Asynchronous Resilient Wireless Sensor Network for Train Integrity Monitoring

Asynchronous Resilient Wireless Sensor Network for Train Integrity Monitoring

Mihai T. Lazarescu, *Senior Member, IEEE,* and Pooya Poolad, *Student Member, IEEE*

***Abstract*—To increase railway use efficiency, the European Railway Traffic Management System (ERTMS) Level 3 requires all trains to constantly and reliably self-monitor and report their integrity and track position without infrastructure support. Timely train separation detection is challenging, especially for long freight trains without electrical power on cars. Data fusion of multiple monitoring techniques is currently investigated, including distributed integrity sensing of all train couplings. We propose a Wireless Sensor Network (WSN) topology, communication protocol, application, and sensor nodes prototypes designed for low power timely train integrity reporting in unreliable conditions, like intermittent node operation and network association (e.g., in low environmental energy harvesting conditions) and unreliable radio links. Each train coupling is redundantly monitored by four sensors, which can help to satisfy the Train Collision Avoidance System (TCAS) and European Committee for Electrotechnical Standardization (CENELEC) SIL 4 requirements and contribute to the reliability of the asynchronous network with low rejoin overhead. A control center on the locomotive controls the WSN and receives the reports, helping the integration in railway or Internet of Things (IoT) applications. Software simulations of the embedded application code virtually unchanged show that the energy-optimized configurations check a 50-car train integrity (about 1 km long) in 3.6 s average with 0.1 s standard deviation and that more than 95 % of the reports are delivered successfully with up to one-third of communications or up to 15 % of the nodes failed. We also report qualitative test results for a 20-node network in different experimental conditions.**

***Index Terms*—Wireless sensor networks, Internet of things, media access protocol, communication protocol, asynchronous, low power, reliability, optimization, design space exploration, safety, train integrity, low maintenance**

I. Introduction

OVER more than 2500 years since the first historical record [1], rail transportation evolved to the large capacity, dependable, efficient, cheap, and safe service on which nowadays we rely for both passengers and freight transportation.

Signaling systems are central for rail transportation quality of service (QoS), safety, and efficient infrastructure utilization, which is expensive and of limited capacity. Current systems, such as axle counters or track circuits, can sense track occupancy only at the block (or section) level, which is inefficient and must change, as stated in different high-level documents mentioned in the article. Level 3 of the European Train Control System (ETCS), the signaling component of the European Railway Traffic Management System (ERTMS) standard [2], requires trains to constantly self-monitor their position and integrity to improve the usage efficiency of railway networks. This way, trains can be modeled as moving blocks, reducing the safety distance between them (hence increasing the track usage) depending on their relative speeds and braking distance [3], [4]. Accidental train separations suddenly vary the train length and speed and must be promptly detected and reported, so that adjacent trains can timely adjust their speed for safety.

ETCS Level 3 compliant on-board detection and signaling of accidental freight train separation are especially challenging because they can have tens of cars, be more than one kilometer long [5], and lack electrical power. On-board train composition and integrity check systems are self-powered and may also provide communication and ETCS services between train ends.

Pneumatic train brake fail-safe operation is among the earliest train separation detection techniques. The ensuing loss of pressure automatically applies the train brakes for safety, but with long latencies, over $14\,\mathrm{s}$ for $2\text{-}\mathrm{km}$ trains, exceeding the required $5\,\mathrm{s}$ reporting interval [3], [6]–[8]. Global Navigation Satellite Systems (GNSSs) enhanced with Inertial Measurement Units (IMUs) [9] can be used to sense train length variations, yet their accuracy may be insufficient, especially in low coverage conditions [10]–[13], may be sensitive to normal train length variations, and need long-range wireless communication between the train ends. Distributed integrity sensing using Wireless Sensor Networks (WSNs) locally assess the status of each coupling, provide train end-to-end communication, can monitor other train car parameters (e.g., bearings health, wheel flat spots), and detect separation location.

Integrity assessment techniques alone can hardly satisfy the stringent Train Collision Avoidance System (TCAS), ETCS Level 3, and the European Committee for Electrotechnical Standardization (CENELEC) Software Integrity Level (SIL) 4 specifications. Currently are of interest data fusion techniques from the most promising methods that we discussed above [14]. In this context, we propose a WSN for train integrity assessment and reporting designed to avoid single points of failure and assuming unreliable operation conditions, like intermittent node failure due to, e.g., low environmental energy harvesting, and unreliable radio frequency (RF) connections due to variable propagation conditions. To improve the Quality of Service (QoS), we optimize for reporting latency and energy, the latter lowering the equipment and maintenance cost, e.g., energy harvesting device size or the battery replacement period [15].

M. T. Lazarescu is with the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy (e-mail: mihai.lazarescu@polito.it).

P. Poolad is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada (e-mail: pooya.poolad@isl.utoronto.ca).

II. Related work

On-board integrity detection systems can rely on a device at the train end or use distributed sensing. Beneficiaries prefer simple, maintenance- and cost-effective systems [3]. Wired solutions can be based on the self-configurable IEC 61375 or IEEE 1473-1999 Type T standard, used mainly for passenger train control, diagnostics, and passenger information [16]. A distributed monitoring system proposed for freight trains uses car modules connected with serial buses across couplings, inferring separations and their location from link faults [8], [17].

Wireless communications have many railway uses [18], [19], especially in the ERTMS standard. A broad class uses GNSS to synchronously locate the train ends and infer the train integrity from the distance between them [12]. TCAS requires detection of separations below $30\,\mathrm{m}$ [11], while GNSS should accept length variations from normal train operation, like starting to brake, move, or bend. GNSS can be unreliable [10] in bad signal conditions, such as tunnels, buildings, urban or natural canyons, anti-noise barriers, roofs (including glass roofs), parallel trains, melting snow, electromagnetic noise from locomotives or sites [13]. Differential GNSS can improve accuracy [20], but relies on base station infrastructures not accepted by ETCS Level 3 specifications. Hence, GNSS-based systems should be complemented with systems operating on different principles [3], [11], e.g., on-board sensors [9] like IMUs. GNSS systems can localize both train extremities to help coordinate nearby traffic [21], but cannot locate the separation [8], [17], and need roofs or other high rise elements which may be incompatible with some car types [3].

Other systems use the time of flight of wireless communications or the brake pipes to evaluate the distance between train ends. The latter may sense the pipe pressure at the far end and signal via radio abnormal levels or can send acoustic signals through the pipes which are evaluated on the locomotive. Brake pipe pressure, airflow, or acoustic transmission echoes can be monitored on the locomotive, but such systems can have long reaction times [22], [23] and cannot locate the separation [8], [17].

On-board sensing options to improve GNSS-based system accuracy are discussed in [11]. They report detection latencies of $200\,\mathrm{s}$ when cruising, and $35\,\mathrm{s}$ under acceleration, significantly higher than the $5\,\mathrm{s}$ specifications [6]–[8]. A switching linear dynamic system can improve GNSS-based separation detection accuracy to $\pm 10\,\mathrm{m}$ for a $475\,\mathrm{m}$ train, with a train separation false positive detection rate of $0.09\,\%$ [9]. They send IMU measurements every second to a ground station and, for safety decisions, the actual train separation time can be calculated compensating the detection delay. Ground station communications may be delayed, with no upper bound, [24], and are not accepted by ETCS Level 3 specifications. Compensating ionosphere, troposphere, clock deviations, and using a railway track database can convert train positioning from a 3D to a 1D problem [20] and lower GNSS errors to meet the CENELEC SIL 4 requirements of hazard rates below $10^{-9}\,\mathrm{h}^{-1}$. Using a dual GNSS system based on different satellite constellations is also beneficial [21], but GNSS methods can fail in areas without good coverage.

In general, GNSS-based methods cannot determine train composition and can be slow detecting train separations because, e.g., of GNSS limitations, train length variations during normal operation, and track topology. They need a long-range reliable wireless connection between the train ends, and some car types are incompatible with a GNSS antenna installation [3].

Distributed WSN sensing can also be used for train integrity detection. WSNs can be optimized for many applications [25], monitor for events with low energy, and be supplied with energy harvested from the environment [26] or small batteries [27]. WSNs can closely monitor each train coupling, detecting and localizing separations without infrastructure support. They can determine train composition and self-configure during exploitation and maintenance.

WSNs covering long freight trains have mostly linear topologies, like those monitoring tunnels, bridges [28], or roads [29]. Most leverage the (almost) linear network topology to improve performance or mitigate potential issues. Proposed protocols are mostly synchronous to reduce channel contentions that waste energy and increase communication delays, such as DiS-MAC [29] designed for roadside and highway monitoring. Network throughputs close to $0.5\,\mathrm{packets/s}$ are demonstrated in various operation conditions, but node clock synchronization over very long networks is especially challenging [30]. Network synchronization may be challenging in variable node operating conditions (e.g., large temperature spread between iced or sunshine exposed nodes), affecting their time reference accuracy [31].

LINE-MAC [32] proposes a trade-off between networking latency and node energy consumption using an asynchronous Medium Access Control (MAC) and adaptive communication protocol for linear network topologies, derived from and sharing the adaptability of the AREA-MAC [33]. It is optimized for generic packet forwarding to the sink, but the adaptive low overhead operation on which it is based can be further optimized for packet formats specific for train integrity assessment.

Node faults can considerably affect the QoS of linear topology WSNs [34]. Node defects or temporary faults due to, e.g., insufficient harvested energy from the environment, can affect WSN sensing and networking unless it is designed with sufficient redundancy. Uneven network traffic distribution can prematurely deplete the energy of some nodes [35] and can be mitigated by application or routing protocol design, or by dynamic traffic redistribution [36]. Train monitoring WSNs can leverage the regular node distribution to optimize node operation and routing protocol [37]. The authors study the performance of five linear routing protocols to determine the optimal transmission distance and hop count, together with data compression techniques for monitoring of the train car rolling bearings.

A train integrity monitoring WSN uses the DMAC layer [38] and communication regularity to extend node sleep, waking up only to forward the packets [6]. Network topology is inferred from radio communication reachability and packet hop count. Train car separations are implied from packet losses. In a refined implementation [15], network formation uses sensor data fusion to form dynamic groups of sensors using car movement correlations. These techniques consider the network propagation time, but not the separation detection time. Also, node time synchronization is assumed to be maintained by the regular

network traffic, but network-level effects of synchronization or (temporary) node loss are not analyzed.

An energy efficiency study of multi-tier multi-hop WSN based on the IEEE 802.15.4 standard relies on multi-car radio communication ranges to increase freight train monitoring reliability and energy consumption [39], [40]. Yet, long-range reliable radio communication in variable railway conditions may be unreliable and network-wide node synchronization, energy consumption, or node temporary faults are not discussed.

In [41], the authors propose a loop network to detect freight train composition and integrity using WSN nodes communicating over a wired bus on the same train car and wirelessly between cars. ZigBee, Bluetooth, and Wi-Fi networks are analyzed to detect broken links (train separations), changes in train composition, or out-of-order packets. The work does not address the energy consumption (which is likely high for Wi-Fi nodes) or the effect of node failures on the system QoS.

It was also proposed a train integrity monitoring network with WSN nodes distributed on the train cars and communicating in a star topology using the IEEE 802.15.4g standard for smart utility networks with the coordinator on the locomotive [42]. The nodes use multiple sensors: GNSS for localization, acceleration and velocity, and the RF received signal strength index (RSSI). Processing is done on the coordinator, which signals accidental train separations when multiple sensing channels indicate it. The system may suffer from the GNSS limitations analyzed earlier. It is also unclear if a network designed for infrequent utility metering communications can support the dense monitoring traffic. Node energy is high due to GNSS receivers. RSSI reliability for car separations detection is also not analyzed.

Safe, effective on-board freight train integrity assessment is still open mainly because of the absence of an electrical power supply on freight train cars. Single solutions hardly satisfy the strict TCAS, ETCS Level 3, ERTMS Level 3, and CENELEC SIL 4 safety requirements. Data fusions from the best technologies, GNSS using inertial sensors, brake pipe monitoring, and distributed wireless, is considered the most promising [14].

