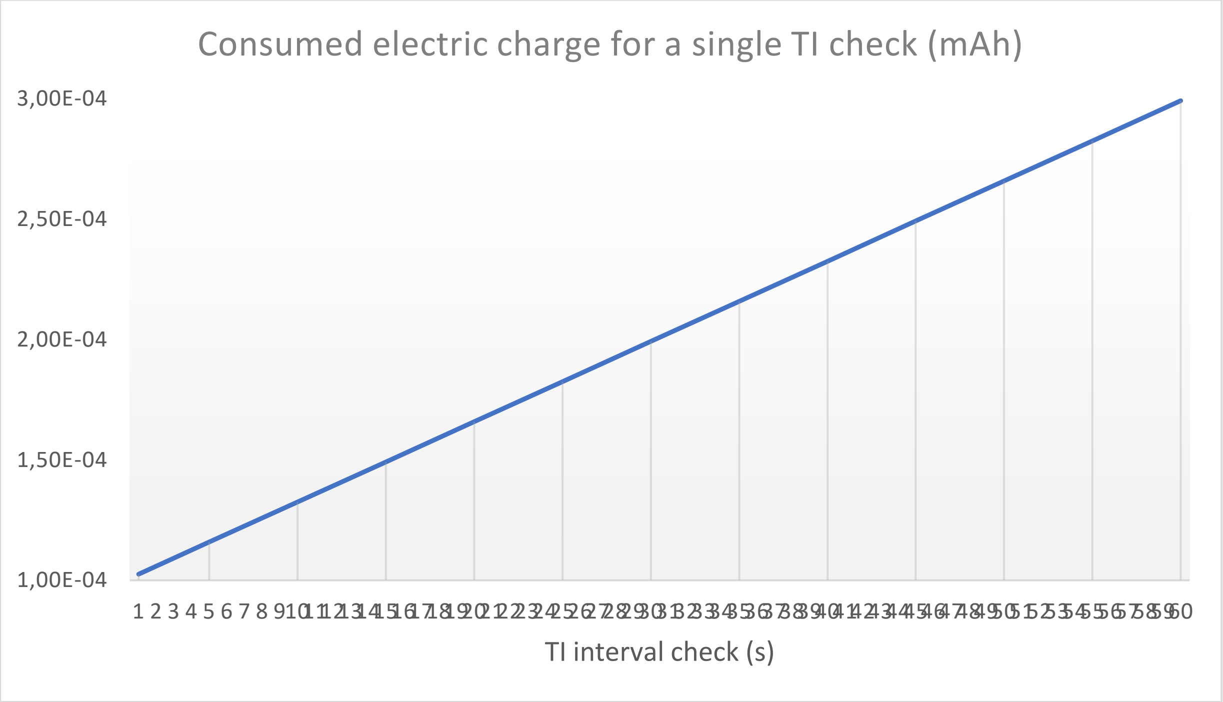


Figure 22: Consumed electric charge of the DS module in the interval 1s – 60s

In order to give a better understanding of the reported data, this has been plotted on some graphs. In Figure 22 the consumed electric charge is plotted with respect to the TI interval check (variating from 1s up to 60s with a linear scale). This interval has been chosen since it is envisaged that the TI check period will variate from a minimum of 5 seconds to a maximum of 1 minute or more. Within this range the consumed electric charge for a TI check process varies from approximately a minimum of $10^{-4}mAh$ up to a maximum of $3 * 10^{-4}mAh$.

From the electric charge consumption, it is possible to compute the energy consumption of the device by forecasting a 3.6V voltage for the power supply. This data is then highly correlated to the electric charge consumption and practically scaled by the voltage supply factor and normalised by the time and current factors. In Figure 23 it is displayed the energy consumption in Joule with respect to the TI interval check (variating from 10s up to 60s with a linear scale). Within this range the consumed energy for a TI check process varies from approximately a minimum of $1.3 * 10^{-3}J$ up to a maximum of $3.8 * 10^{-3}J$.

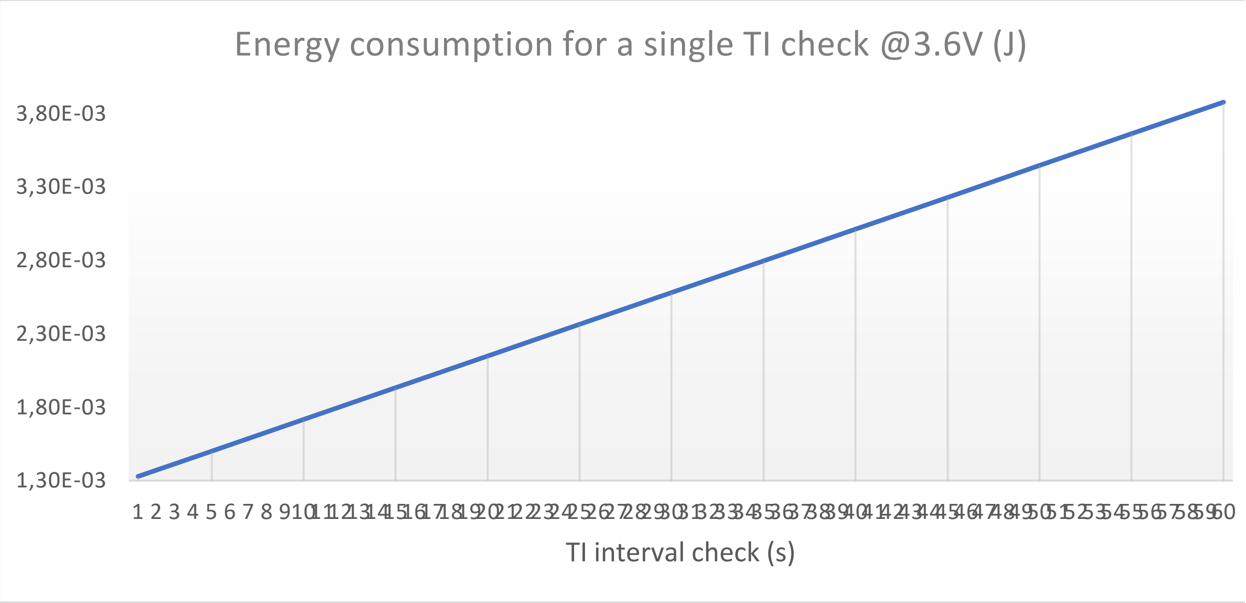


Figure 23: Consumed energy of the DS module in the interval 1s – 60s

Continuing this analysis, the next factor considered is the average power consumption for a single TI check. Taking always in consideration a 3.6V voltage for the power supply, in Figure 24 the average power consumption in Watts is reported with respect to the TI interval check (always variating from 1s up to 60s with a linear scale).

It can be seen that the average power consumption has a hyperbolic trend, having a maximum consumption of approximately 1.3mW for a 1s TI check interval and drastically decreasing to a minimum of approximately 65µW for a 60s TI check interval.

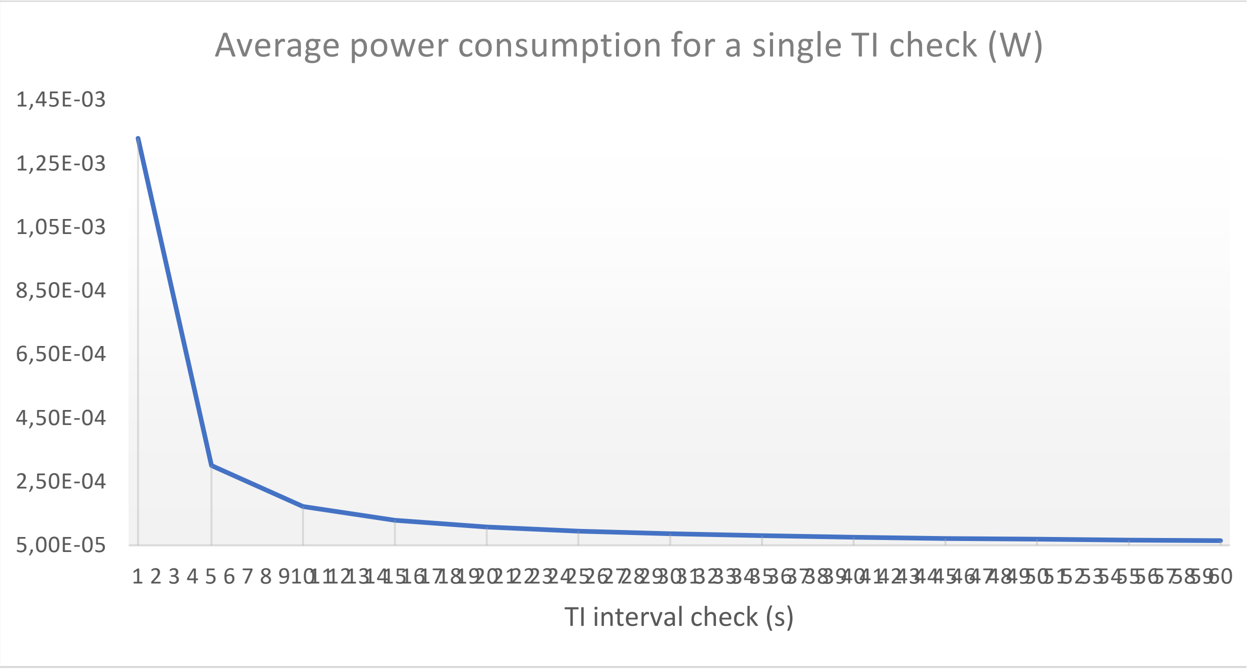


Figure 24: Average power consumption of the DS module in the interval 1s – 60s

4. SIMULATION OF THE COMMUNICATION PROTOCOL

4.1 Introduction

The WSN network simulation has three main purposes:

1. Check the suitability of the network protocol for train integrity assessment within the constraints of the requirements (timing, energy, error recovery);
2. Assess the optimization potential of the protocol and necessary optimizations;
3. Implement the application code very close to actual implementation on the embedded operating system (FreeRTOS) to speed up its porting, testing and debugging.

4.2 Simulation of Suitability of WSN Protocol

The WSN protocol main purposes are:

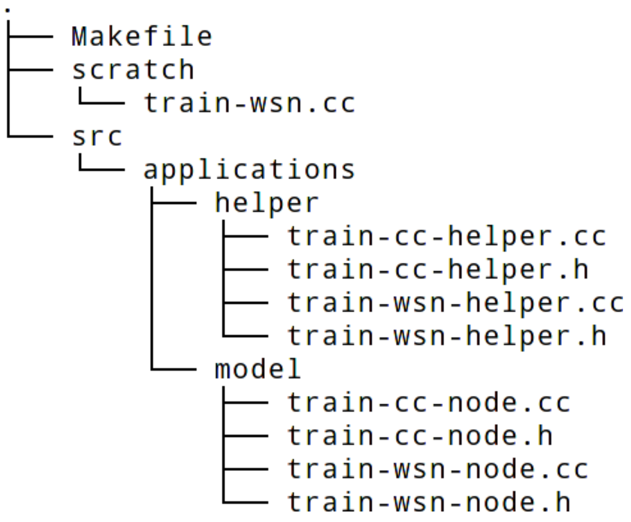
- Periodically collect train integrity data from all train couplings and report it back to train CC within the specified time (and energy) constraints;
- Avoid providing false confirmations of train integrity;
- Report measurement or other operation errors (and unknown state of train integrity in case of errors).

At this stage, the simulation model focused mainly on the first point, i.e. its suitability to collect train integrity data within the time constraints, and a first estimation of node average energy requirements.

4.2.1 Simulation Software

For this purpose, application behaviour described in Section 2.3 and Appendix 6 was implemented in the widely-used NS-3 simulator for wireless networks, in C++ programming language. This implementation allows maximum flexibility in accessing simulation resources and is optimal for both implementation and simulation speed.