We propose a WSN topology, asynchronous communication protocol, application, and sensor nodes prototypes for autonomous on-board train integrity detection, optimized for energy and reporting latency [6]–[8]. Low energy consumption can reduce the equipment and maintenance cost, e.g., smaller battery or energy harvesting device, or longer battery replacement period [15]. To improve the system QoS, we consider unreliable WSN operating conditions, like intermittent node failures, multiple network rejoins, and failing RF communications. The WSN is controlled by and reports to a Control Center (CC) on the locomotive, which can also act as a hub to simplify the integration of the system in railway or Internet of Things (IoT) applications.

To the best of our knowledge, our main contributions are:

- WSN design considering all most relevant distributed train integrity monitoring features: reporting latency and energy (for power supply and/or maintenance cost), and QoS operating with random communication and node failures;
- multi-objective latency-energy WSN optimization;
- QoS characterization for node and communication failures.

Fig. 1. The train model has a locomotive at one end and N train cars (Car 1, ..., Car N) coupled with mechanical couplings (J_1, ..., J_N) whose structural integrity must be checked to assert the overall train integrity.

III. Train Integrity Monitoring Problem Definition and System Requirements

We use the train model shown in Fig. 1, made of couplings that connect the train components. At one end is the locomotive, coupled through coupling J_1 to the first train car, Car 1, and so on up to the last car, Car N, for an N-car train. We distributedly assess the train integrity by checking all couplings, J_i, $i \in \{1, ..., N\}$.

Train integrity monitoring systems are considered safety devices subject to tight CENELEC SIL 4 false positive report requirements (reporting integer train while it is separated). Checking, testing, and certification of standard compliance are beyond the scope of this work, but we both observe the maximum reporting latency requirement of $5\,\mathrm{s}$ [6]–[8], and design the system to reduce false positives. We also test the effects of factors affecting the QoS, such as RF propagation variability, node failures (e.g., for insufficient power or defects), networking overhead for nodes leaving and rejoining the network, and optimize reporting latency and energy consumption.

Generally, a good location of the WSN nodes is the train car bogies because, among others, it is available on most car types [3] and undamped mechanical energy can be harvested to power the sensor nodes [43]. However, RF propagation close to the ground and the irregular metallic surfaces on the car sides is more difficult. Hence, we conservatively assume that the maximum RF propagation is a train car length, including couplings (or about $25\,\mathrm{m}$ [44]), and that the propagation conditions are unstable [45]. RF propagation analysis is beyond the scope of this work, but we will discuss some well-known issues.

We optimize the network energy consumption to reduce the energy supply cost (e.g., smaller batteries or energy harvesting devices), or the cost of maintenance (e.g., change batteries less often [15]). Hence, we assume that WSN operation should be resilient to temporary node failures, e.g., for insufficient energy.

With the above considerations, we define the following WSN design requirements:

- periodical distributed train integrity monitoring, for early detection and reporting of integrity loss, with a period that can change dynamically, e.g., based on train speed;
- support monitoring periods as short as $5\,\mathrm{s}$ [6]–[8] for trains up to $740\,\mathrm{m}$ long, the maximum allowed in Europe [5];
- minimize node energy consumption to lower energy supply cost (e.g., smaller batteries or energy harvesting devices), or maintenance cost (e.g., longer battery duration [15]);
- node idle current consumption around $1\,\mathrm{mA}$ or lower;
- WSN operation resilient to single points of failure, node and communication failures, and with low overhead for node leaving or rejoining the network;
- reduce false positive report likelihood, i.e., asserting the train is connected while it is actually separated;

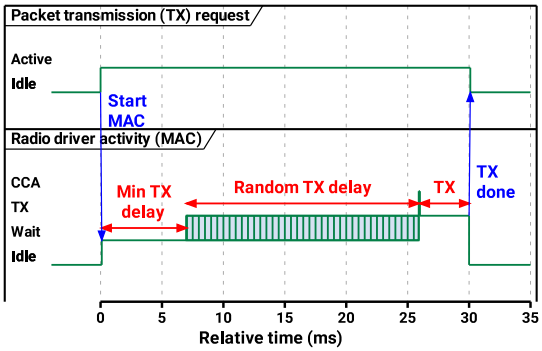


Fig. 2. An application transmission (TX) request starts the TX procedure: a fixed delay is followed by a random delay, a brief clear channel assessment (CCA) and, if clear, the actual TX, else repeat (or abort) the TX procedure.

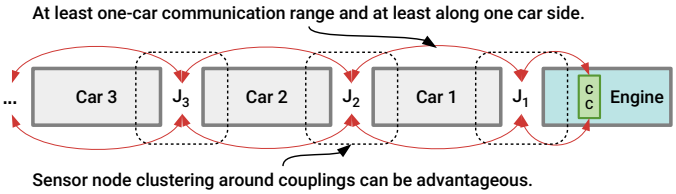


Fig. 3. Train physical structure forces periodically distributed almost unidimensional WSNs, for which we conservatively assume reliable communications for only one car length and on only one side of the cars, but the nodes overseeing the same coupling can communicate among them in a cluster.

- use a CC on the locomotive to control network operation and collect the monitoring reports for engineer notification and easier integration in railway or IoT applications.

We consider that a train integrity monitoring system with these characteristics can both monitor the train stand-alone, as well as complementing monitoring systems based on different technologies, to improve the overall monitoring reliability [14].

IV. Train Integrity Monitoring System Design

Synchronous WSN communication protocols may reduce the energy consumption, but establishing and maintaining synchronous node times can be costly across networks of several tens of hops or when the nodes rejoin the network after, e.g., periods of insufficient harvested energy. Asynchronous protocols can communicate at any time, hence spend energy continuously monitoring the communication channel, but nodes can rejoin the network by simply starting to monitor the communications [46].

We select asynchronous networks using sub-GHz RF bands, which have better link budgets and higher energy efficiency [47]. However, since channel coherence bandwidth is usually larger than the sub-GHz bands, antenna diversity can mitigate better destructive interferences from multipath propagation than the spread spectrum frequency hopping techniques [48].

Node transmissions (TXs) follow the sequence in Fig. 2. After an initial fixed delay, which improves network efficiency (see Section VIII-B), the node waits for a random time within a window to reduce packet collision risk, then briefly checks the channel state (CCA) and, if free, it transmits the packet.

Train geometries force a linear, almost unidimensional periodically distributed WSN (see Fig. 3). Given the uncertain RF propagation discussed in Section III, we conservatively assume only one-car RF communication ranges, and only on the same car side. Hence, we exclude network topologies needing multi-car communications, like clusters or stars, but the nodes overseeing the same train coupling can communicate directly in a cluster to improve networking efficiency. Moreover, routing protocols incurring high rerouting or network join overheads support poorly temporary node failures.

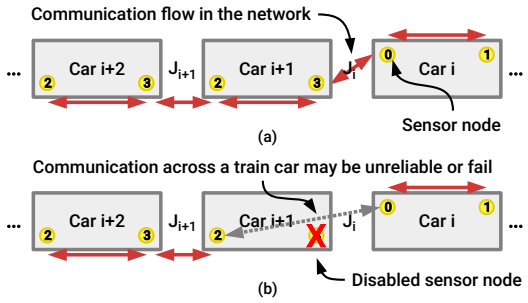


Fig. 4. (a) Nodes only on one car side may force communication to change sides across couplings (red arrows), but (b) networking may fail if the nodes fail, because the communication between opposite car sides may not succeed.

Installing only one sensor at each car end, hence two overseeing each coupling (see Fig. 4, [3]), reduces the system cost and provides redundancy to both communication and integrity assessment (if each node can determine the integrity of the coupling independently, without interfacing with the facing node). However, they may end up on different train sides, forcing the communication to switch sides [see Fig. 4(a)], which has two major adverse effects. First, networking has single points of failure, where it changes car sides [see Fig. 4(b)]. Communication may not go around the failed nodes diagonally, across a train car. Second, coupling integrity assessments relying on measuring the distance between two facing sensors will fail, creating another single point of failure.

The nodes can also be placed in the center of the car sides facing the other cars [3]. This reduces the distance between the sensors overseeing the same coupling, which can help to discriminate the cars of trains on parallel tracks based on distance. However, it cannot provide sensing redundancy unless the nodes can assess independently the coupling integrity, without facing node cooperation. Energy harvesting may be less efficient [43] or add wiring costs between harvesters on bogies and nodes. RF propagation may be more difficult in the relatively narrow channel between the ground and the (irregular) bottom of the cars, which includes obstacles like bogie axles.

We use four nodes to oversee each coupling to avoid single points of failure [see Fig. 5(a)]. Networking continues around the failed nodes on the same train side [see Fig. 5(b)], and the two operational facing nodes continue checking the coupling.

V. Sensor Node Design

Sensor nodes must support the basic system and networking requirements discussed in Section III and Section IV: flat hierarchy multi-hop homogeneous networking, asynchronous MAC and communication protocol with low overhead for nodes joining or leaving the network, and redundant networking and coupling measurements. Hence, the nodes include:

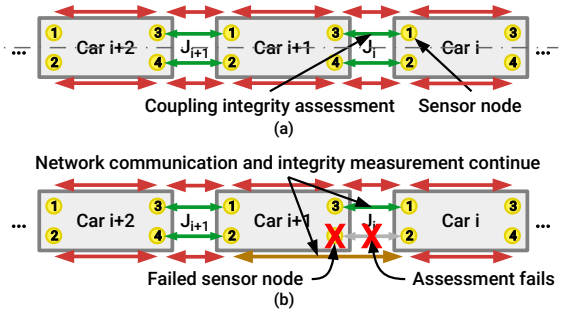


Fig. 5. (a) Four nodes per car allow symmetrical networking and coupling integrity checks and (b) networking or coupling checks go around failed nodes.

TABLE I
CURRENT CONSUMPTION OF 868 MHz TRANSCEIVERS SUPPLIED AT 3.3 V IN TRANSMISSION MODE (TX) AT 10 dBm, RECEPTION MODE (RX), ESTIMATED LOW POWER (IDLE, POLLING) CHANNEL LISTENING MODE (LPL), AND SLEEP MODE (POWER DOWN) WITH FAST WAKE-UP (PD).

Device	Operation mode			
	TX (mA)	RX (mA)	LPL (mA)	PD (nA)
Texas Instruments				
CC1101	30	16.9	1.9	200
CC112x	32	17	1.8	120
STMicroelectronics				
S2-LP	11.5	7.2	0.7	700
Atmel (Microchip)				
ATA5428	17.3	10.3	0.5	10
Semtech				
SX1272	22	10.5	0.7	100
SX1276	25	10.8	0.8	200

- a sensor to reliably measure the coupling integrity;
- a radio transceiver that supports the MAC, WSN protocol, and RF propagation requirements;
- a processing unit (MCU) with adequate processing, storage, and peripheral interfacing capabilities;
- a power supply (e.g., battery or energy harvesting device).

To avoid interacting with the car safety systems, like couplings or buffers, coupling integrity can be assessed by checking the inter-car distance with distance sensors (DS), such as inductive [49], capacitive [50], or RF [41], [51], with varying accuracy, reliability, and energy consumptions. The DS must allow satisfying the TCAS and CENELEC SIL 4 specifications, and consume low power. It may assist network formation and discrimination of adjacent train cars. DS selection and certification are outside the scope of this work. We model it at a high level, with a fixed delay that includes both the DS measurement and MCU communication times.

For the radio communication requirements discussed in Section IV, the most important radio performance are Low-Power Listening (LPL) and reception (RX) modes. Table I shows the maximum current consumption at 3.3 V supply for some widely used narrow-band transceivers. LPL consumption is mostly around or below 1 mA, while the sleep one (PD) is negligible. Note that LPL consumption is estimated (see Fig. 6), because of its spotty documentation. RX mode consumption may matter most because of communication overhearing, while

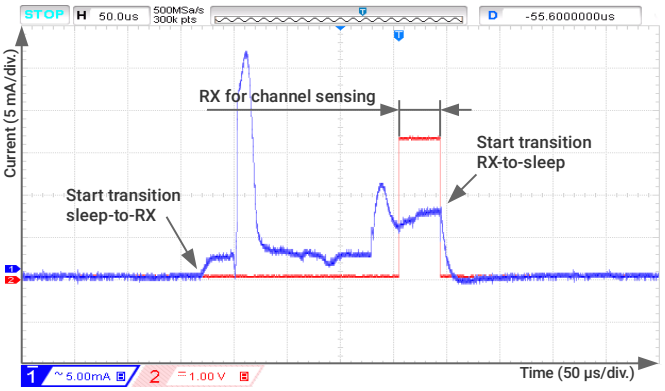


Fig. 6. STMicroelectronics S2-LP current consumption (trace one, blue, < 1 mA average), transitioning briefly (≈ 250 µs) sleep-to-receive (RX), to check for radio activity. Trace two, red, square wave, is high only during RX.