Figure 25 shows the file structure of the simulation project.



```
.
├── Makefile
├── scratch
│   └── train-wsn.cc
└── src
    └── applications
        ├── helper
        │   ├── train-cc-helper.cc
        │   ├── train-cc-helper.h
        │   ├── train-wsn-helper.cc
        │   └── train-wsn-helper.h
        └── model
            ├── train-cc-node.cc
            ├── train-cc-node.h
            ├── train-wsn-node.cc
            └── train-wsn-node.h
```

Figure 25: Structure of NS-3 simulation project to check for WSN suitability

File *train-wsn.cc* under directory *scratch* is the top-level file of the simulation. It includes the *main()* function which declares:

- Simulation-specific command line arguments;
- Various random generators;
- Shared RF channel;
- WSN nodes for:
    - CC on train engine;
    - Sensing nodes for all train couplings;

Each node includes a model of the physical layer (PHY), medium access control (MAC) layer, and a carrier-sense multiple access collision avoidance (CSMA-CA) layer.

PHY layer is used to simulate the propagation of the RF packets in the physical environment along the train. We use for this the *SimpleSpectrumChannel* model provided by NS-3 simulator. Since the RF propagation conditions along the train are unknown and can vary widely based on, e.g., environmental and terrain conditions, as well as mounting position of the nodes on the train cars, we have modified the RF propagation model of the NS-3 channel to a worst-case rectangular-shaped attenuation with the distance: no attenuation up to a given distance (e.g., 60 m, which is about 3 train cars) and very strong attenuation beyond that. This type of attenuation forces equal-amplitude conflicts in case two packets overlap within the propagation window, which would prompt to always drop both packets, hence to maximum packet loss.

MAC layer allows the nodes to sense the status of the RF channel before engaging in transmissions. In the actual implementation (Section 4.2.4) we have disabled the collision avoidance feature of the layer because we handle it in the application itself.

For each node we define also the position and the network address at node creation time, since the focus of this simulation is the behavior of the operational network, not the network formation process (which is implemented in the application-ready code described in Section 4.2.4).

Finally, the *main()* function starts the network simulation, and ends once the simulation finishes.

Files under *src/applications/helper* define the application-specific implementation of various NS-3-specific housekeeping functions, while the files under *src/applications/model* implement the behavioral code of the actual WSN application as follows: *train-cc-node.cc* implements the application running on the CC node locate on the train engine, and train-wsn-node.cc implements the application running on each sensor node that monitors a train coupling. The implementations follow closely the behavior described in detail in Section 2.3 and Appendix 6.

4.2.2 **Results**

NS-3-based simulation set up as explained in Section 4.2.1 was used mainly for three purposes:

1. To check the validity of the WSN protocol as it was defined in Section 2.3 and Appendix 6;
2. To check the distribution of train integrity assessment and collection times;
3. To obtain a rough estimation of the energy consumption of the nodes.

For this purpose, we have run the simulation multiple times, with different packet lengths (all packets of the same length), and each time randomizing protocol parameters regarding the MAC to cover most operation and propagation cases that may occur in a real network.

Table 7 shows the train integrity assessment times obtained from NS-3-based simulations for different packet payloads, for 250 kbit/s TX data rate, packet overhead 35 bytes (preamble, sync word, etc.), and 10 ms measurement time of coupling separation.

Table 7: Train integrity assessment time simulation results for various packet payloads, for 250 kbit/s TX data rate, 35 bytes of packet overhead (preamble, sync word, CRC, etc.), and assuming 10 ms measurement time of coupling separation.

Payload (bytes)	Simulation runs	Average TI assessment time (s)	Minimum TI assessment time (s)	Maximum TI assessment time (s)
90	895	4.35	4.0	4.7
80	922	4.24	3.9	4.6
70	908	4.04	3.6	4.5
60	897	3.86	3.5	4.2

The distribution of the measurements is shown in Figure 26, Figure 27, Figure 28, Figure 29. As expected, the randomization of simulation parameters spreads the time needed by the WSN to perform train integrity assessment, but for all payload lengths considered this time stays below the required maximum time of 5 s.

We should also note that the integrity assessment time may further decrease because 10 ms allocated for the measurement of coupling separation is rather conservative. The actual time required for measurement of coupling separation will be determined after the DS sensor will be made available.

Average node TX energy consumption is well approximated by the analysis presented in Section 3.1.2. However, RX energy has a significant overhearing component, which is due to nodes receiving packets that are not directed to them. The overhearing depends very much on the propagation conditions, which can be determined only experimentally, for the actual mounting solution of the nodes on the train cars and for the most various propagation conditions that can be encountered in exploitation.

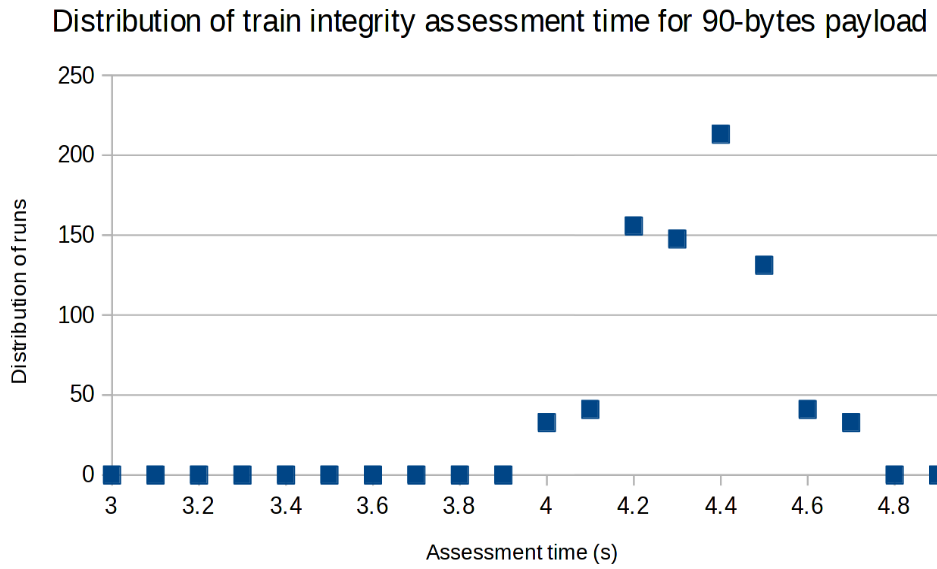


Figure 26: Distribution of train integrity assessment times for 90-byte packet payloads, 250 kbit/s TX data rate, 35-byte packet overhead, and 10 ms measurement time of coupling separation

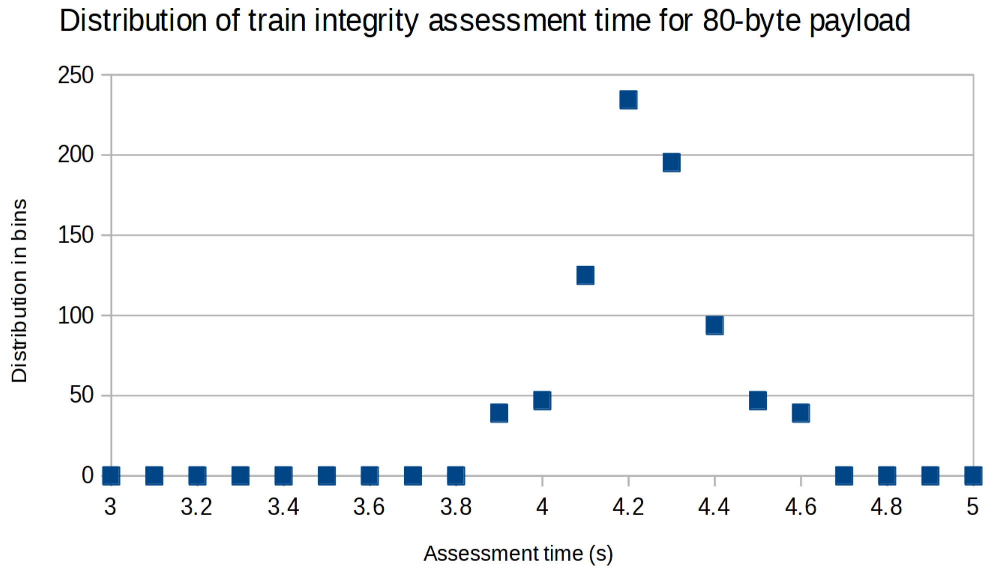


Figure 27: Distribution of train integrity assessment times for 80-byte packet payloads, 250 kbit/s TX data rate, 35-byte packet overhead, and 10 ms measurement time of coupling separation

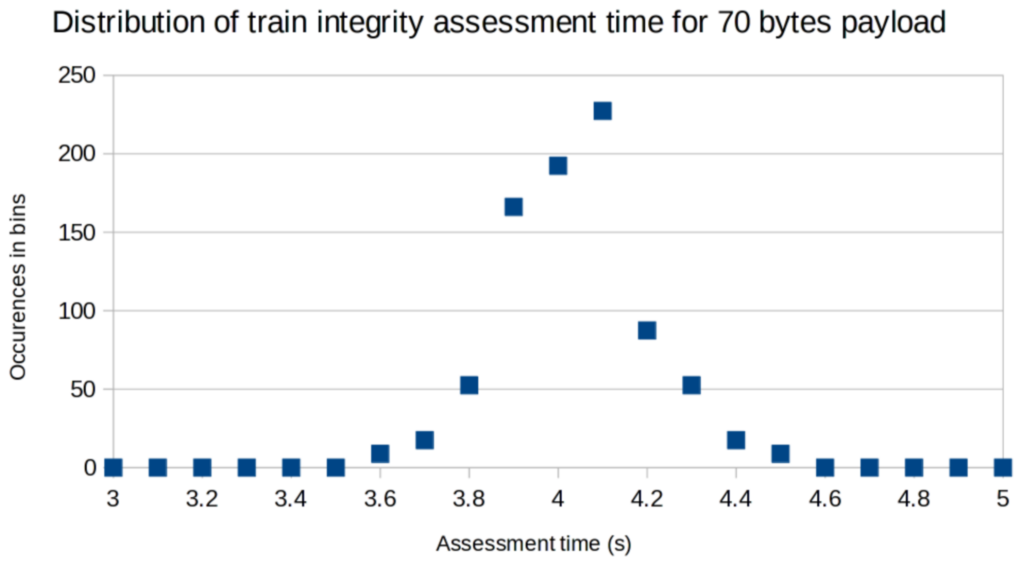


Figure 28: Distribution of train integrity assessment times for 70-byte packet payloads, 250 kbit/s TX data rate, 35-byte packet overhead, and 10 ms measurement time of coupling separation

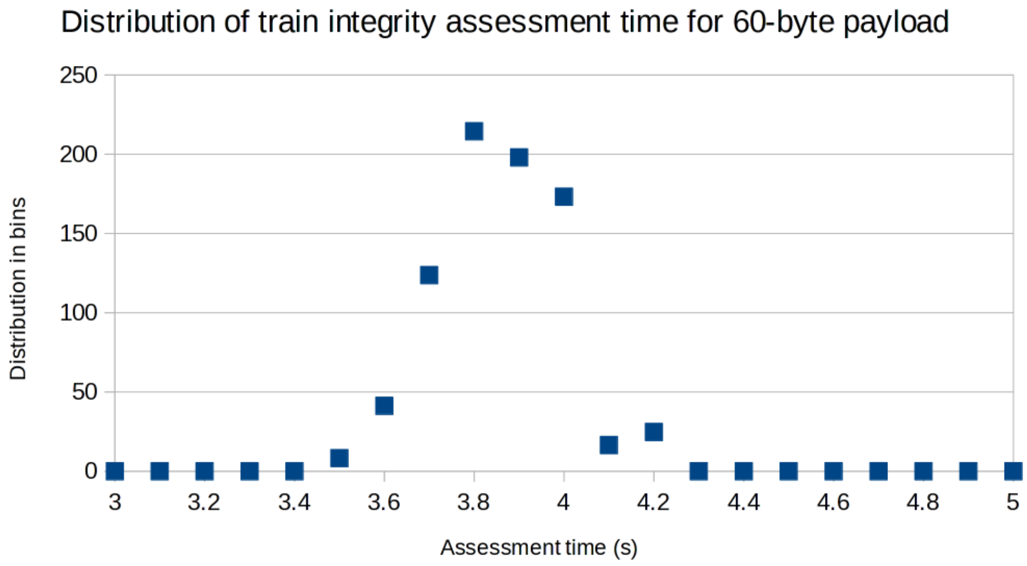


Figure 29: Distribution of train integrity assessment times for 60-byte packet payloads, 250 kbit/s TX data rate, 35-byte packet overhead, and 10 ms measurement time of coupling separation

4.2.3 Exploration of Optimization Potential of WSN Protocol

To reduce both energy consumption and assessment time, we have considered using variable length packets for the forward and the backward propagation of the data through the WSN.