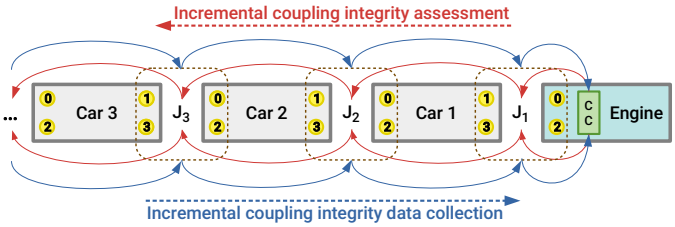


Fig. 7. The control center (CC) on the locomotive starts the train coupling integrity assessment. The sensor node clusters around train couplings assess their integrities in sequence. The nodes at the end of the train start the assessment collection and its return to the CC, along the same multi-hop path.

RX sensitivity is often limited by the environmental RF noise.

We select the STMicroelectronics S2-LP transceiver because it has low LPL, and very low TX and RX mode current consumptions, and embedded support for antenna diversity. Its evaluation kit, STEVAL-FKI868V1 [52], uses an ultralow-power STM32L152RE MCU with an ARM Cortex-M3 CPU at 32 MHz with typical 8.6 mA active current consumption, 512 KB flash, 16 KB of EEPROM, and 80 KB RAM [53]. The MCU resources are sufficient to support the WSN application and communication protocol (discussed later). Most processing is triggered by radio events (TX or RX), which consume significantly more energy [54]. MCU resources should be optimized for energy and cost after the optimization and consolidation of the WSN application, protocol, and encryption requirements (processing and time), which is left for future work.

VI. TRAIN INTEGRITY MONITORING PROTOCOL DESIGN

WSN communication to support train integrity assessment over long trains is challenging, as discussed in Section IV. The CC node on the locomotive (see Fig. 7) controls the WSN operation, receives the coupling integrity reports, and can support the integration in railway or IoT applications [55].

Monitoring the integrity of each train coupling with several local sensors allows detecting train separations earlier than by monitoring full train length variations, which can be affected by variations during normal operation due to track topology or measurement errors [10], [13], [20]. The sensing lag of the brake pipe changes in case of train separation can also be too long [3], [6]–[8]. Moreover, long direct wireless data

TABLE II
TWO-BIT COUPLING STATUS ENCODING CONVENTION

Status bits	Coupling status
0 0	unknown
0 1	normal
1 0	broken
1 1	error

communication between train end and locomotive (extending from several hundreds of meters to kilometers) can often be unreliable, because the last car roof may not be available to install antennas [3], and RF communications along train sides can be affected by variable multipath interferences or repeatedly obstructed, e.g., when bending in tunnels or around hills.

We use a WSN-based distributed coupling integrity check, and train-long communications made of short segments, each at most one car length, to pass the CC commands to the nodes and the reports back to the CC. We assume that both network discovery and formation, and security establishment (neither addressed in this work) are done whenever the train composition changes, and before the WSN starts assessing the train integrity. Each node must have a unique factory ID, usually provided by the MCU (e.g., the STM32 family MCUs have 12-byte long unique IDs), but to improve network performance (e.g., shorter packets, less memory) we assume that the nodes are assigned shorter IDs, as shown in Fig. 7, concatenated to the coupling ID

$$\text{node short ID}\big|_{1\text{ byte}} = \underbrace{\text{J J J J J J}}_{\text{6-bit coupling ID}}\ \underbrace{\text{N N}}_{\text{2-bit node ID}} \quad . \qquad (1)$$

This allows extracting the node and coupling IDs using low effort bit mask and shift operations and can be used two bytes for trains beyond 60 cars. Network-specific encryption keys reject packets from adjacent trains, regardless of their network IDs.

The train integrity assessment protocol has two major objectives: to periodically check all train coupling integrity, and to deliver the aggregated results to the CC. The process is started by a CC command and makes two train traversals: a forward leg, from locomotive (CC) to train end, during which the integrity of each coupling is sequentially assessed, followed by a backward leg, during which all integrity assessments are collected and returned to the CC. During the forward leg, the four nodes assessing the integrity of each coupling fill a

$$\text{coupling status byte} = \underbrace{b_7\, b_6}_{\text{node 3}}\ \underbrace{b_5\, b_4}_{\text{node 2}}\ \underbrace{b_3\, b_2}_{\text{node 1}}\ \underbrace{b_1\, b_0}_{\text{node 0}} \qquad (2)$$

whose bits $b_i, i \in \{0, 1, \ldots, 7\}$ are independently controlled in groups of two by each of the four nodes overseeing the coupling, using the encoding in Table II. The CC analyzes the node reports in the coupling status bytes to determine the coupling integrity, according to the application requirements. For instance, the CC may report train separation even if only one node reports coupling integrity loss. The trade-off depends, among others, on the accuracy and reliability of the DS, which is beyond the scope of this work. The CC may also collect statistics on node broken, error, or unknown reports (see Table II) over different time frames to present other system status

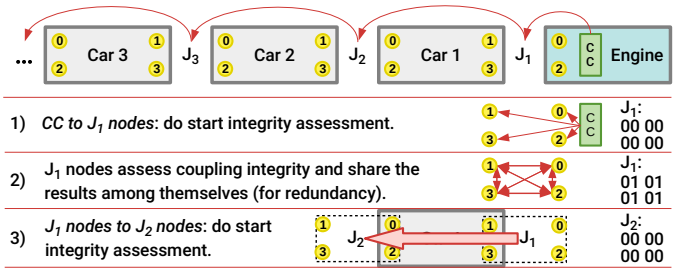


Fig. 8. Distributed coupling integrity assessment starts in step 1 with the control center (CC) request to the nodes of the first coupling, J_1. They reset their coupling status byte (bits shown on the right) and assess the coupling integrity. In step 2, they locally broadcast the results to fill the status byte. In step 3, they forward the start request to the nodes of the next coupling, J_2.

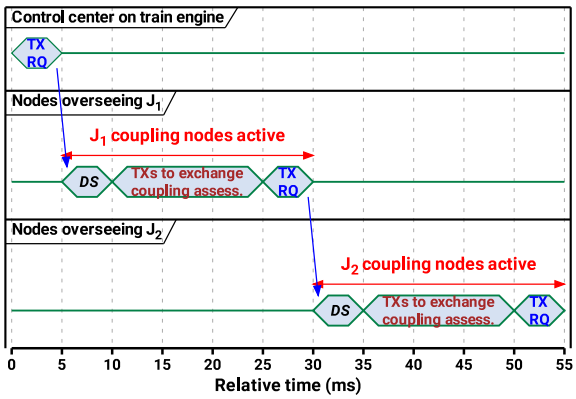


Fig. 9. Sequence (approximate timing) of the major events of the coupling integrity assessment. The control center sends the request (RQ) to the first coupling nodes, which check its integrity (DS), then fill and broadcast the assessment results. Finally, the J_1 nodes TX RQ next coupling nodes, repeating the steps until the train end.

views, including, e.g., node and network operation reliability or permanent node faults in need of maintenance.

Fig. 8 shows an overview of the main protocol communications for coupling assessment, including request propagation along the forward leg (from CC to train end). Fig. 9 shows the main activity sequence (approximate timing) over the same period. Row 1 in Fig. 8 shows the CC broadcasting the "start assessment" request (TX RQ on the top plot in Fig. 9) to all nodes overseeing the first coupling, J_1. Each node resets its copy of the coupling status byte, then starts checking the coupling integrity with the DS (DS activity on the middle plot in Fig. 9). Next, they disseminate the results to their peers using in-coupling broadcasts (row 2 in Fig. 8). In the end, all nodes have an identical copy of the coupling status byte, for redundancy, filled with their own assessment and those of their peers. Finally, they forward the "start assessment" request to the nodes overseeing the next coupling (thick arrow on row 3 in Fig. 8 and TX RQ in the middle plot in Fig. 9), which repeats the process.

Fig. 10 shows the main events and in-coupling data dissemination for a generic example of integrity assessment of coupling J_i (see row 2 in Fig. 8). The coupling nodes use their DSs to assess its integrity upon receiving an *RQ* message from the previous coupling nodes. Then, they fill the results encoded as in Table II in their coupling status byte copy, shown in Fig. 8

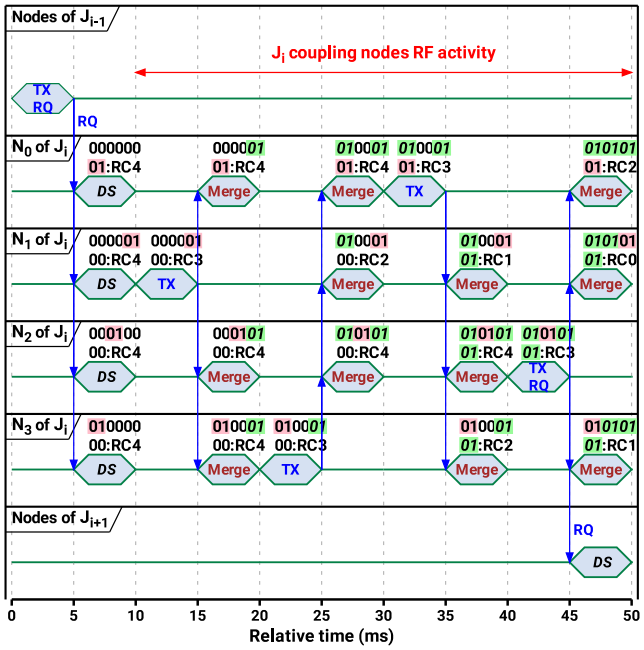


Fig. 10. Event sequence example for assessing J_i coupling integrity (approximate timing). Nodes N_0–N_3 check the integrity with their DSs upon previous coupling request, *RQ*, set data repeat counters (RC), update the status bytes (pink), and start broadcasting them (TX). Peers merge broadcast data (green) and decrement RC whenever they receive their own data. When done, the nodes forward the *RQ* request to the nodes of the next coupling, J_{i+1}.

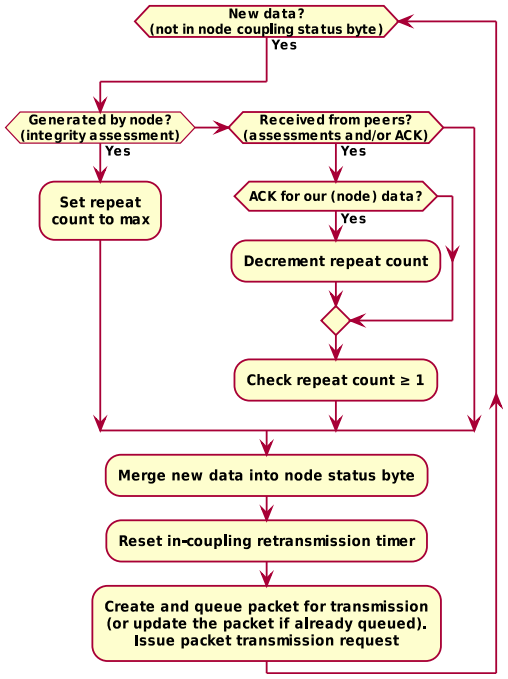


Fig. 11. New data (internal, e.g., from coupling assessment, or external, from peers) are merged by the node into its coupling status byte copy, and the data repeat counter is adjusted each time new data is produced or is echoed by peers.

above the activities, and set the broadcast repeat counter to four, RC4 (the number of repeats is a protocol parameter which we optimize later). When they start broadcasting, one node transmits (TX) first, e.g., N_1, but all nodes are equivalent and the order is irrelevant. N_1 decrements its repeat counter (RC3) and keeps the message for additional broadcasts if needed. All peers merge N_1 data with their own results and attempt to broadcast their updated status bytes. We assume N_3 transmits now and, as N_1 before, it decrements its repeat counter (RC3) and keeps the message. Note that N_1 decrements again its repeat counter (RC2) because the N_3 broadcast included N_1 data. As before, all peers (including N_1) merge N_3 data with their status byte copy. When the procedure repeats, e.g., with N_0 broadcasting, we note that now the N_2 status byte holds a copy of all peer assessments. Hence, the subsequent N_2 broadcast doubles as an integrity assessment request (*RQ*) for the nodes of the next coupling, J_{i+1}. They will start their DS phase while the nodes N_0, N_2, and N_3 of J_i will still broadcast, acting also as *RQ* repeats (omitted in Fig. 10 for readability).