Specifically, packet size during the forward pass in which the integrity of all couplings is assessed (see Section 2.3.2 and Section 2.3.3) can be much smaller because there is no need to transfer each time the whole vector with all the measurements. In fact, only the byte of the current coupling needs to be transferred while assessing the integrity of each coupling.

The vector with the assessment results for all couplings can be assembled on the return path, which is described in Section 0. As the vector propagates back to the CC, the nodes in each coupling will fill the assessment result in the entry corresponding to their coupling.

Shorter packets on the forward propagation path will bring most energy reduction benefits, since according to the analysis in Section 3.1.2 most of the network traffic occurs in that operation phase (roughly twice as many packets than during the backward propagation).

Other optimization venue is slotted communication of inter-coupling nodes (TDMA - Time Division Multiplexing Access), to increase the predictability of protocol propagation time and reduce the potential for packet collisions.

Optimizations will be considered during the implementation of the application code, described in Section 4.2.4.

4.2.4 Simulation for Test and Validation of WSN Application Code

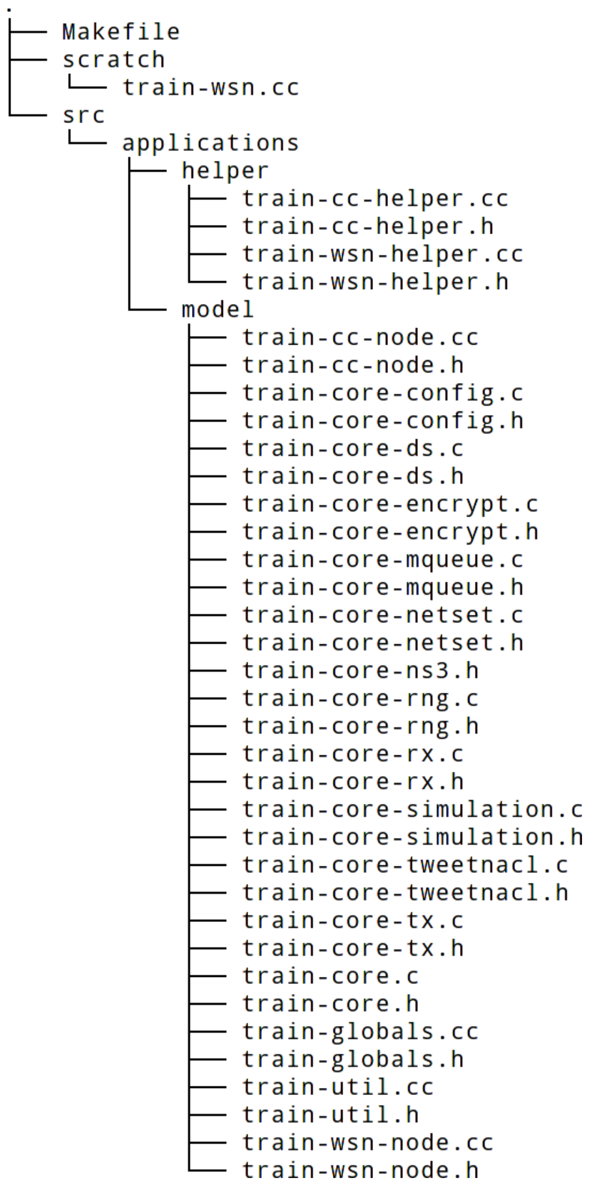
The main purpose of this simulation model is twofold:

1. Implement and validate the application behavioral code in C language and written close to the actual implementation on the embedded real-time operating system (FreeRTOS). This will allow to port the simulated application code on the nodes with just minor modifications, accelerating application validation and debug;
2. Extend the application behavior to include also network formation and encryption requirements.

Implementation

For this purpose, the main application behavior described in Section 2.3 and Appendix 6 is extended with network formation and security functions, described in Section 2.2. Code is implemented this time in C programming language and are created stubs to/from the C++ NS-3 simulator and other layers that emulate closely the interface to FreeRTOS on the embedded system (nodes). For instance, random number generators are implemented in the application code instead of using the primitives provided by NS-3, as used by the pure NS-3 simulation described in Section 4.2.

Figure 30 shows the file structure of the simulation project. It is based on the same structure and code as the NS-3 simulation described in Section 4.2. The C++ code in file *train-wsn.cc* under directory *scratch*, in C++ files under directory *src/applications/helper*, and in C++ files *train-cc-node.cc* and *train-wsn-node.cc* under directory *src/applications/model* is incrementally converted to C code in the new C files under *src/applications/model*, while additional code is added to implement the new functions (network formation and security) as follows.



```
.
├── Makefile
├── scratch
│   └── train-wsn.cc
└── src
    └── applications
        ├── helper
        │   ├── train-cc-helper.cc
        │   ├── train-cc-helper.h
        │   ├── train-wsn-helper.cc
        │   └── train-wsn-helper.h
        └── model
            ├── train-cc-node.cc
            ├── train-cc-node.h
            ├── train-core-config.c
            ├── train-core-config.h
            ├── train-core-ds.c
            ├── train-core-ds.h
            ├── train-core-encrypt.c
            ├── train-core-encrypt.h
            ├── train-core-mqueue.c
            ├── train-core-mqueue.h
            ├── train-core-netset.c
            ├── train-core-netset.h
            ├── train-core-ns3.h
            ├── train-core-rng.c
            ├── train-core-rng.h
            ├── train-core-rx.c
            ├── train-core-rx.h
            ├── train-core-simulation.c
            ├── train-core-simulation.h
            ├── train-core-tweetnacl.c
            ├── train-core-tweetnacl.h
            ├── train-core-tx.c
            ├── train-core-tx.h
            ├── train-core.c
            ├── train-core.h
            ├── train-globals.cc
            ├── train-globals.h
            ├── train-util.cc
            ├── train-util.h
            ├── train-wsn-node.cc
            └── train-wsn-node.h
```

Figure 30: Structure of C simulation project to implement and validate embedded WSN application code

File *train-core-config.c* under *src/applications/model* implements configuration functions for various parameter of the node.

File *train-core-ds.c* under *src/applications/model* implements a draft of the interface with the distance sensor and related data processing.

File *train-core-encrypt.c* under *src/applications/model* implements the interface functions with the encryption/decryption library.

File *train-core-mqueue.c* under *src/applications/model* implements the message queue of the node, which is used for both TX and RX activities, queue management functions and packet-specific comparison functions.

File *train-core-netset.c* under *src/applications/model* implements various functions related to node discovery and network formation.

File *train-core-ns3.c* under *src/applications/model* implements a thin layer of stub functions for the bidirectional interface between NS-3 C++ code and application C code. These functions will be replaced with OS-specific functions when the application code will be ported on FreeRTOS on the nodes.

File *train-core-rng.c* under *src/applications/model* implements the pseudo-random generators needed by various node functions.

File *train-core-rx.c* under *src/applications/model* implements all functions that deal with the contents of the received packets.

File *train-core-simulation.c* under *src/applications/model* implements some helpers used during the simulation of the code, which will be removed from the embedded version (e.g., log print helpers).

File *train-core-tweetnacl.c* under *src/applications/model* is a library of functions for key generation, encryption, decryption and verification of packet payloads.

File *train-core-tx.c* under *src/applications/model* implements all functions related to RF transmission of packets.

File *train-core.c* under *src/applications/model* implements the core operation of the WSN node application code in the form of a finite state machine (FSM) that encodes the actions to be taken based on the current state and internal or external events, as well as transitions to next state based on the outcome of node operations, as well as the main data structure used by the node.

File *train-core-globals.cc* under *src/applications/model* defines some global variables needed for the NS-3 part of the simulation code.

File *train-core-util.cc* under *src/applications/model* implements some utility functions needed for the NS-3 part of the simulation code.

Network discovery requires several activities to run concurrently, for instance the discovery of the peer nodes across train couplings is performed in parallel to sending CC feedback on already discovered nodes and exchanging public and session (symmetric) keys between nodes and CC.

Simulation has shown that there is a high likelihood of channel congestion and packet collision which requires implementation of adequate handling at application level. We are also exploring the implementation of TDMA slotting, at a suitable level (e.g., separate in time peer discovery across coupling from encryption key exchanges).

5. ANTENNAS DESIGN

In many application scenarios, where electromagnetic propagation is subject to strong reflections, the selection of a long-range standard is the key factor. The second fundamental element is the power consumption, since sensors and actuators are nowadays supplied by small batteries, renewable sources and cutting-edge technologies like energy-harvesting system which provides limited store of energy. In view of this, an accurate research on long-range and low-power standards has been performed to find a possible candidate for the application of interest.

5.1 SUB-GHZ STANDARDS ANALYSIS

- LR-WPAN, 6LoWPAN, 802.15.4, IEEE P802.1ah, Bluetooth/LE, ZigBee, Thread, Wireless Hart ISA100 fall in the low-power local area networks with a less than 1km range thus are not considered for this evaluation [[7], [8]].
- Cellular solutions (e.g. GSM/3G/4G/5G) are not intended for this application since operating in licensed frequencies (requiring the involvement of a mobile operator) [9].
- Low-Power Wide-Area Network (LPWAN) technologies [[10], [11], [12], [13], [14], [15]], summarized in Table 9 and discussed here:

    - **SigFox** is a long-range and low-power standard operating at 868/915 MHz with a very narrow bandwidth (100 Hz). This standard provides a coverage of 3 to 10 km in urban environment and from 30 to 50 km in suburban area. However, SigFox suffers of two main limitations. First the rate is very low and not flexible (100bps), second the transmit time is restricted as defined by ETSI regulation [16]. These two factors exclude this standard for the intended application.
    - **LoRa** is a more flexible low-power and long-range standard that operates at 160-433/868/915 MHz with a coverage of 2-5 km in urban environment and up to 15 km in suburban area (someone tested 20 km with directional antennas, but this statement need further verification) [17]. The achievable rate, which span from 0.25 to 50 kbps, strictly depends on the link distance. Practically speaking, the higher the distance the lower the rate. LoRa differs from SigFox from the fact that the former has some available channels not limited in transmit duty cycle and achieves higher data rates.
    - **DASH7** born for wireless sensors and actuators with a protocol stack able to operate at 433/868/915 MHz This standard provides a communication range up to 2 km and data rate up to 167 kbps.
    - **Weightless** is a family of three open standards for low-power wireless communication in public or private networks for Internet of Things (IoT) end devices with limited throughput and relaxed latency. Although it can operate in any frequency band, it is currently defined for operation in license-exempt sub-GHz frequency bands (e.g. 138 MHz, 433 MHz, 470 MHz, 780 MHz, 868 MHz, 915 MHz, 923 MHz). Rates are very flexible, for example, Weightless W can reach up to 1 Mbps, however they can reach coverage ranges up to 2 km with power consumption higher than other standards (e.g. LoRa).

Table 8. This table summarize a list of low-power standards suitable for this project.

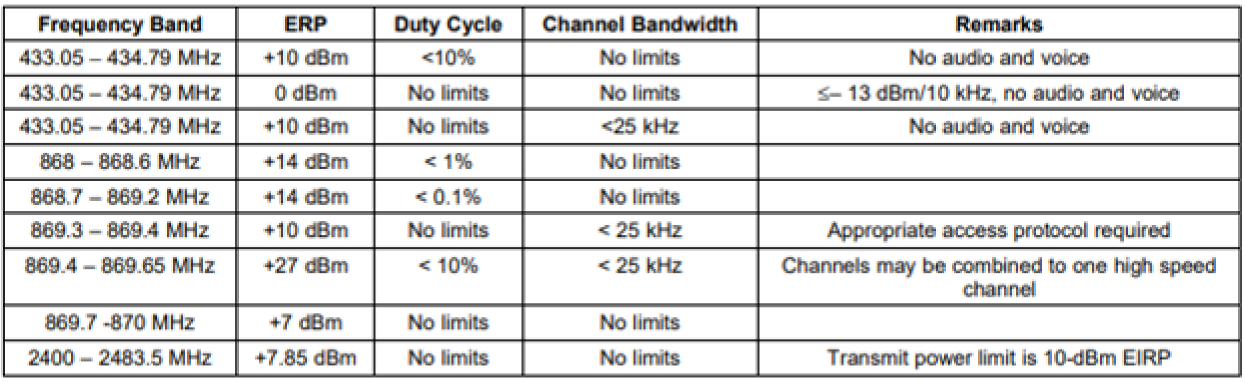
Standard	Op. Freq [MHz]	Range [km]	Data Rate [kbps]	Consumption	Limitation
Lora	160/ 433/ 868/ 915	3 - 10 urban 30 - 50 suburban	0.25 -50 Self-adaptive depending on distance	Very low-power	Transmit Duty Cycle, regulated by ETSI
SigFox	868/ 915	3 - 10 urban 30 - 50 suburban	0.1	Very low-power	Number of transmitted packets Transmit Duty Cycle, regulated by ETSI
DASH7	433/ 868/ 915	2	167	Very low-power	Range Transmit Duty Cycle, regulated by ETSI
Weightless	138/ 433/ 470/ 780/ 868/ 915/ 923	2	Flexible rate up to 1Mbps	High consumption with respect other standards	Transmit Duty Cycle, regulated by ETSI

According to this analysis, LoRa could be the best candidate for this specific application. In the following, LoRa will be deeply analysed showing some of its characteristics.

5.1.1 LoRa Standard

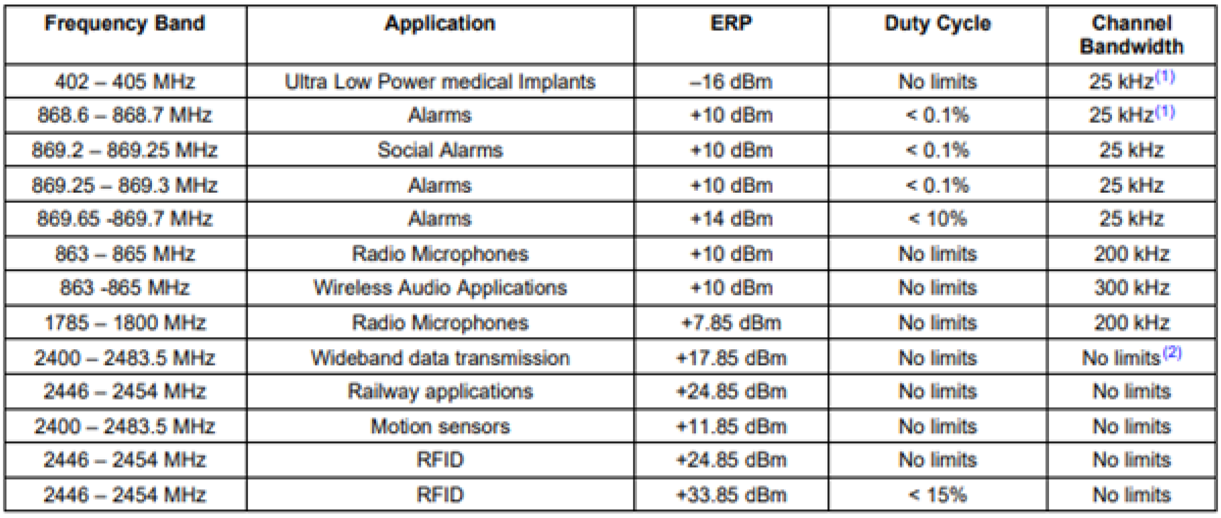
LoRa devices operate at 433/868/915 MHz bands, thus we report Table 9 and Table 10 to illustrate the European regulation under the bandwidth of interest [18].

Table 9: Maximum radiated power limit, ERP and Duty cycle for Non-Specific Short-Range Devices in Europe



Frequency Band	ERP	Duty Cycle	Channel Bandwidth	Remarks
433.05 – 434.79 MHz	+10 dBm	<10%	No limits	No audio and voice
433.05 – 434.79 MHz	0 dBm	No limits	No limits	≤– 13 dBm/10 kHz, no audio and voice
433.05 – 434.79 MHz	+10 dBm	No limits	<25 kHz	No audio and voice
868 – 868.6 MHz	+14 dBm	< 1%	No limits	
868.7 – 869.2 MHz	+14 dBm	< 0.1%	No limits	
869.3 – 869.4 MHz	+10 dBm	No limits	< 25 kHz	Appropriate access protocol required
869.4 – 869.65 MHz	+27 dBm	< 10%	< 25 kHz	Channels may be combined to one high speed channel
869.7 -870 MHz	+7 dBm	No limits	No limits	
2400 – 2483.5 MHz	+7.85 dBm	No limits	No limits	Transmit power limit is 10-dBm EIRP

Table 10: Maximum power limit, ERP and Duty cycle for License-Free Specific Application in Europe

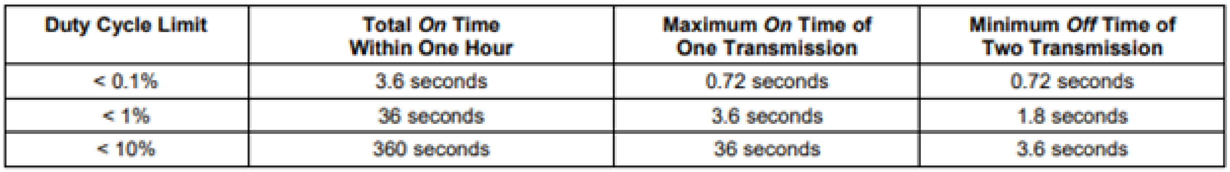


Frequency Band	Application	ERP	Duty Cycle	Channel Bandwidth
402 – 405 MHz	Ultra Low Power medical Implants	–16 dBm	No limits	25 kHz(1)
868.6 – 868.7 MHz	Alarms	+10 dBm	< 0.1%	25 kHz(1)
869.2 – 869.25 MHz	Social Alarms	+10 dBm	< 0.1%	25 kHz
869.25 – 869.3 MHz	Alarms	+10 dBm	< 0.1%	25 kHz
869.65 -869.7 MHz	Alarms	+14 dBm	< 10%	25 kHz
863 – 865 MHz	Radio Microphones	+10 dBm	No limits	200 kHz
863 -865 MHz	Wireless Audio Applications	+10 dBm	No limits	300 kHz
1785 – 1800 MHz	Radio Microphones	+7.85 dBm	No limits	200 kHz
2400 – 2483.5 MHz	Wideband data transmission	+17.85 dBm	No limits	No limits(2)
2446 – 2454 MHz	Railway applications	+24.85 dBm	No limits	No limits
2400 – 2483.5 MHz	Motion sensors	+11.85 dBm	No limits	No limits
2446 – 2454 MHz	RFID	+24.85 dBm	No limits	No limits
2446 – 2454 MHz	RFID	+33.85 dBm	< 15%	No limits

The idea behind this application list is to protect some frequency bands for dedicated purposes. Alternatively, any application can be defined as a non-specific short-range device. For example, an alarm system can be built using either a dedicated frequency band for alarms or a band allocated for non-specific short-range devices. Using the dedicated band has the advantage that no other applications are allowed to use that band, which decreases the probability of interference. Using a general-purpose band might have the advantage of more readily available components, but it would have to consider that other applications may use this frequency band, resulting in a higher probability of interference.

The ERP is related to the EIRP by the relation ERP = EIRP – 2.15 dB. For both Table 9 and Table 10, the duty cycle is defined as the maximum total transmitter on time as a fraction, expressed as a percentage, of the total time in a one-hour period. Additionally, the duration of one individual transmission and the minimum off time between two consecutive transmissions are limited, as detailed in Table 11.

Table 11: Transmit Duty Cycle Limits according to [16]



Duty Cycle Limit	Total *On* Time Within One Hour	Maximum *On* Time of One Transmission	Minimum *Off* Time of Two Transmission
< 0.1%	3.6 seconds	0.72 seconds	0.72 seconds
< 1%	36 seconds	3.6 seconds	1.8 seconds
< 10%	360 seconds	36 seconds	3.6 seconds

It needs to be noticed that, when transmit duty cycle is reached in one channel, the same radio can use another channel to continue with the transmission of data. Since the transmit duty cycle limitation can be critical for the nodes placed at the farthest distances, we found the best matching in Table 9 for bandwidths:

- 433.05 - 434.79 MHz, transmitting with an ERP = 0 dBm (EIRP = 2.15dBm)
- 869.7 – 870 MHz, transmitting with an ERP = 7 dBm (EIRP = 9.15dBm)

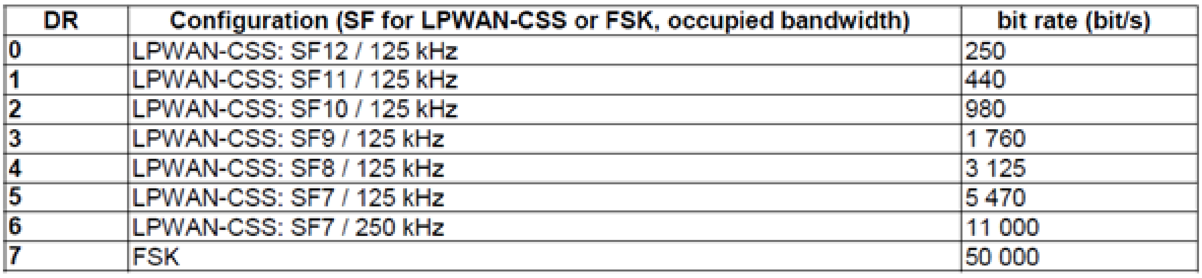
LoRa 433 MHz has only 3 channels, 433.175 MHz, 433.375 MHz and 433.575 MHz, while 868 MHz has more channels (e.g. flexible from 863 MHz to 870 MHz with 0.2 MHz steps), however device producers suggest the following list of default channels to avoid interference problems:

CH_10_868	865.20 MHz
CH_11_868	865.50 MHz
CH_12_868	865.80 MHz
CH_13_868	866.10 MHz
CH_14_868	866.40 MHz
CH_15_868	866.70 MHz
CH_16_868	867.00 MHz
CH_17_868	868.10 MHz

5.1.2 LoRa Rates

LoRa is based on Spread Spectrum technology and has an adaptive rate regulator. Possible rates are reported in Table 12.

Table 12: Data rates allowed by the LoRa WAN protocol and the related Spreading Factor (SF)



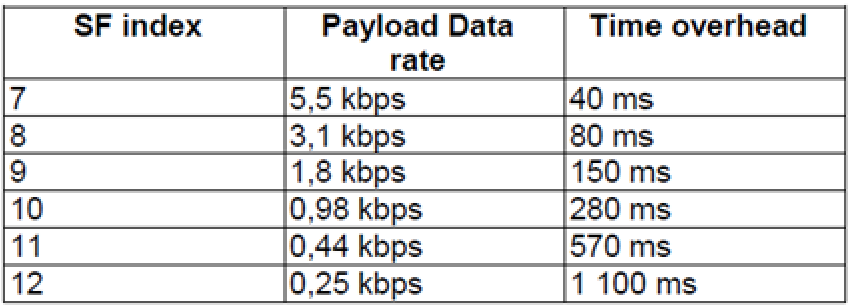
DR	Configuration (SF for LPWAN-CSS or FSK, occupied bandwidth)	bit rate (bit/s)
0	LPWAN-CSS: SF12 / 125 kHz	250
1	LPWAN-CSS: SF11 / 125 kHz	440
2	LPWAN-CSS: SF10 / 125 kHz	980
3	LPWAN-CSS: SF9 / 125 kHz	1 760
4	LPWAN-CSS: SF8 / 125 kHz	3 125
5	LPWAN-CSS: SF7 / 125 kHz	5 470
6	LPWAN-CSS: SF7 / 250 kHz	11 000
7	FSK	50 000

An end node tries and estimate the highest data rate i.e. the lower SF it can use and be received correctly by the Network Server. Starting with that estimation (which, in any case, can be the highest data rate i.e. the lower SF) it initiates the transmissions. If no reply is received within the next expected downlink transmissions, the end node may try to establish connectivity by switching to the next lower data rate (i.e. the next higher SF) that provides a more robust connectivity. The end node will further lower its data rate (i.e. increase the SF) step by step until the communication with Network Server is established. Even if, from the protocol point of view, the gateway could seem a simple machine, it is a quite complex one from the signal processing point of view. As a matter of fact, the receiver in the gateway is a highly configurable parallel machine acting like (at least 8) different independent receivers, configurable on different channel frequencies (each of them can demodulate any SF index from 7 to 12 in parallel) [19].