At the train end, the nodes emit a command propagating back to the CC and collecting all status assessment bytes in a

$$\text{train status vector} = |\,\text{SB}_N\,|\,\text{SB}_{N-1}\,|\,\ldots\,|\,\text{SB}_2\,|\,\text{SB}_1\,| \quad (3)$$

for N train cars, made of all status bytes SB_n, $n \in \{1, \ldots, N\}$. Note that when starting the coupling integrity assessment, the nodes also set a timer slightly longer than the time when the backward propagation packets are expected to pass through their coupling. If the forward leg propagation is interrupted before reaching the end of the train (e.g., because of a wide train separation, multi-node failures, or RF interferences), the last nodes on the forward leg will start propagating back to the CC the train status vector [see (3)] when the timer expires.

In-coupling broadcasts are repeated to improve the communication robustness so that each node receives the peer data and builds a full copy of the coupling status byte. They are driven by two basic rules (see Fig. 11). First, any node that produces new data (e.g., a coupling integrity assessment) should ensure that those data are broadcast the designated number of times. For this, the nodes set a counter (RC in Fig. 10) and decrement it each time the new data they produced are broadcast, either by the nodes themselves or by their peers. Second, any node receiving new data from peers will merge them with its own data, then broadcast the merged result at least one time (along with all the assessment data that it gathered). Hence, most transmissions act also as data retransmissions for one or more peer nodes. We can follow this process on the example in Fig. 10, e.g., the data broadcast by N_3 at 20 ms includes the data produced by the node itself [encoded on bits six and seven, see (2), highlighted red], as well as the data collected by N_3 from the previous broadcast of N_1 at 10 ms [encoded on bits two and three, see (2), highlighted green]. The peer nodes (N_0, N_1, and N_2) merge at 25 ms the data from the broadcast of N_3. For all of them, the data produced by N_3 (encoded on bits six and seven) are new, because they were never broadcast before. Hence, they all trigger the path "new data → received from peers" in Fig. 11, which sets the repeat count (RC in Fig. 10) to one, unless it was already set higher. We see in Fig. 10 that all RCs have higher values, and specifically: four for N_0 and N_2 (set when they assessed the coupling integrity during DS phase, and never broadcast), and two for N_1. The latter was four after the DS phase, decremented once after N_1 broadcast its assessment at 10 ms, and decremented again

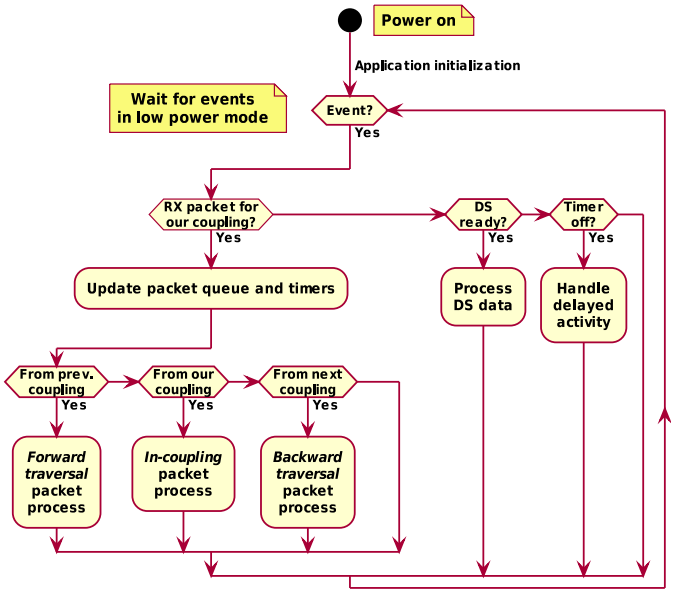


Fig. 12. Node application top events: packet reception (RX), coupling integrity check (DS), and various timer expirations. Forward traversal RXs start DS. Backward traversal RXs collect DS results. Node application sets expiration timers for most tasks, including for packet retransmissions (re-TX).

Listing 1. Code example of packet transmission (TX) in ns-3 simulation. The C function `train_wsn_tx()` arms a `ms` delay timer which calls `doTX()` method at expiration. `doTX()` transmits the next queued packet, obtained by the C function `tx_packet_get()`.

```
extern "C" void train_wsn_tx(uint8_t ms) {
 GetApplication()->GetObject<trainApp>()->scheduleTX(ms);
 [...] }
void trainApp::scheduleTX(uint8_t ms) {
 Simulator::Schedule(MilliSeconds(ms), &trainApp::doTX);
 [...] }
void trainApp::doTX() {
 McpsDataRequest(Create<Packet>(tx_packet_get()));
 [...] }
```

Listing 2. Code example of embedded (FreeRTOS) packet transmission (TX). `train_wsn_tx()` arms the `tx_timer` timer to `ms` delay. Upon expiration, it calls `tx_do()` that starts task `TXTaskHandle` to send the next queued packet.

```
void train_wsn_tx(uint8_t ms) {
 xTimerChangePeriod(tx_timer, ms, 0);
 [...] }
void stubs_init(void) {
 tx_timer = xTimerCreate([...], tx_do);
 [...] }
void tx_do(TimerHandle_t xTimer) {
 xTaskGenericNotify(TXTaskHandle,TX_START_EVENT,[...]);
 [...] }
```

when N_3 rebroadcast the N_1 assessment once more at $20\,\mathrm{ms}$.

VII. Node Application Design

Node application implements all node functions and the train integrity monitoring protocol. Fig. 12 shows the main external (e.g., peer messages) or internal (e.g., timer expirations) processed events, using few top-level states.

Peer communications can come from the previous coupling (e.g., the *RQ* in Fig. 10), the same coupling (e.g., the RF activity in Fig. 10), or the next coupling [e.g., the status vector, see (3)]. Forward train traversal packets for integrity assessment hold

$$\mathrm{P_{fwd}}: \,|\,\mathrm{pkType}\,|\,\mathrm{rqID}\,|\,\mathrm{toCID}\,|\,\mathrm{fromNSID}\,|\,\mathrm{stByte}\,|\,\mathrm{sig}\,|, \quad (4)$$

i.e., packet type (one-byte pkType), unique request ID (one-byte rqID), ID of target coupling (one-byte toCID), ID of the originating node (two-byte fromNSID), coupling status byte [stByte, see (2)], and a 32-byte cryptographic signature (sig), for a total of 38 bytes. Backward traversal packets hold

$$\mathrm{P_{BWD}}: \,|\,\mathrm{pkType}\,|\,\mathrm{rqID}\,|\,\mathrm{fromCID}\,|\,\mathrm{stVect}\,|\,\mathrm{sig}\,|, \quad (5)$$

with one-byte fromCID the ID of the source coupling and 50-byte stVect [see (3)] the train integrity status vector, for 85 bytes total. Adding the S2-LP radio modem 11-byte overhead, the forward and backward traversal packets have 49 and 96 bytes, respectively, plus the packet preamble (discussed later).

Recently seen RX messages (SM_i) and outgoing TX messages (AM_i) are stored in a message queue made of two concatenated first in, first out (FIFO) data structures

$$\mathrm{Queue}: \underbrace{\mathrm{AM}_1\,\mathrm{AM}_2\,\ldots\,\mathrm{AM}_n}_{\rightarrow\ \text{Active message FIFO}\ \rightarrow}\,|\,\underbrace{\mathrm{SM}_1\,\mathrm{SM}_2\,\ldots\,\mathrm{SM}_m}_{\rightarrow\ \text{Seen message FIFO}\ \rightarrow}. \quad (6)$$

The seen message FIFO helps to discard old messages. TX messages stay active until resolved (transmitted, retransmitted, or obsoleted). We leave FIFO optimization for future work since its size strongly depends on the RF communication range.

Queued messages can trigger timers. E.g., upon the RX of the *RQ* message shown in Fig. 10, all J_i nodes start DS for coupling measurement and set timers for maximum DS time. Similarly, TX messages are loaded into the active FIFO [see (6)] attached to a repeat counter and to a retransmit timer (see Section VI).

Notably, this reactive operation mode allows a node to rejoin the network by just starting to listen and react to ongoing communications. Also, the CC can change the train integrity monitoring period simply by issuing the request commands at the appropriate time, to which the nodes react as shown in Fig. 9. This minimizes the communication overhead but serializes the integrity assessments, which thus weigh more on the overall assessment time which is limited to $5\,\mathrm{s}$ (Section III, [6]–[8]).

We designed the node application in three steps. First, we implemented a high-level C++ model of the integrity assessment shown in Fig. 8 in the ns-3 network simulator [56] to check that it completes within the allotted $5\,\mathrm{s}$ (Section III, [6]–[8]). Then we select FreeRTOS [57] as embedded OS from various options [58] for its features, reduced resource requirements (a typical FreeRTOS kernel needs 4000–9000 B of code [59]), integration with the configuration tool of our development kit, and easy porting of projects [60] to the IEC 61508-SIL 3 certified Safety Critical Real-Time OS (SAFERTOS) [61] (but can be used other embedded operating systems that support typical features like timers and threads). Thus, we develop the application behavior directly in the target C language, but we test it in ns-3 simulations with interfaces that simplify its porting to FreeRTOS. For instance, Listing 1 and Listing 2 show the implementation of the `train_wsn_tx()` interface, which schedules the TX of a queued packet. Listing 1 uses the ns-3 simulator primitives to implement the delay timer, which then calls a method that TX the first queued packet. Listing 2

shows the same interface using the FreeRTOS primitives to implement an equivalent flow to unblock the message TX task.

Ported almost unchanged to FreeRTOS, the application behavioral code, including stubs, drivers, and the FreeRTOS kernel needs 73 120 bytes for code and 31 822 bytes for data. We check the application operation on a testbed with 23 nodes: 20 covering five couplings, 2 at the train end, and 1 for the CC. We analyze the node and network behavior in various conditions, comparing the testbed behavior with the corresponding ns-3 simulations to accelerate code debugging and validation.

VIII. Simulation and Experimental Results

Experimentation focuses on 1) checking hardware node operation, energy consumption, and validation, 2) simulations for train integrity protocol optimization, and 3) simulations for QoS evaluation with variable node and communication failures.

First, we qualitatively observe and optimize the node hardware operation: first in controlled test conditions, then in a scaled-down network with a CC and 22 nodes, covering five train couplings and the train end, working in close proximity to increase the communication channel congestion.

Next, we make quantitative measurements and optimizations using ns-3 simulations running the embedded WSN application node code almost unchanged (see Section VII). All simulations use 200 nodes that cover 50 couplings, for a train length of $\approx 1\,\mathrm{km}$, about $35\,\%$ longer than the current $740\,\mathrm{m}$ limit in Europe [5]. We conservatively model a lossless communication channel covering three couplings ($\approx 60\,\mathrm{m}$).

We also simulate the WSN QoS degradation with node and communication failures, by disabling various fractions of nodes or communications, respectively.

Finally, we qualitatively check the network operation in outdoor tests, with RF propagation resembling train car conditions.

We are mostly interested if WSN behavior is correct, both at the node and the network level. E.g., if the application code developed and tested in ns-3 simulations works correctly, if the integrity assessment is correct in normal and fault conditions, if there are any systematic errors over long runs, if the application states and transitions are the expected ones for various types of events, or if the packet loss is not excessive. We detect errors mostly from the analysis of simulation logs (for network operation) and operation, code, and signal trace analysis in isolated, out-of-network conditions (for node and drivers operation).

A. Hardware node configuration and testing

We use the S2-LP transceiver maximum data rate, $250\,\mathrm{kbit/s}$, to minimize packet TX times, communication channel congestion, and meet the integrity assessment tight timing constraints.

Longer packet preambles reduce the LPL duty cycle but increase the TX and RX consumption, and channel congestion. The latter can increase the packet loss, hence the retransmission energy. The network-optimal LPL duty cycle is hard to find due to hardware constraints, especially when packet overhearing depends on variable RF propagation conditions.

Considering the performance of several state-of-the-art transceivers (see Table I) and the application requirements, we aim for an LPL current around $1\,\mathrm{mA}$. Configuring the sniffing time around $38\,\mu\mathrm{s}$, by trial and error we settle for a 50-byte preamble (which is comparable with our shortest packet length of 49 bytes, see Section VII). With these, we measure less than $1\,\mathrm{mA}$ average LPL current, visible also in Fig. 13.

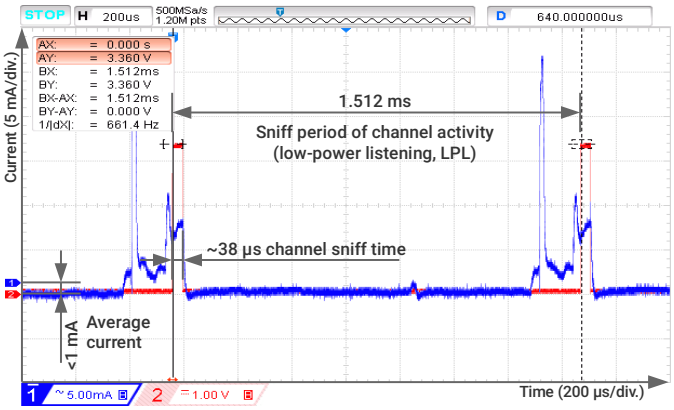


Fig. 13. S2-LP transceiver current consumption and timing while sniffing the channel for incoming transmissions in low-power listening. The average current consumption is $<1\,\mathrm{mA}$ for sniff time $\approx 38\,\mu\mathrm{s}$ and sniff period $\approx 1.5\,\mathrm{ms}$.

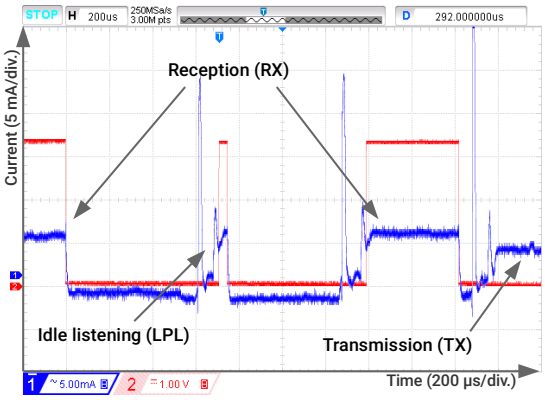


Fig. 14. S2-LP transceiver mode switching: reception (RX), idle channel listening (LPL), and $0\,\mathrm{dBm}$ transmission (TX), after full RX.

We test on hardware the peripheral operation and interfacing. We check LPL RX with one node in LPL and another transmitting $4\,\mathrm{ms}$ packets first every $15\,\mathrm{ms}$, then randomly between $10\,\mathrm{ms}$ and $20\,\mathrm{ms}$. Packet loss is less than $0.01\,\%$ over several hours, and lower at night. The losses are acceptable, mostly due to external interferences. Then, we test the TX/RX interlocking with the test node attempting to TX overlapped on various RX phases: preamble, sync word, payload, packet transfer to the microcontroller. We check that the RX is not interrupted and the TX is postponed, e.g., Fig. 14 shows an example of radio mode switching between RX (left), LPL (middle), and a new RX closely followed by a TX (rightmost) — note that we shortened the packets and the LPL periods for readability and the TX is low power. Finally, we test the carrier sense function with two nodes that randomly transmit packets while avoiding collisions.

B. Simulations of train integrity protocol

We use ns-3 simulations of the train integrity WSN protocol to functionally check the code and protocol, and optimize the operation reliability and energy consumption.

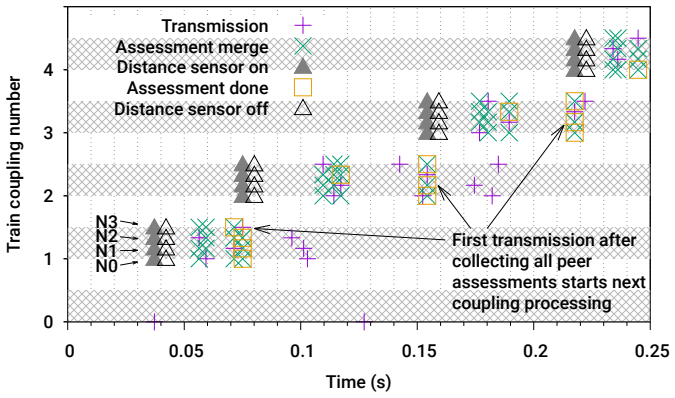


Fig. 15. Coupling integrity assessment starts with distance sensor measurements upon request from previous coupling nodes, then exchange the results between the coupling nodes, and ends by forwarding the request to the next coupling nodes. Coupling and node IDs are encoded on the Y-axis.

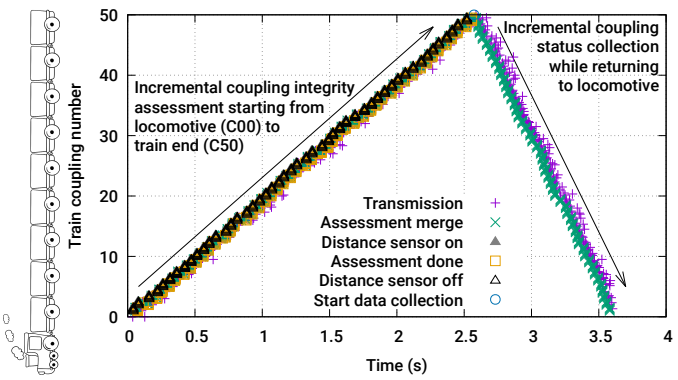


Fig. 16. Propagation of node activity in time (X-axis) along the train (Y-axis) during one train integrity assessment for 50 couplings ($\approx$1 km long train), starting from the locomotive and propagating towards the end of the train, then propagating the result collection back to the locomotive.

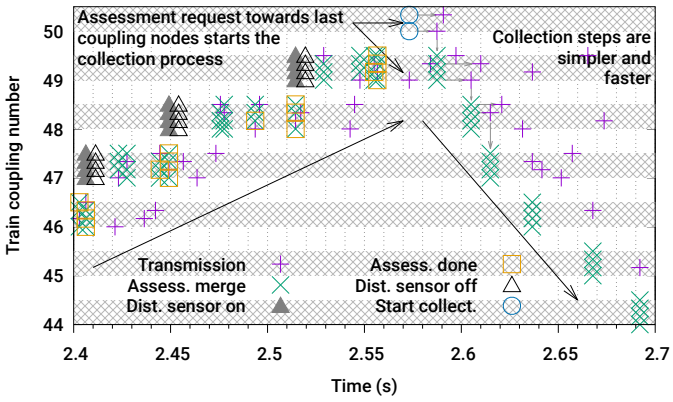


Fig. 17. Coupling integrity assessment events at the end of the train and the start of backward traversal collecting coupling assessment results. Coupling and node IDs are encoded on the Y-axis.

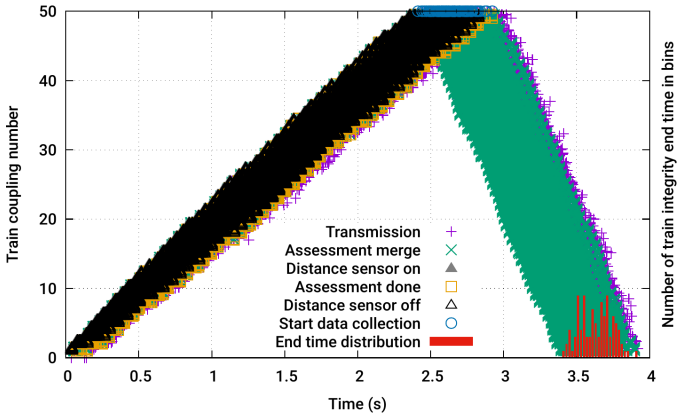


Fig. 18. Propagation of node activity in time (X-axis) along 50 train couplings ($\approx$1 km long train, left Y-axis) for 100 randomized train integrity assessment simulations using the best protocol parameter setup, and with the assessment duration distribution plotted on the bottom right (right Y-axis).

Fig. 15 shows in detail the main events discussed in Section VI and outlined in Fig. 8, Fig. 9, and Fig. 10, simulated for each node, grouped by coupling on the Y-axis. Within a coupling Y interval, the nodes overseeing it have increasing offsets within the hatched stripes. Coupling C0 is virtual, associated with the CC. Close to 0.04 s, the CC requests a train integrity assessment, and all nodes of the first coupling, C1, start DS (solid triangles in Fig. 15). We assume that the DS lasts 5 ms and an empty triangle marks its end. Then, the coupling nodes broadcast their assessments (+ mark), determining peer data merges ($\times$ mark). Squares show when nodes are done collecting all peer data, hence their next TX also requests the next coupling (C2) nodes to start the assessment. Once they start their coupling assessment, C2 nodes ignore redundant requests sent by other C1 nodes, around 0.1 s, the same way C1 nodes ignored a redundant CC request around 0.13 s. C2 nodes proceed as C1 nodes, then they activate the C3 nodes and so on, traversing forward the whole train. Fig. 16 shows the progressive node activation in time (X-axis) and along the train (Y-axis) during coupling integrity assessment (left ramp), while doing the activities shown in detail in Fig. 15. The nodes at the train end start the backward train traversal ($\circ$ marks at the top of Fig. 16 and Fig. 17) to collect and return to the CC all integrity assessments in the status vector [see (3)]. Collect packets are longer (see Section VII) but propagate faster (steeper right ramp slope in Fig. 17) because they need fewer node actions per coupling: fill the assessed status in the vector and forward it.

Fig. 18 shows a qualitative overview of the evolution in time (X-axis) along 50 train couplings ($\approx$1 km long train, left Y-axis) of the main train integrity assessment events for 100 random simulations for the best network parameter setup (see below). The left ascending ramps are made of events occurring during the forward train traversals, like those shown in Fig. 16. At the top, around 2.6 s, are started the backward traversals that form the right descending ramps. Forward train traversal slopes (left ramp) have higher variance because of the higher number and complexity of the in-coupling operations and message exchanges than on the backward train traversal (right ramp, Fig. 17). At the bottom right of Fig. 18 is plotted the assessment time distribution (right Y-axis), a Gaussian between 3.41–3.92 s, with 3.63 s average and 0.11 s standard deviation (SD).

Train integrity reporting time depends on the node TX delays discussed in Section IV. It decreases when reducing the minimum TX delay and window, but channel congestion and aborted transmissions increase. Hence, we explore the WSN protocol performance for five minimum TX delay values

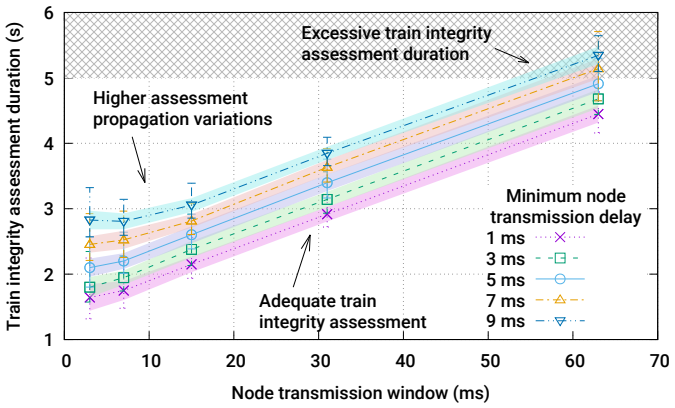


Fig. 19. Variation of train integrity assessment duration [mean, min, max, and standard deviation (SD) for 100 assessments] function of min transmission delay and window for four in-coupling data repetitions and 5 ms distance measurement time. Color bands are SD, error bars are min and max times.