5.1.3 LoRa Packets

The LPWAN-CSS packet of data consists of a preamble, a PHY (Physical Layer) header and an actual payload. The preamble is used for detection and synchronization purposes, the PHY header describes the payload length which ranges from 13 to 255 bytes. Indeed, the minimum size of a LoRa WAN physical payload is 13 bytes. A 16 bits CRC is also transmitted. In Table 13 the time overhead (i.e. the time used for the transmission of the preamble, the PHY header and the CRC) and the payload data rate are shown as function of the spreading factor.

Table 13: Payload Data Rate and Time overhead per packet, excluding data as a function of the SF index



SF index	Payload Data rate	Time overhead
7	5,5 kbps	40 ms
8	3,1 kbps	80 ms
9	1,8 kbps	150 ms
10	0,98 kbps	280 ms
11	0,44 kbps	570 ms
12	0,25 kbps	1 100 ms

5.1.4 Example of Implementation

Figure 31 illustrates a possible implementation of a LoRa network based. The nodes are currently distributed in a circle with a maximum distance of 15 km with respect the gateway. Each node (after rates negotiation) will use a rate that depends on the position it is placed. For example, the farthest device will employ the minimum rate (250 bps / SF = 12). Each node communicates with the Gateway in a bidirectional manner.

An important feature of a LoRa networks is that, every node can communicate with others as long as they operate on the same frequency channel.

Transmit Time Analysis

LoRa network need to be properly designed, especially in bandwidth with transmit duty cycle imposed. In the following, the worst/best case scenario to transmit a LoRa packet is reported.

$$LoRa\ Maximum\ Payload\ = 255 Bytes = 2048 bits$$

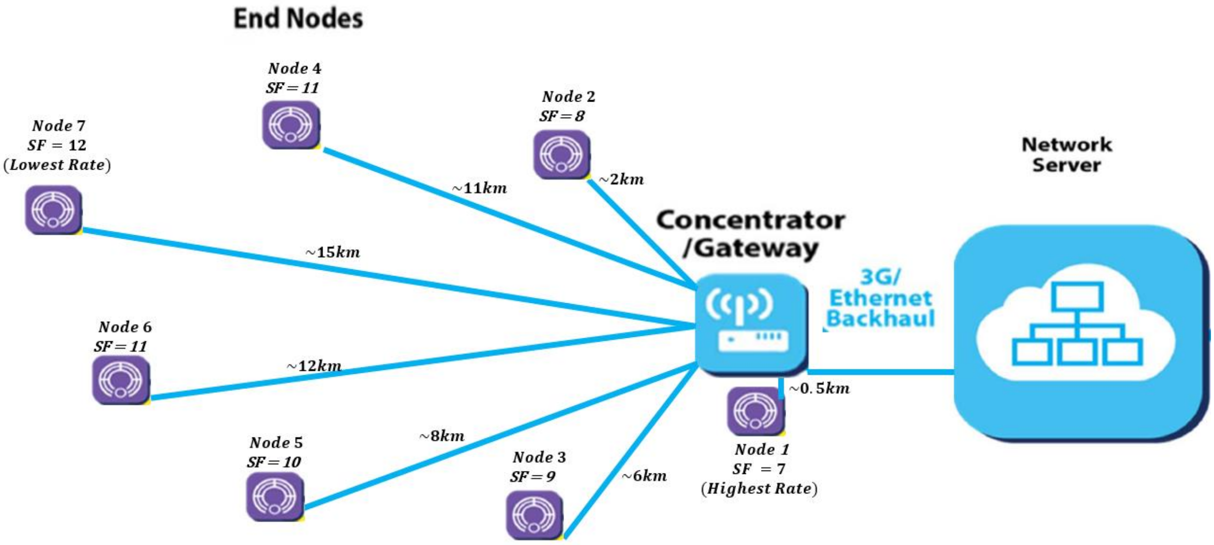


Figure 31: Example of a possible architecture of a LoRa network

For the node 12 in Figure 31, that is placed at 15Km we suppose that transmit at minimum rate, thus:

$$Rate\ (SF = 12) = 250bps$$

$$TX_{time}\ (1\ packet, SF = 12) = \frac{LoRa\ Maximum\ Payload}{Rate\ (SF = 12)} + Time\ overhead\ (SF = 12) = 9{,}3s$$

Where the maximum payload is:

$$LoRa\ Maximum\ Payload\ \ = 255Bytes = 2048bits$$

While in the best case (e.g. node 1), this node exploits the highest rate (@125KHz bandwidth):

$$Rate\ (SF = 7) = 5{,}5Kbps$$

$$TX_{time}\ (1\ packet, SF = 7) = \frac{LoRa\ Maximum\ Payload}{Rate\ (SF = 7)} + Time\ overhead\ (SF = 7) = 0{,}42s$$

In conclusion, the design of a LoRa network strictly depends from two factors:

- Link distance of the farthest node
- Quantity of data to be transmitted

5.1.5 Link Budget Analysis

To perform the link budget calculation we account for the characteristics of the commercial device Adafruit RFM - Low Power Long Range LoRa Transceiver Module, based on Semtech Radio SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver [[19], [20], [21], [22]]. The first parameter of interest is the receiver sensitivity, a function of the data rate. Here, we report an extract from the radio datasheet.

Table 14: LoRa Receiver Sensitivity as a function of the used data rate (Spreading Factor)

Description	Conditions	Typical	Unit
RF sensitivity, Long-Range Mode, highest LNA gain, *LnaBoost* for Band 1, using split Rx/Tx path 125 kHz bandwidth	SF = 6	-118	dBm
	SF = 7	-123	dBm
	SF = 8	-126	dBm
	SF = 9	-129	dBm
	SF = 10	-132	dBm
	SF = 11	-133	dBm
	SF = 12	-136	dBm

$$LoRa\ RX_{Sensitivity}(SF = 12) = -136\text{dBm (Typical)}$$

$$LoRa\ RX_{Sensitivity}(SF = 7) = -123\text{dBm (Typical)}$$

At 868MHz, we define the $EIRP = \ 9dBm$ which is allowed on each sub-band and we describe the propagation as

$$RSSI\ (dBm) = \ EIRP - \ 20Log10\left(\frac{4\ \pi\ d}{\lambda}\right) + G_{RX}$$

Where RSSI is the Receive Signal Strength indication in dBm, G_{RX} is the gain of the antenna at receiver side and is set to $5dBi$, while λ is the wavelength $69{,}3cm$.

Theoretically, the maximum achievable link (with +20dBm of link margin) is

- About 3,5 km with the rate 5.5Kbps (SF = 7)
- 16,5 km with the lowest rate 250bps (SF=12)

As shown in the link budget calculation reported in Figure 32.

The same computation can be performed at 433MHz, with the exception that the EIRP is set to 2.15dBm, thus G_{RX} will be 0dBm. In this case:

Theoretically, the maximum achievable link (with +20dBm of link margin) is

- About 8 Km with the rate 5.5Kbps (SF = 7)
- 35 Km with the lowest rate 250bps (SF=12)

As shown in the simulation reported in Figure 33.

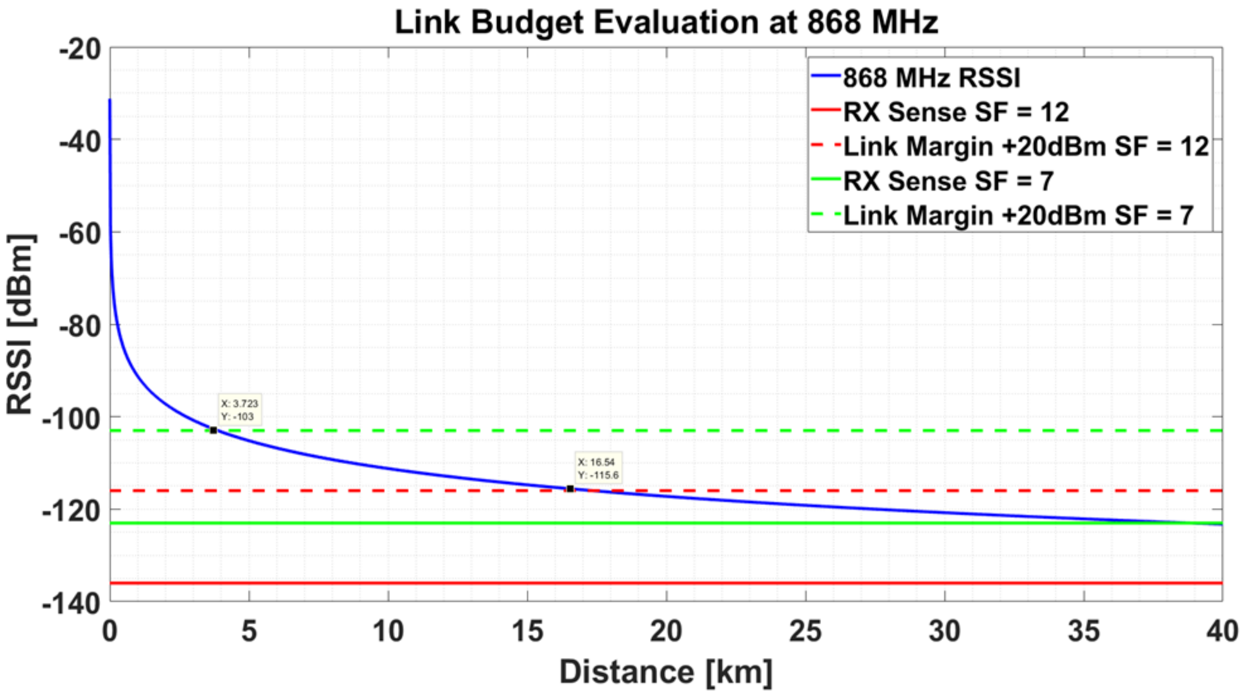


Figure 32: Evaluation of the Link Budget at 868 MHz: the maximum distance achievable with a specific data rate is sampled at +20dBm with respect the receiver sensitivity for the selected rate.

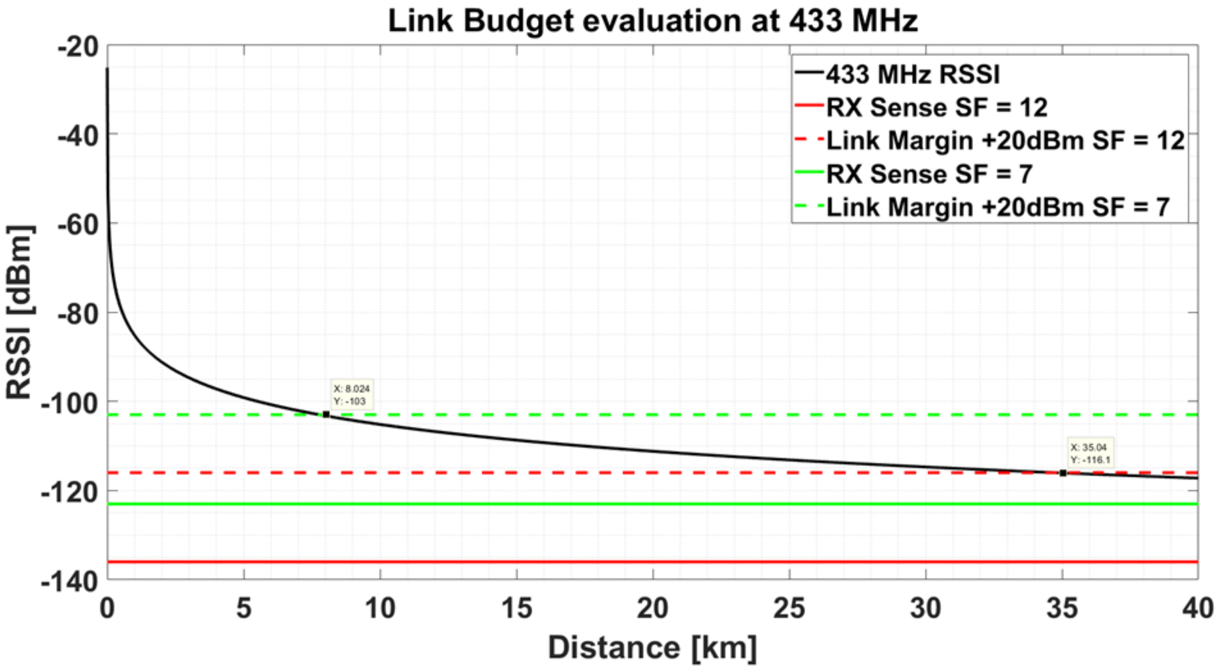


Figure 33: Evaluation of the Link Budget at 433 MHz: the maximum distance achievable with a specific data rate is sampled at +20dBm with respect the receiver sensitivity for the selected rate

5.2 Design of Transmission and Receiver Antennas

The concept of directive antenna for energy saving and/or interference reduction aims at maximizing radiation in the intended direction, while reducing the energy wasted in unwanted directions.
The employment of directive antenna on wireless devices has two complementary advantages over one with a standard (omnidirectional) antenna:

1. at the same link distance, we have the same performance with a lower transmit power. Since power consumption of the wireless device is proportional to the transmitted power, then a power saving is achieved;
2. with the same transmit power, one has a higher received power, which allows to transmit data faster. This implies that a given quantity of information can be transferred in a shorter time, i.e. with a smaller energy.