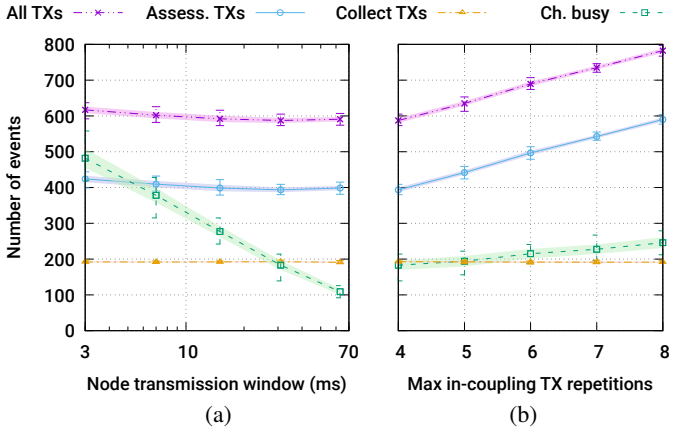


Fig. 20. Transmissions (TX) and channel busy events for 100 full train integrity assessments with 5 ms TX min delay, function of (a) TX delay range for four in-coupling data repetitions, and (b) maximum in-coupling data repetitions for 31 ms TX window. Color bands are standard deviation, error bars are min/max.

TABLE III
WSN PARAMETERS EFFECT ON TRAIN INTEGRITY (TI) ASSESSMENT PERFORMANCE. FEWER TRANSMISSION (TX) REDUCE ENERGY.

TX reps.	TX delay min (ms)	TX delay win. (ms)	TI time (s)	TXs per node total	TXs per node assess	TXs per node coll.	Chan. busy	TI OK (%)
Best configurations								
4	7	31	3.63	2.92	1.95	0.97	168	**100**
4	9	31	3.85	**2.91**	**1.93**	0.97	**158**	**100**
5	5	**15**	2.64	3.22	2.25	**0.96**	295	**100**
5	7	**15**	2.87	3.19	2.22	**0.96**	262	**100**
6	**3**	**15**	**2.42**	3.50	2.54	**0.96**	345	**100**
6	7	**15**	2.91	3.47	2.50	**0.96**	277	**100**
Worst configurations								
7	**1**	**3**	**1.63**	3.90	2.94	**0.95**	796	**87**
4	7	**3**	2.45	**3.08**	**2.11**	0.96	**442**	86
6	**1**	**3**	1.67	3.68	2.72	**0.95**	773	84
6	3	**3**	1.83	3.60	2.63	0.96	593	83
8	**1**	**3**	1.64	4.14	3.18	**0.95**	817	83
5	**1**	**3**	1.65	3.44	2.47	0.96	750	77

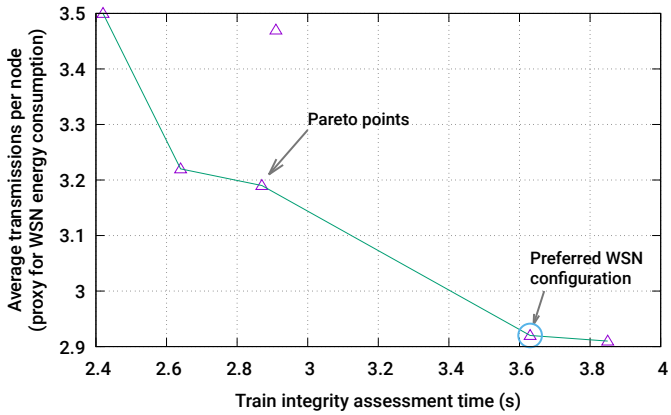


Fig. 21. Five of the six best WSN configurations are Pareto points in the multi-objective optimization of train integrity assessment time and average packets transmitted by the nodes, which are a proxy of WSN energy consumption.

(1 ms, 3 ms, 5 ms, 7 ms, and 9 ms), five TX window widths (3 ms, 7 ms, 15 ms, 31 ms, and 63 ms), and five in-coupling data repetitions (4, 5, 6, 7, and 8). We simulate 100 times each parameter combination, 12 500 total simulations.

Fig. 19 shows the train integrity assessment duration function of TX minimum delay and window width (see Section IV): average, SD (colored area), and error bars for min/max times. TX minimum delay only adds to the assessment average for larger TX windows, while for smaller windows it also worsens channel congestion, hence increases the min/max range and SD.

Fig. 20 shows total TXs and channel busy (failed CCA) events function of TX window width and the number of in-coupling broadcasts (same representations as in Fig. 19). Fig. 20(a) shows that channel busy events, their min/max ranges, and SD increase exponentially for smaller TX windows. Forward traversal TXs increase only slightly because the CCA avoids most conflicts, and backward traversal TXs are virtually constant. More in-coupling data repetitions increase both the channel busy events and the number of TXs on forward train traversal. The TXs increase less than in-coupling data repetitions, because most in-coupling TXs repeat data for more than one node, as discussed in Section VI. Backward traversal TXs are mostly constant because in-coupling data exchanges stop as soon as the next coupling picks up the forward request.

Table III shows the best and worst six network parameter combinations for all 12 500 simulations, in order of decreasing communication reliability (TI OK), increasing average TX per node, in-coupling data repetitions (TX reps.), and train integrity assessment duration (TI time). The best configurations report within the 5 s requirement [6]–[8] and almost all are Pareto points (see Fig. 21). We prefer the configuration with lower average TXs per node and less in-coupling data repetitions because they minimize the overall networking energy consumption (each TX potentially triggers an RX in all nodes in RF communication range). For this prototype, we ignore other node energy consumption, e.g., for data processing or encryption. We also prefer configurations finishing the assessment earlier, because they may less likely exceed the 5 s TI time requirement [6]–[8] due to unforeseen events, e.g., offline nodes or communication errors. The best network configurations have larger TX windows, but not the largest that may violate the TI time requirement, and also have longer TX fixed delays. The worst

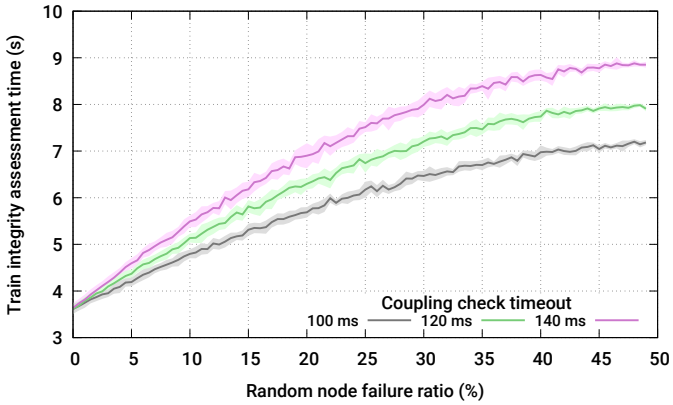


Fig. 22. Train integrity reporting time and standard deviation function of faulty node ratio and the maximum time allowed for coupling integrity checks.

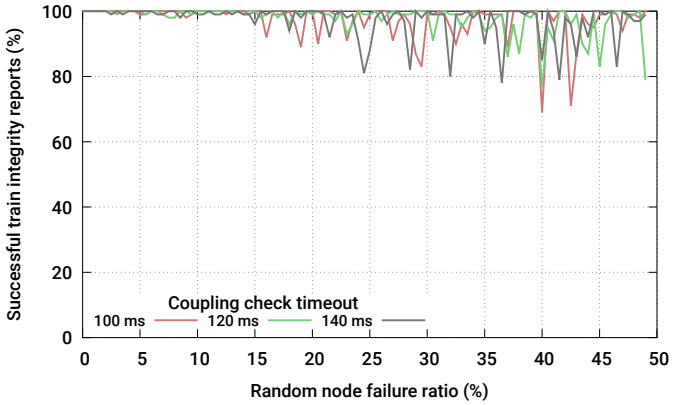


Fig. 23. Successful train integrity reports function of fail node ratio and the maximum time allowed for coupling integrity checks.

configurations have shorter TX windows and fixed TX delays.

We can estimate node networking energy consumption from the TX average and RF communication range. For example, for a three couplings RF communication range, TXs are received by 3 coupling nodes+4 nodes×3 couplings×2 sides = 27 nodes. So, 2.92 TXs average per node would trigger 27×2.92 = 78.84 RXs. This is an upper bound, because RF range may be much shorter, may not span train sides, and nodes closer to train ends have fewer nodes in RF range. Networking energy depends on RF propagation, which is outside the scope of this work.

C. Simulation of train integrity resilience to errors

Through ns-3 simulations, we evaluate WSN resilience to lost messages and node failures as 1) variation of reporting time, 2) the number of incomplete or missing reports, and 3) routing reconfiguration around failed nodes. The nodes use the best configuration from Section VIII-B.

Node failures are simulated from zero (no node failure) to 100 failed nodes, i.e., 50 % of the 200 WSN nodes. Disabling more nodes would leave at least a coupling with less than two facing nodes, for which the DS fails (it needs two facing nodes). Each node fault ratio is simulated 100 times with random faulty node distributions to check routing reconfiguration capability.

Missing (failed) nodes trigger timeouts in the operational coupling nodes: timers associated with the RC counters (see Section VI and Fig. 10), and a coupling check timer. The former is marginally longer than the normal node TX delay and has a limited effect. We explore instead the effects of the latter for a few realistic values: 100 ms, 120 ms, and 140 ms. Note that the 100 ms timeout is only slightly longer than the node average time to complete in-coupling transmissions (see Fig. 2 and Fig. 10) using the best node configuration in Table III

$$5\,\text{ms}\;\textit{DS}{+}4\;\textit{TX reps}\cdot\underbrace{\left(7\,\text{ms}\;\textit{fixed}+\frac{31\,\text{ms}\;\textit{random}}{2}\right)}_{\textit{Average delay for one TX}}=95\,\text{ms}. \tag{7}$$

Fig. 22 shows the train integrity report delay function of faulty node ratio for three coupling check timeouts. The assessment of couplings with faulty nodes wait for coupling check timer expiration, hence increase with the expiration times. The

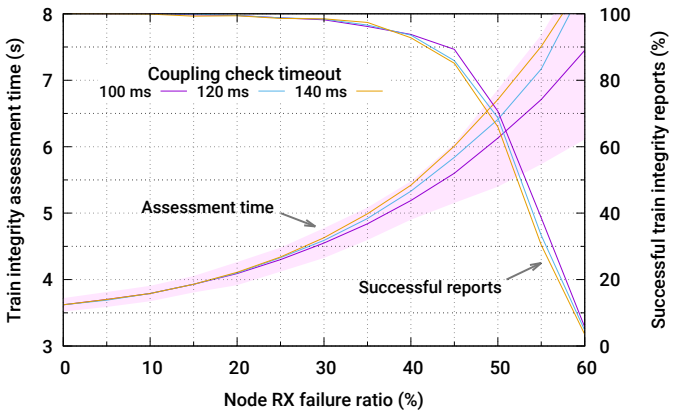


Fig. 24. Successful train integrity reports and reporting latency (with the standard deviation colored area plotted only for 100 ms for readability) function of the network-wide failed message reception (RX) ratio and the maximum time allowed for coupling integrity checks.

increase rates slow for higher node fail ratios because more faulty nodes per coupling do not increase the check timeout.

Reporting times beyond the 5 s requirement (Section III, [6]–[8]) is acceptable because most nodes may fail due to insufficient supply energy, e.g., when energy harvesting decreases with train speed and reporting requirement is also longer [6]–[8].

Fig. 23 shows the valid train integrity reports function of faulty node ratio for three coupling check timeouts. We note that virtually all reports are valid low fail node ratios, and the invalid reports do not seem to depend on the coupling check timeout.

We simulate the WSN reporting performance function of the communication error ratio by randomly discarding RX messages in each node. Fig. 24 shows the dependence on the RX error ratio of both the train integrity reporting latency and of the valid report ratio, for three coupling check timeouts. Each combination was simulated 1000 times, 39 000 simulations total.

We note in Fig. 24 a steep drop of the valid report ratio after 35–40 % RX failure ratios, regardless of the coupling check timeout, reaching 0 % valid reports shortly above the 60 % RX failure ratio. High RX failure ratios disrupt mostly the request forward propagation along the train, leading to incomplete reports. Reporting latency and SD also increase with the RX

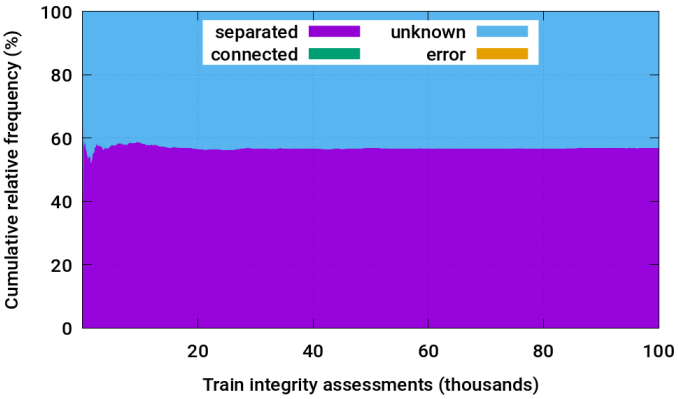


Fig. 25. Cumulative relative frequency over 100 000 train integrity assessments with random node and communication failure rates and one random train coupling separated. The train is reported either *separated* or in *unknown* state when reporting fails due to network errors.

fail ratio, because the retransmissions increase. However, with up to a third of messages lost, more than 95 % of train integrity reports are valid and within the required 5 s latency [6]–[8].

To check for WSN assessing and reporting errors in case of node and networking failures, we have simulated 100 000 train integrity assessments with the train separated at a random coupling, and for random ratios of failed nodes and communications. As discussed in Section VI, the CC inspects all node reports. If one node reports a disconnected coupling, the CC reports that the train is *separated*. Else, if there are no node reports for at least one coupling, the CC reports the train status as *unknown*. Else, if the nodes report all couplings connected, but at least one node reports internal errors, the CC reports the WSN status in *error*. If none of the above, then the CC reports the train as *connected*. Note that based on the actual application needs, other summarizing logic or level of detail can be devised from on the node reports.

As shown by the cumulative relative frequency in Fig. 25, the train state is either reported correctly as *separated* or in *unknown* state when node and communication errors prevent the assessment or reporting. The WSN never reports the train connected, which would be a false positive for which the CENELEC SIL 4 standard imposes very tight limits.

D. Qualitative network field tests

We use experimental tests to qualitatively check the network operation and stability in different operation conditions, with 22 sensor nodes that cover five couplings, the train end, and a CC. Laboratory tests keep all nodes in close range to increase the overhearing and communication channel congestion. Outdoor tests try to emulate the train-side propagation conditions.

In the laboratory (Fig. 26), all nodes are within about 1 m to increase packet overhearing and collisions. Using high-speed serial connections, we monitor the CC requests and reports, and node states and activities. We check that the behavior of the embedded application code corresponds to the ns-3 simulations, both emulating the DS operation (see Section V). We also check the network resilience to single points of failure by turning off one node at a time. We check that the monitoring

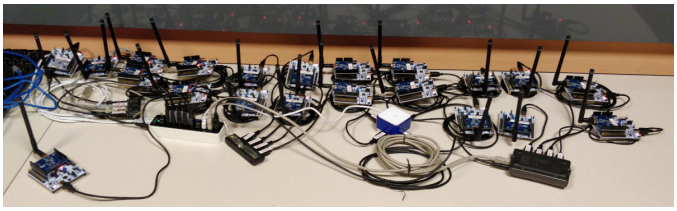
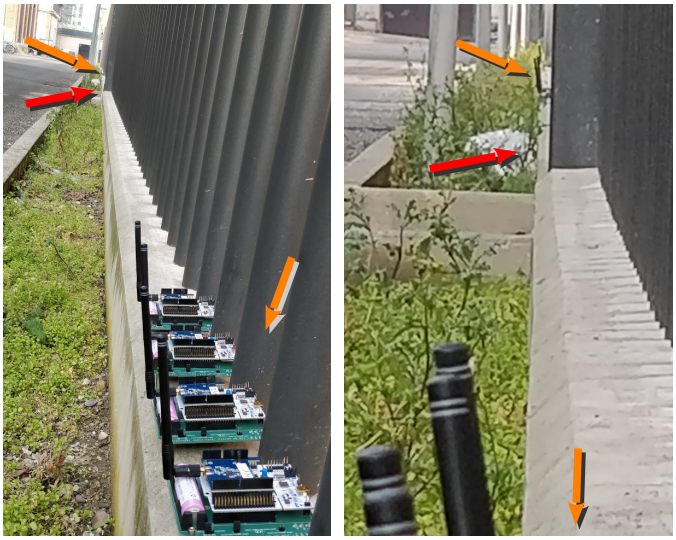


Fig. 26. Nodes in the lab test in close range to increase communication channel congestion, connected to a server for monitoring and debug.



(a) (b)

Fig. 27. Sensor nodes (e.g., top and bottom orange arrows) are grouped by couplings separated by a car length (≈ 20 m) along an undulated metallic fence with obstructions on the direct line of sight between sensor groups (e.g., middle, red arrow), and close to the ground. (a) normal view, (b) 4× zoom.

report is delivered to the CC and that it has the *unknown* code in the slot of the failed node (see Table II). Finally, we check the application and network stability over long runs, of several days, with the CC requesting train integrity reports every 5 s.

To test RF propagation and network operation, we install the network outdoors. We group the four coupling nodes together and separate them by an approximately train car length (20 m), along a metallic fence with protrusions that partially block the direct communication, as shown in Fig. 27. Node antennas are about 25 cm from the fence and 0.5 m from the ground, in conditions close to what we expect on train sides. The RF links were good between adjacent node groups (i.e., couplings), but spotty beyond that. We check again network operation and stability, which is not impaired by the reduced node connectivity. Also here we emulate the DS operation (see Section V).

Field tests allowed us to debug the application parts left out of the software simulations (e.g., drivers and interfaces), and to gain confidence that the software simulation results closely match the hardware operation.

IX. Comparison with the State-of-the-Art

Train integrity detection systems monitoring train brake pipes [22], [23] are considered too slow [3]. Our method is faster and predictable (≈3.6 s with 0.1 s SD for a 50-car train),

independent on train safety systems (brake pipes), and can locate the separations [8], [17].

Systems based on GNSS can degrade or fail in certain areas, GNSS nodes have relatively high energy consumption, may be incompatible with some train car types [3], can be slow detecting train separations (35–200 s [11]), may need track databases [20], connections with ground stations [9] with unbounded delays [62], or multiple GNSS systems coverages, cannot locate the separations, and need anyway a reliable communication between the train ends. Our method provides the end-to-end communication, is faster and more predictable ($\approx 3.6\,\mathrm{s}$ with $0.1\,\mathrm{s}$ SD for a 50-car train), independent of external infrastructures as required by the ERTMS Level 3, compatible with most car types, has lower energy requirements per node, and can locate the separations along the train [8], [17].

WSN protocols optimized for linear topologies focus on synchronization, data throughput, and latency, disregarding service reliability with faulty nodes or communications [28]–[30]. They use synchronous MACs, which incur network synchronization and rejoin overheads. Our system uses an asynchronous MAC with negligible node rejoin overhead (discussed in Section VII) and around $1\,\mathrm{mA}$ idle current consumption, lower than the $\approx 13\,\mathrm{mA}$ reported in [28]. In [34] are analyzed the effects of node faults and network size on monitoring coverage in dense WSNs, but without considering communication errors or network rejoin overheads. The effects of traffic distribution in prevalently unidirectional network are analyzed in [35], but uneven traffic load is a minor issue in our bidirectional flat hierarchy network, where only the nodes towards train ends may overhear and communicate fewer packets.

Hierarchical WSN protocols relying on multi-car radio communication ranges to improve reliability and energy consumption have been proposed [32], [36], [37], [39], [40], but they do not consider the RF communication variability, node and communication failures, latency constraints, or network rejoin overheads. Moreover, we avoid multi-car links (hence hierarchical clustering) that may often be unreliable. We improve network QoS through node, communication, and data redundancy, and optimize the WSN for both latency and energy.

A mixed network for train integrity monitoring using wired connections on cars and ZigBee wireless between them was proposed [41]. Like in our proposal, each coupling is monitored by four nodes and the locomotive query travels to the end of the train and back in less than two seconds on a four-car train. They do not provide node energy consumption, separation accuracy or latency, nor address operation reliability when nodes or communications fail, or consider network rejoin overheads. Wired networking on cars is more reliable but incur wiring costs. We address reliability and performance and cover a much longer 50-car train in $\approx 3.6\,\mathrm{s}$ average with $0.1\,\mathrm{s}$ SD. Other methods propose mixing GNSS, onboard IMU sensing, and RSSI from links based on the IEEE 802.15.4g standard for smart utility in star topology [42]. Current consumption is high, 50–$75\,\mathrm{mA}$, and are not discussed the reliability of long-range radio communications (see Section VI), network resilience to node and communication failures, reporting latency, or node compatibility with car types [3]. RSSI-based detection is used also in [6] using an adaptive synchronous MAC, which bears some overheads of synchronous MACs. Network topology assumes one node at each car end, communicating directly with up to three neighbors. The last node sends every second a message towards the locomotive, and train separation is inferred from lost RF communications. As they span more than three car lengths, the separation is likely detected beyond the TCAS specification of $30\,\mathrm{m}$ [11] and the $5\,\mathrm{s}$ latency requirement [6]–[8]. Node idle current is comparable ($\approx 1\,\mathrm{mA}$), but network energy optimization or resilience to node faults are not discussed.

X. Conclusion and future work

Along GNSS with IMU and brake pipe monitoring, WSNs are considered one of the enabling systems for onboard autonomous train integrity check, particularly for long freight trains without electrical power on cars. TCAS requires detecting train separations below $30\,\mathrm{m}$ wide [11], high-speed separations should be reported in less than $5\,\mathrm{s}$ [6]–[8], and CENELEC SIL 4 sets tight false positive limits, i.e., reporting a separated train as connected.

WSN parameters can be tuned to optimize the trade-off between the reporting latency and energy, based on application specifications and priorities. E.g., the integrity of a 50-car ($\approx 1\,\mathrm{km}$) train can be reported in $\approx 3.6\,\mathrm{s}$ average with $0.1\,\mathrm{s}$ SD. Asynchronous MACs reduce node network rejoin overhead, such as after being temporarily off due to insufficient energy harvested from the environment, albeit asynchronous MACs may increase the node idle consumption. Providing sufficient redundancy in network topology, sensing, and communication resources make it resilient to single points of failure and help to maintain a high QoS even in the case of multiple node and communication failures. A control center on the locomotive can control network operation and collect reports, simplifying system integration in railway and IoT applications.

Safe on-board train integrity monitoring is a complex problem, of which distributed monitoring of train couplings is only one part. The reliability and energy efficiency of this WSN prototype should be improved, the WSN extended to support train composition discovery, network formation, and security establishment. It should also be tested on trains, including its integration with the complementary techniques mentioned above, to reach the required safety and reliability levels.

Acknowledgment

Most of this work was developed for the ETALON (Energy harvesting for signalling and communication systems) project, grant agreement 777576, a Shift2Rail IP2 Open Call project of the Horizon 2020 Program of the European Commission.

References

[1] M. J. T. Lewis, "Railways in the Greek and Roman world," in *Early Railways: A Selection of Papers from the First International Early Railways Conference*. Newcomen Society, London, UK, January 2001, pp. 8–19.

[2] R. Bloomfield, "Fundamentals of European Rail Traffic Management System – ERTMS," *IET Conference Proceedings*, pp. 165–184, January 2006.

[3] (2018, August 31) Train Integrity Concept and Functional Requirements Specifications. Deliverable 4.1 X2Rail-2 EU Project, H2020. [Online]. Available: https://projects.shift2rail.org/download.aspx?id=d4965943-15b1-4977-9f32-ca44c947ca06

[4] O. Leveque, "ERTMS boosted: ERTMS with relative braking distance," in *AusRAIL PLUS 2012, Rail's Role in Society, 27-28 November 2012, Canberra, ACT, Australia*, 2012.