Both result in a lower energy-per-bit cost; of course, the two advantages can be combined, and/or the distance can be increased with all the rest unchanged.
Directive antennas should be exploited on an end-to-end communication to maximize the quality of the link. The minimization of the radio energy consumption is a direct consequence. With the sole purpose of giving an example, Figure 34 shows an experiment with the emissions of LoRa packets for different transmit powers (i.e. same packet length and data rate). Of course, the absorbed current increases as the transmit power increases (e.g. +5 dBm, +13 dBm, +20 dBm).

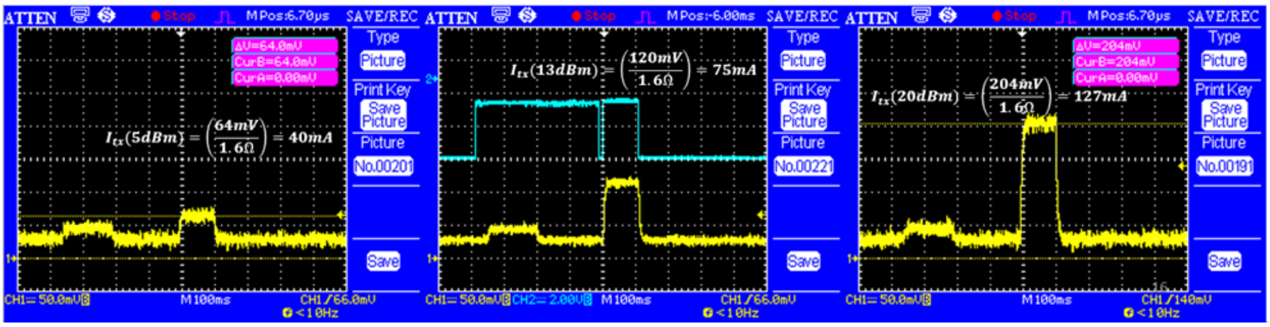


Figure 34: Absorbed current when transmitting a LoRa packet with +5dBm, +13dBm, +20dBm, from left to right respectively

Thanks to the higher antenna gain provided by directive antennas, radio nodes can reduce the transmit power expecting a power saving.
Instead in Figure 34, a node placed at longer distances will found a better signal quality and thus exploiting the adaptive rate regulator present in the LoRa protocol will find the best rate that can be used (less time to transmit data, less energy).

5.2.1 Antennas for Sub-GHz Communication

In Europe, the two permissible license exempt frequency bands are at 433MHz and 868MHz. The conventional view of radio range is that range is proportional to wavelength (inversely proportional to frequency), which would suggest that 433MHz would provide better range (for the same radio technology) than 868MHz. However, a simple ¼ monopole antenna is too long to be robust at 433MHz, forcing adoption of a compact helical antenna. Compact helical antennas are quite narrowband, and can be easily detuned by the presence of metal within (approximately) ¼ of a wavelength (~17cm), which is typically the case on a bogie. For this reason, 868MHz may be the preferable wavelength, with a ¼ wave monopole antenna providing more efficient transmission in a mechanically crowded environment.

Wire antennas, like dipoles and monopoles are some of the oldest, simplest to realize and cheapest on the market. The preferred form is usually a $\frac{\lambda}{4}$ monopoles, showed in Figure 35a, especially when constraints on size apply. This antenna is usually connected to a 50Ω coaxial cable to be compatible with almost all the radio devices. Radiation is of the form of a donut (i.e. omnidirectional in the horizontal plane), showed in Figure 35b, which is favourable indoor (e.g. router Wi-Fi) in order spread the signal in any direction of a house. Unfortunately, for minimum power consumption aid and/or maximum achievable link coverage, dipole radiation is not efficient (i.e. energy wasted in undesired directions).

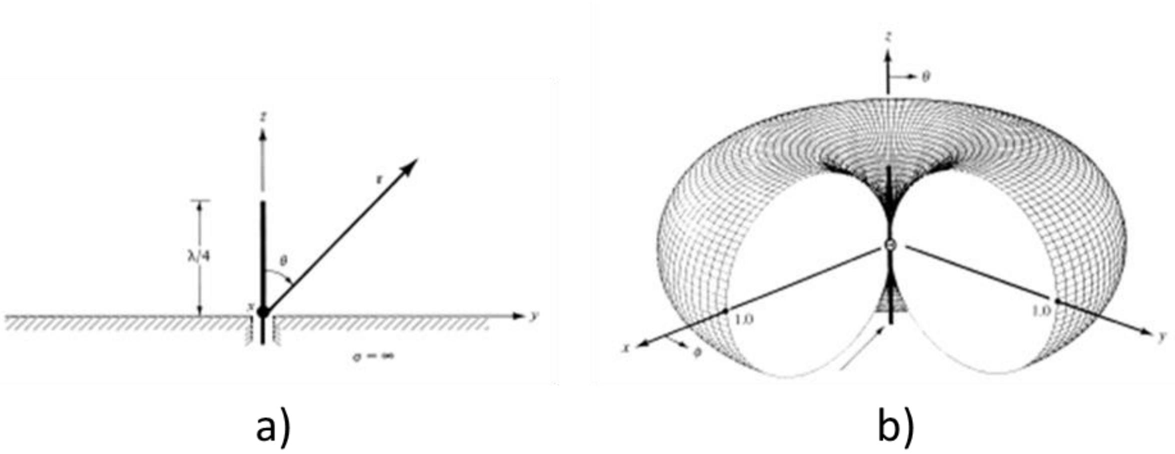


Figure 35: a) Geometrical representation of a $\frac{\lambda}{4}$ monopole antennas; b) the radiation pattern (3D) of the monopole antenna

Axial mode Helical antenna is a conducting wire wound in the form of a screw thread forming a helix above a ground plane, as shown in Figure 36a. An axial mode Helical antenna has more complex geometry but provides superior properties with respect wired antennas. First of all, helical antennas exhibit greater broadband characteristics than those of the dipoles. Second, the radiation is spread along the axis of the helix, and it is similar to that of an end-fire array, thus providing a high directivity in one direction (no loss of energy), as showed in Figure 36b.

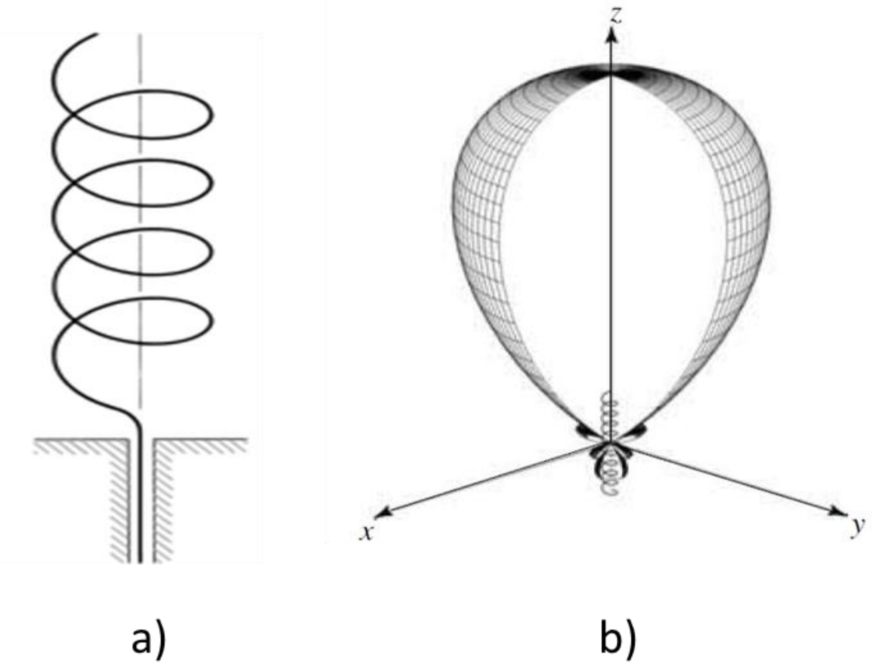


Figure 36: a) Geometrical representation of a standard helix antenna; b) the radiation pattern (3D) of an axial mode helix antenna

The radiation characteristics of the antenna can be varied by controlling the size of its geometrical properties compared to the wavelength. In particular, the number of turns defines the directivity of the helix antenna, see Figure 37.

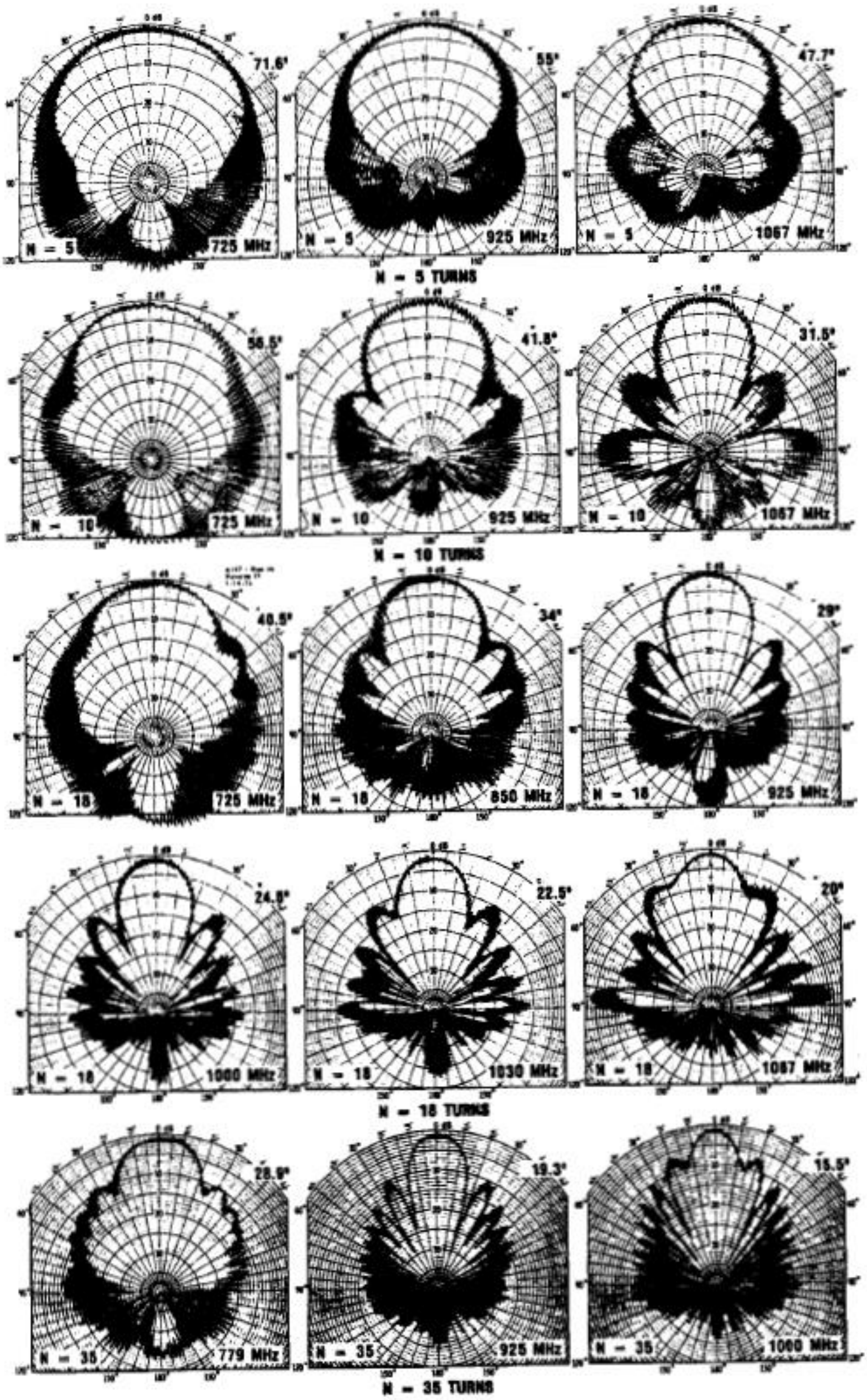


Figure 37: A series of radiation pattern characteristics form a standard axial mode Helix antenna by increasing the number of coil turns

The input impedance is critically dependent upon the variation angle of the wire (i.e. pitch angle) and the size of the conducting wire, especially near the feed point, and it can be adjusted by controlling their values. The general polarization of the antenna is elliptical. However circular and linear polarizations can be achieved over different frequency ranges.

5.2.2 Antennas for UWB Communication

The tapered slot antenna is the most popular directive antenna for UWB applications due to its simple structure and small size. This antenna is composed by a flared slot line etched on a dielectric substrate to produce a directive pattern from a surface wave. A single element radiates this directive pattern over a wide bandwidth.

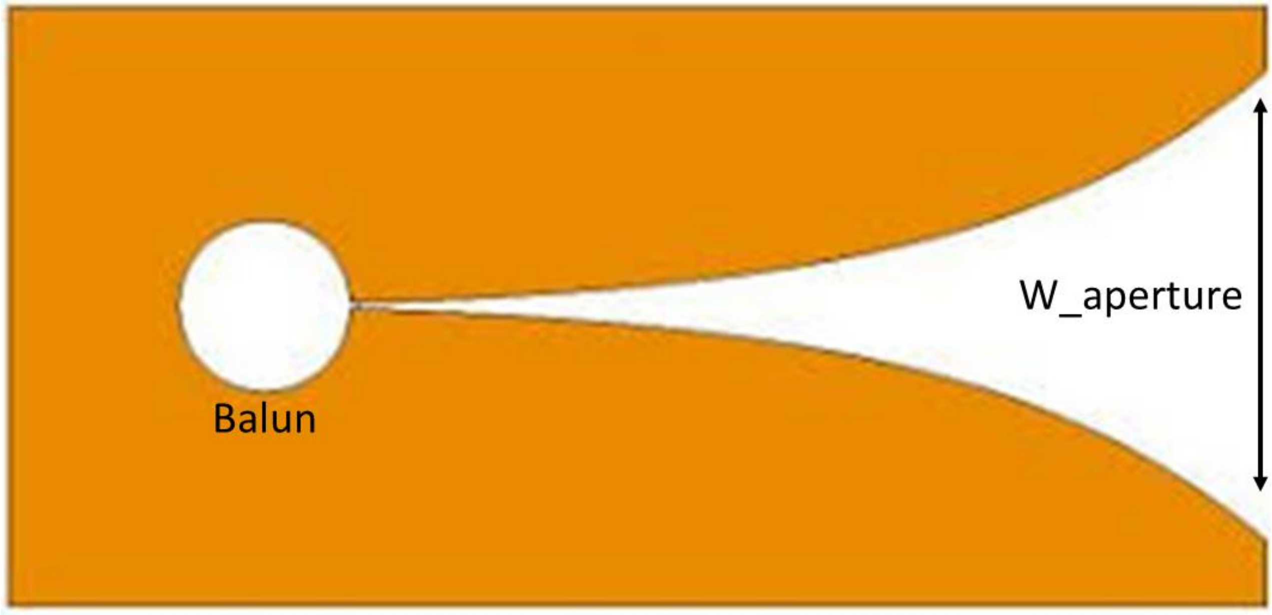


Figure 38: An example of Wideband tapered slot antenna

An exponentially tapered slot antenna, also called a Vivaldi antenna in Figure 38, radiates nearly equal E- and H-plane beamwidths that change only slightly as frequency increases, thus ideal for UWB applications. The beamwidth of the radiated pattern is related to the final aperture of the tapered slot. In other words, the higher the aperture (size) the higher the directivity. This concept can also be summarized in Table 15, which reports the relationship between the beamwidth and the final aperture of the slot in term of wavelength.

Table 15: Expected beamwidth radiation based on the slot aperture

W aperture in terms of (λ)	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6
Beamwidths (°)	50	42	38	33	31	30	30	30	32	35

To give an idea, Figure 39 shows an example of radiated field from the tapered slot antenna.

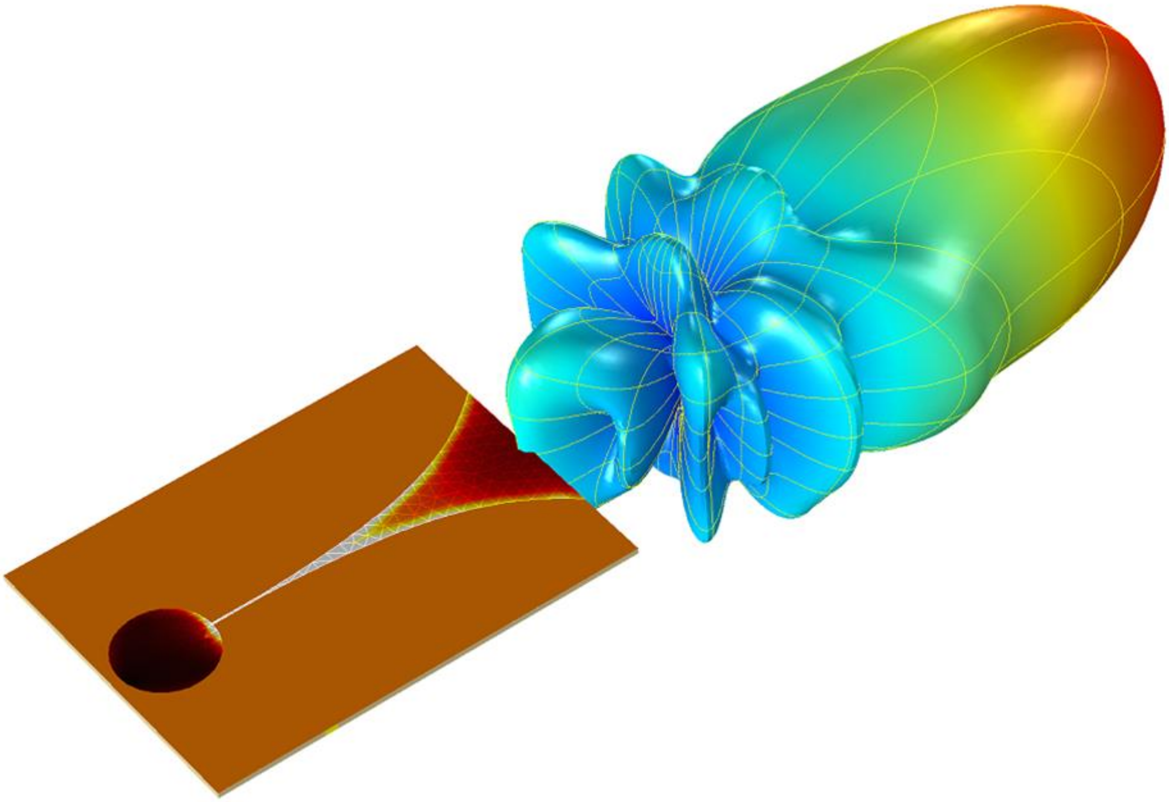


Figure 39: An example of radiated pattern from the tapered slot antenna

A balun (impedance matching circuit) is added on the back side of the antenna to match the desired input impedance, usually 50Ω for standard radio. Total length of the antenna is about 1.5 times the wavelength (1.5 λ) considering also the matching balun. For a 6GHz operation, the size of this planar antenna with an aperture of (λ) would be L=75mm, W=50mm.

6. APPENDIX

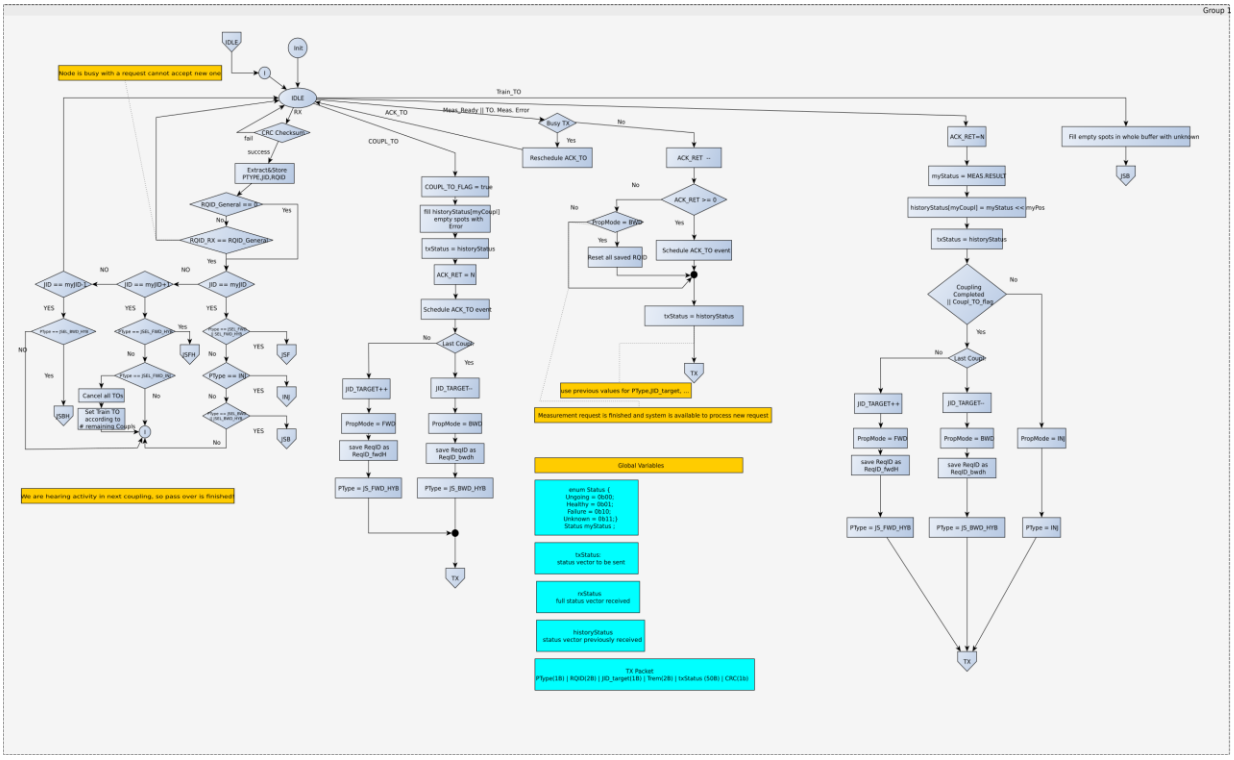


Figure 40: Sensor node state and processing diagram (part 1)