[5] (2020, August 10) Longer trains — Facts and experiences in Europe. Community of European Railway and Infrastructure Companies (CER) Working Group. 4th edition. [Online]. Available: http://www.cer.be/publications/latest-publications/longer-trains-facts-experiences-europe-0

[6] H. Scholten, R. Westenberg, and M. Schoemaker, "Trainspotting, a WSN-based train integrity system," in *Eighth International Conference on Networks*. IEEE, 2009, pp. 226–231.

[7] (2020, August 10) Moving Block System Specification. Siemens. Deliverable D5.1 of the Start-up activities for Advanced Signalling and Automation Systems project, EU H2020 GA No. 730640. [Online]. Available: https://projects.shift2rail.org/download.aspx?id=a81c93c2-36a5-46cf-8bd8-4924ae612dd7

[8] (2020, August 10) Train Integrity Methods Power Requirements and System Analysis. Siemens. Deliverable D3.4 of the Energy harvesting for signalling and communication systems Shift2Rail IP2 H2020 Open Call project, GA 777576. [Online]. Available: https://projects.shift2rail.org/download.aspx?id=a81c93c2-36a5-46cf-8bd8-4924ae612dd7

[9] S. Li, B. Cai, W. Shangguan, E. Schnieder, and F. G. Toro, "Switching LDS detection for GNSS-based train integrity monitoring system," *IET Intelligent Transport Systems*, vol. 11, no. 5, pp. 299–307, 2017.

[10] C. Tiberius and E. Verbree, "GNSS positioning accuracy and availability within Location Based Services: The advantages of combined GPS-Galileo positioning," in *2nd ESA/Estec workshop on Satellite Navigation User Equipment Technologies, GS Granados (Ed), ESA publications division, Noordwijk*. Citeseer, 2004, pp. 1–12.

[11] A. Acharya, S. Sadhu, and T. Ghoshal, "Train localization and parting detection using data fusion," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 1, pp. 75–84, 2011.

[12] Y. An, B.-G. Cai, B. Ning, J. Wang, and G.-W. Shang, "Research on train integrity monitoring method based on GPS and virtual-satellite," *Journal of the China railway society*, vol. 34, no. 9, pp. 40–44, 2012.

[13] (2020, April 7) Satellite Technology for Advanced Railway Signalling. STARS EU Project, H2020. [Online]. Available: https://cordis.europa.eu/project/id/687414

[14] (2020, April 16) 3TIMS - ERTMS Level 3 Train Integrity Monitoring Solution. [Online]. Available: https://business.esa.int/projects/3tims

[15] H. Scholten, R. Westenberg, and M. Schoemaker, "Sensing train integrity," in *Sensors Journal*. IEEE, 2009, pp. 669–674.

[16] H. Kirrmann and P. A. Zuber, "The IEC/IEEE train communication network," *IEEE Micro*, vol. 21, no. 2, pp. 81–92, 2001.

[17] S. Oh, Y. Yoon, K. Kim, and Y. Kim, "Design of train integrity monitoring system for radio based train control system," in *12th International Conference on Control, Automation and Systems*. IEEE, 2012, pp. 1237–1240.

[18] G. Shafiullah, A. Gyasi-Agyei, and P. Wolfs, "Survey of wireless communications applications in the railway industry," in *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*. IEEE, 2007, pp. 65–65.

[19] P. Fraga-Lamas, T. M. Fernández-Caramés, and L. Castedo, "Towards the Internet of smart trains: A review on industrial IoT-connected railways," *Sensors*, vol. 17, no. 6, p. 1457, 2017.

[20] A. Neri, F. Rispoli, P. Salvatori, and A. Vegni, "A train integrity solution based on GNSS double-difference approach," in *Proc. 27th Int. Technical Meeting of the Satellite Division of the Institute of Navigation*, 2014, pp. 34–50.

[21] A. Neri, F. Rispoli, and P. Salvatori, "An analytical assessment of a GNSS-based train integrity solution in typical ERTMS level 3 scenarios," in *Proc. Eur. Navigat. Conf. (ENC)*, April 2015.

[22] M. McClanachan, B. Payne *et al.*, "Improving Brake Propagation in Long Freight Trains," *CORE 2008: Rail; The Core of Integrated Transport*, p. 537, 2008.

[23] W. Wei and Y. Lin, "Simulation of a freight train brake system with 120 valves," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 223, no. 1, pp. 85–92, 2009.

[24] R. Sarfati, O. Eudes, and P. Tane. (2020, August 10) ETCS over GPRS/Edge capacity study in station environment (ERTMS objective 1 & 2). O-3300 of the Call CEF 2014-EU-TM-0279-S under the EU Connecting Europe Facility (CEF) program. [Online]. Available: https://uic.org/IMG/pdf/o-3300_cef-activity_12-etcs_over_gprs_edge_capacity_study_in_station_environment_v1.0.0.pdf

[25] I. Beretta, F. Rincon, N. Khaled, P. R. Grassi, V. Rana, and D. Atienza, "Design exploration of energy-performance trade-offs for wireless sensor networks," in *DAC Design Automation Conference 2012*. IEEE, 2012, pp. 1043–1048.

[26] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.

[27] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on emerging and selected topics in circuits and systems*, vol. 3, no. 1, pp. 45–54, 2013.

[28] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, 2007, pp. 254–263.

[29] T. Karveli, K. Voulgaris, M. Ghavami, and A. H. Aghvami, "DiS-MAC: A MAC protocol for sensor networks used for roadside and highway monitoring," in *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE, 2009, pp. 1–6.

[30] D. De Caneva, P. L. Montessoro *et al.*, "A synchronous and deterministic MAC protocol for wireless communications on linear topologies," *Int'l J. of Communications, Network and System Sciences*, vol. 3, no. 12, p. 925, 2010.

[31] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "Permasense: investigating permafrost with a wsn in the swiss alps," in *Proceedings of the 4th workshop on Embedded networked sensors*, 2007, pp. 8–12.

[32] R. Sokullu and E. Demir, "Investigating energy efficiency and timeliness for linear wireless sensor networks," *Procedia Computer Science*, vol. 37, pp. 24–31, 2014.

[33] P. Kumar, M. Gunes, Q. Mushtaq, and B. Blywis, "A real-time and energy-efficient MAC protocol for wireless sensor networks," in *2009 IFIP International Conference on Wireless and Optical Communications Networks*. IEEE, 2009, pp. 1–5.

[34] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Modeling the performance of faulty linear wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 7, p. 835473, 2014.

[35] M. Noori and M. Ardakani, "Characterizing the traffic distribution in linear wireless sensor networks," *IEEE Communications Letters*, vol. 12, no. 8, pp. 554–556, 2008.

[36] M. Zimmerling, W. Dargie, and J. M. Reason, "Localized power-aware routing in linear wireless sensor networks," in *Proceedings of the 2nd ACM international conference on Context-awareness for self-managing systems*, 2008, pp. 24–33.

[37] W. Nan, M. Qingfeng, Z. Bin, L. Tong, and M. Qinghai, "Research on linear wireless sensor networks used for online monitoring of rolling bearing in freight train," in *Journal of Physics: Conference Series*, vol. 305, no. 1. IOP Publishing, 2011, p. 012024.

[38] G. Lu, B. Krishnamachari, C. Raghavendra *et al.*, "An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks," in *Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2004.

[39] P. Mahasukhon, H. Sharif, M. Hempel, T. Zhou, and T. Ma, "Multi-tier multi-hop routing in large-scale wireless sensor networks for freight-train monitoring," in *ITS America 20th Annual Meeting & ExpositionITS America*, 2010.

[40] P. Mahasukhon, H. Sharif, M. Hempel, T. Zhou, T. Ma, and P. L. Shrestha, "A study on energy efficient multi-tier multi-hop wireless sensor networks for freight-train monitoring," in *2011 7th International Wireless Communications and Mobile Computing Conference*. IEEE, 2011, pp. 297–301.

[41] B. Allotta, P. D'Adamio, D. Faralli, S. Papini, and L. Pugi, "An innovative method of train integrity monitoring through wireless sensor network," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. IEEE, 2015, pp. 278–283.

[42] N. Barkovskis, A. Salmins, K. Ozols, M. A. M. García, and F. P. Ayuso, "WSN based on accelerometer, GPS and RSSI measurements for train integrity monitoring," in *4th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2017, pp. 662–667.

[43] Y. Pan, F. Liu, R. Jiang, Z. Tu, and L. Zuo, "Modeling and onboard test of an electromagnetic energy harvester for railway cars," *Applied Energy*, vol. 250, pp. 568–581, 2019.

[44] (2020, August 10) Transportation of cars by railways. Rail Wagons. U. A. B. EUROFIRE. [Online]. Available: http://www.eurofire.lt/en/rail-wagons

[45] J. Lu, G. Zhu, and C. Briso-Rodríguez, "Fading characteristics in the railway terrain cuttings," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2011, pp. 1–5.

[46] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.

[47] P. Tuset-Peiró, A. Anglès-Vazquez, J. López-Vicario, and X. Vilajosana-Guillén, "On the suitability of the 433 MHz band for M2M low-power wireless communications: propagation aspects," *Transactions on*

Emerging Telecommunications Technologies, vol. 25, no. 12, pp. 1154–1168, 2014.

[48] M. Kubaszek, J. Macheta, M. Zapart, Ł. Krzak, and C. Worek, "Evaluation of Antenna Diversity Scheme for 868MHz Narrowband Communication Systems," in *2018 International Conference on Signals and Electronic Systems (ICSES)*. IEEE, 2018, pp. 196–200.

[49] S. Tumanski, "Induction coil sensors—a review," *Measurement Science and Technology*, vol. 18, no. 3, pp. R31–R46, January 2007.

[50] J. Iqbal, M. T. Lazarescu, O. B. Tariq, and L. Lavagno, "Long range, high sensitivity, low noise capacitive sensor for tagless indoor human localization," in *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*. IEEE, 2017, pp. 189–194.

[51] D. E. Houssaini, A. Guesmi, S. Khriji, T. Keutel, K. Besbes, and O. Kanoun, "Experimental Investigation on Weather Changes Influences on Wireless Localization System," in *2019 IEEE International Symposium on Measurements Networking (M N)*, July 2019, pp. 1–6.

[52] (2020, August 10) STEVAL-FKI868V1. STMicroelectronics. [Online]. Available: https://www.st.com/en/evaluation-tools/steval-fki868v1.html

[53] (2020, August 10) STM32L152RE. STMicroelectronics. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32l152re.html

[54] C. Antonopoulos, A. Prayati, T. Stoyanova, C. Koulamas, and G. Papadopoulos, "Experimental evaluation of a WSN platform power consumption," in *2009 IEEE International Symposium on Parallel & Distributed Processing*. IEEE, 2009, pp. 1–8.

[55] M. T. Lazarescu, "Wireless sensor networks for the internet of things: Barriers and synergies," in *Components and Services for IoT Platforms*. Springer, 2017, pp. 155–186.

[56] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.

[57] (2020, August 10) FreeRTOS — Real Time Operating System for embedded systems. Amazon Web Services. [Online]. Available: https://www.freertos.org/

[58] (2020, August 18) Comparison of real-time operating systems. Wikipedia. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Comparison_of_real-time_operating_systems&oldid=970164473

[59] (2020, August 17) FreeRTOS kernel fundamentals. Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/freertos/latest/userguide/dev-guide-freertos-kernel.html

[60] (2020, August 18) FreeRTOS and SAFERTOS Comparison. WITTENSTEIN aerospace and simulation. [Online]. Available: https://www.highintegritysystems.com/safertos/upgrade-from-freertos-to-safertos/

[61] (2020, August 18) SAFERTOS, the safety certified RTOS. WITTENSTEIN aerospace and simulation. [Online]. Available: https://www.highintegritysystems.com/safertos/

[62] N. Zhang, X. Zhang, H. Liu, and D. Zhang, "Optimization scheme of forming linear WSN for safety monitoring in railway transportation," *International Journal of Computers Communications & Control*, vol. 9, no. 6, pp. 800–810, 2014.