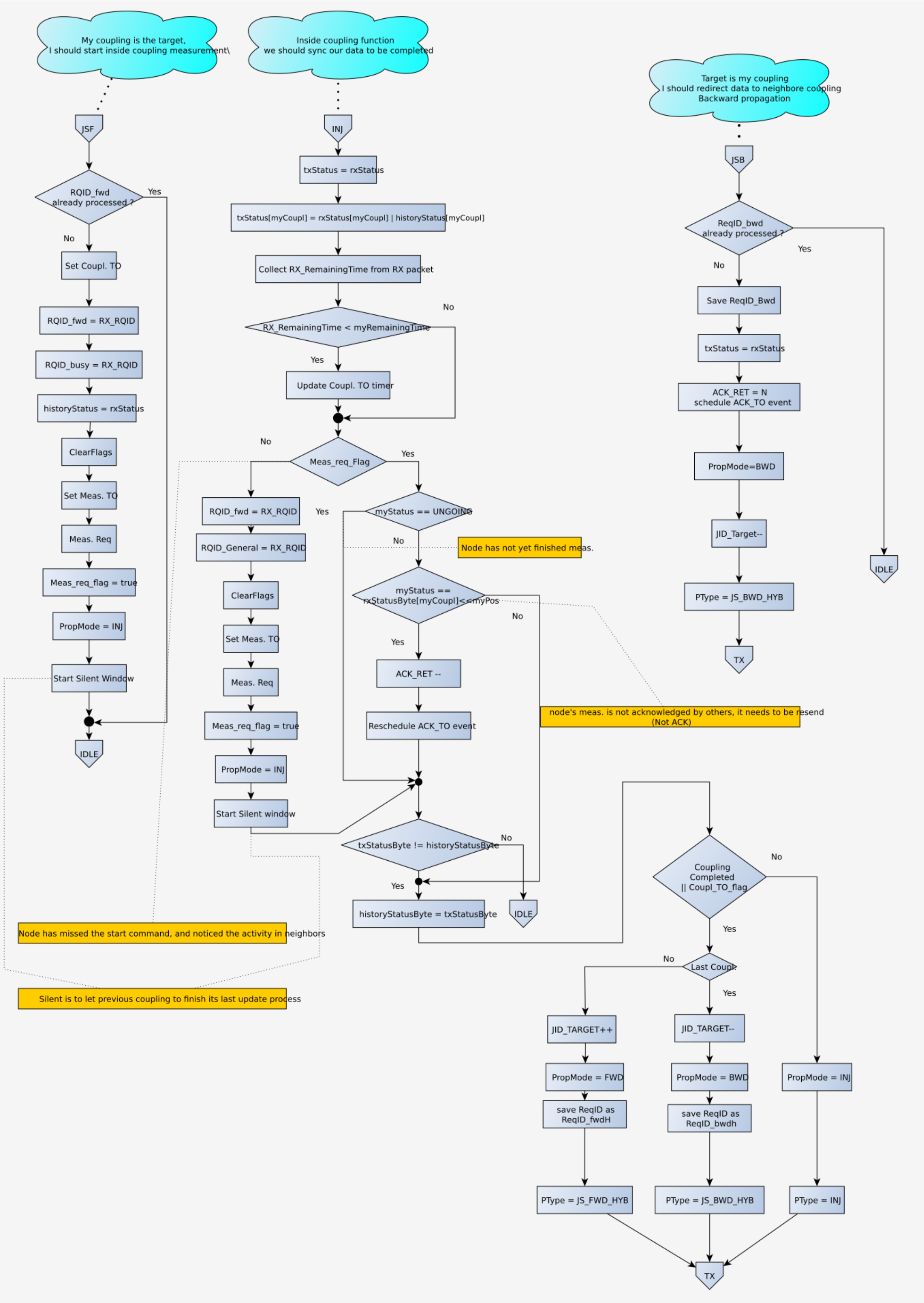


Figure 41: Sensor node state and processing diagram (part 2)

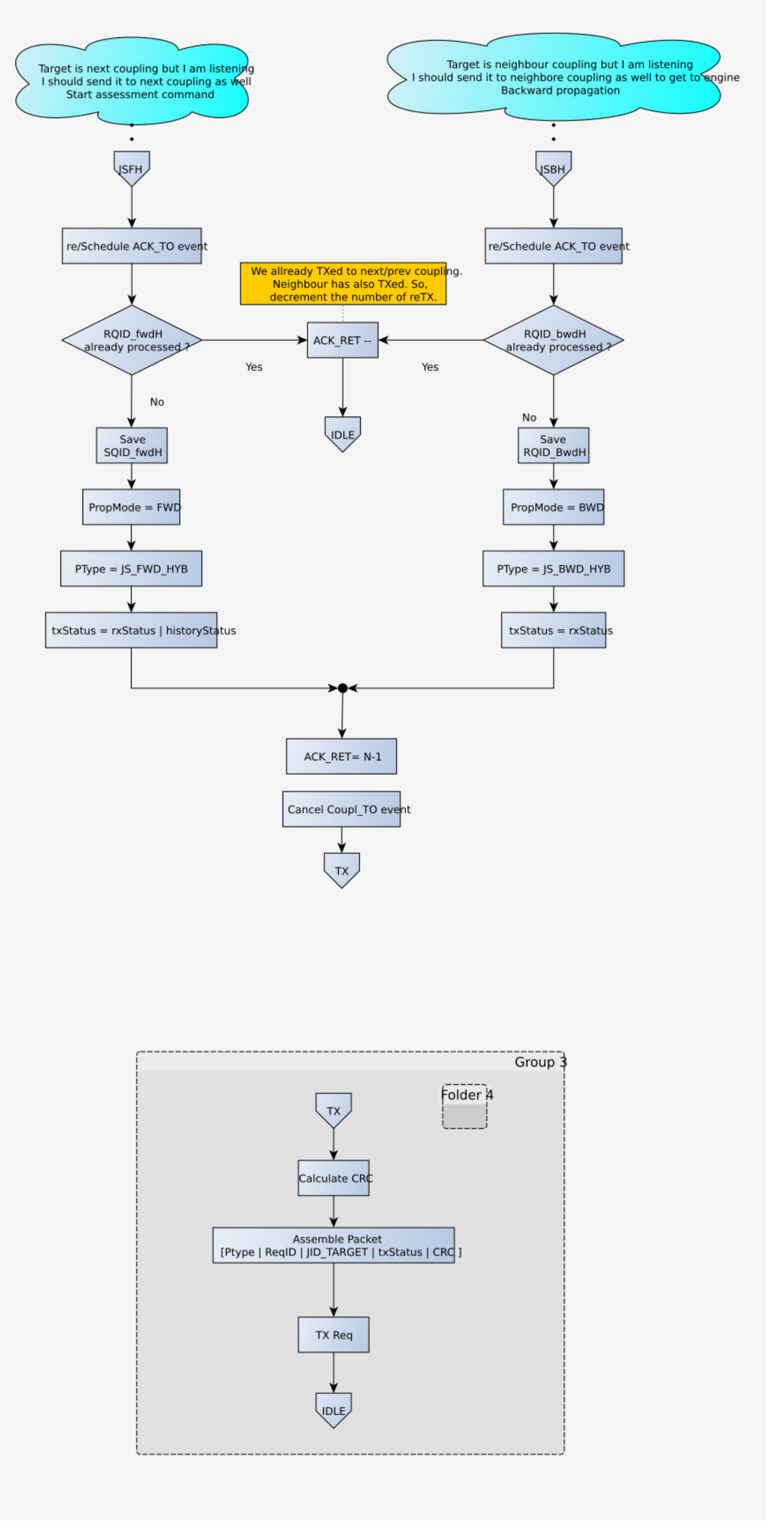


Figure 42: Sensor node state and processing diagram (part 3)

7. CONCLUSIONS

In this document it is has been outlined the current proposal for monitoring TI over a WSN with a custom light-weighted communication protocol.

Architectural decisions have been taken considering the Requirements outlined in D2.2 “System Requirements Specification”, giving high importance to the solutions robustness for reaching a SIL4 enabling, power consumption for running the solution with Energy Harvesting and software simplicity.

The usage of WSN for TI monitoring has been studied in various manners, as reported in D3.1 “Trade-off Analysis for On-board and Track-side Communication Systems”, giving good results on the protocol connectivity, but resulting a poor precision of TI.

By employing UWB technology for measurements, it is possible to greatly increase the TI check precision, reaching a granularity around one or maximum two meters; while solutions based on the communication radio link result in a TI check granularity on the tens of meters.

The document also reports simulations of the protocol, confirming the potentiality and the compliancy to the defined Requirements. With the simulation data, other potential improvements are outlined that could benefit the solution, keeping in mind that the protocol must be kept as light weighted as possible,

Security aspects are also considered, presenting a solution that is secured both through encryption (data is secured and only readable/usable by authorized entities) and by localization (the SN must be geometrically aligned, or it will not be accepted as part of the network). This maximizes the network security and prevents malicious attacks.

For correct functionality of the custom solution, also specific design for the antenna has been taken on, by creating special directional beam antenna that ensure the communication with only the wanted nodes and increase the power of the communication (thus meaning an increased communication distance).

In order to check and prove the possibility of running the Sensor Node with an Energy Harvesting solution, a deep power consumption analysis has been performed for each hardware component of the SN. Results show that the energy consumption for a TI check drastically decreases with an increasing TI check interval (TI is checked less frequently). Since in the ETALON scope TI check intervals depend on the train’s speed, TI is checked more frequently when the train’s speed is higher, meaning that also the EH solution will produce more energy.

8. REFERENCES

[2] STEVAL-FKI868V1 Sub-1GHz transceiver development kit based on S2-LP

[3] STM32L152RE MCU

[4] S2-LP Ultra-low power, high performance, sub-1GHz transceiver

[5] Raspberry Pi 3 Model B+

[6] DWM1001 UWB module

[7] H. Wang, «A Survey of Enabling Technologies of Low Power and Long Range Machine-to-Machine Communications», IEEE Communications Surveys & Tutorials, vol. 19, n. 4, pp. 2621 – 2639, 2017

[8] J. Chen, «Transmit energy-efficiency for long-range wireless communications from battery-powered unmanned systems», IEEE Transactions on Aerospace and Electronic Systems, vol. 51, n. 4, pp. 2944 – 2959, 2015

[9] A. Augustin, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things», Sensors, vol. 16, pp. 1466 - 1484, 2016

[10] H. Al-Kashoash, «Comparison of 6LoWPAN and LPWAN for the Internet of Things», Australian Journal of Electrical and Electronics Engineering, vol. 13, n. 4, pp. 268-274, 2017.

[11] K. Yaw-Wen, «Design of a wireless sensor network based IoT platform for wide area and heterogeneous applications», IEEE Sensors Journal, pp. 1 - 1, 2018

[12] M. Centenaro, «Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios», IEEE Wireless Communications, vol. 23, n. 5, pp. 60 - 67, 2016

[13] F. Adelantado, «Understanding the Limits of LoRaWAN», IEEE Communications Magazine, vol. 55, n. 9, pp. 34 – 40, 2017

[14] M. Kim, «A consumer transceiver for long-range IoT communications in emergency environments», IEEE Transactions on Consumer Electronics, vol. 62, n. 3, pp. 226 - 234, 2016

[15] W. San-Um, «A long-range low-power wireless sensor network based on U-LoRa technology for tactical troops tracking systems», in Third Asian Conference on Defence Technology (ACDT), Phuket, Thailand, 2017

[16] R. Kishore, «Radio data infrastructure for remote monitoring system using Lora technology», in International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017

[17] H.-C. Lee, «Monitoring of Large-Area IoT Sensors Using a LoRa Wireless Mesh Network System: Design and Evaluation», IEEE Transactions on Instrumentation and Measurement, pp. 1 – 11, 2018

[18] ETSI, «Final draft ETSI EN 300 220-1 V2.4.1 (2012-01); Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW». Available: http://www.etsi.org/deliver/etsi_en/300200_300299/30022001/02.04.01_40/en_30022001v020401o.pdf. [Online 2018 April 3]

[19] Adafruit, «Adafruit Feather M0 with RFM95 LoRa Radio - 900 MHz - RadioFruit». Available: https://www.adafruit.com/product/3178. [Online 2018 April 14]

[20] CEPT, «ERC Recommendation 70-03: Relating to the use of Short Range Devices (SRD)». Available: https://www.ecodocdb.dk/download/25c41779-cd6e/Rec7003e.pdf. [Online 2018 May 4]

[21] ETSI, «ETSI TR 103 526 V1.1.1 (2018-04) System Reference document (SR docSRdoc); Technical characteristics for Low Power Wide Area Networks. Chirp Spread Spectrum (LPWAN-CSS) operating in the UHF spectrum below 1 GHz». Available: http://www.etsi.org/deliver/etsi_tr/103500_103599/103526/01.01.01_60/tr_103526v010101p.pdf. [Online 2018 May 12]

[22] Adafruit, «RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0». Available: https://cdn-learn.adafruit.com/assets/assets/000/031/659/original/RFM95_96_97_98W.pdf?1460518717. [Online 2018 May 18]