Failure management strategies for IoT-based railways systems

Francesca Righetti, Carlo Vallati, Giuseppe Anastasi
Department of Information Engineering
University of Pisa
Pisa, Italy
name.surname@unipi.it

Giulio Masetti, Felicita Di Giandomenico
ISTI
CNR
Pisa, Italy
name.surname@isti.cnr.it

***Abstract*—Railways monitoring and control are currently performed by different heterogeneous vertical systems working in isolation without or with limited cooperation among them. Such configuration, widely adopted in practical deployments today, is in contrast with the integrated vision of systems that are at the foundation of the smart-city concept. In order to overcome the current fractured ecosystem that monitors and controls railways functionalities, the adoption of a novel integrated approach is mandatory to create an all-in-one railway system. To this aim, new IoT-based communication technologies, like wireless or Power Line Communication technologies, are considered the main enablers to integrate in a very rapid and easy manner existing vertical systems. In this work, we analyse the architecture of future railways systems based on a mix of wireless and Power Line Communication technologies. In our analysis, we aim at studying possible failure management strategies on rail-road switches to improve the level of reliability, crucial requirement for systems that demand maximum resiliency as they manage a critical function of the infrastructure. In particular, we propose a set of solutions aimed at detecting and handling network and sensor failures to ensure continuity in the execution of the basic control functions. The proposed approach is evaluated by means of simulations and demonstrated to be effective in ensuring a good level of performance even when failures occur.**

***Index Terms*—Smart Station, Smart City, added value services, failure recovery strategies**

I. Introduction

Security and reliability are crucial requirements for systems that monitor and control critical infrastructures like railways systems. In order to cope with external threats and minimise the possibility of malfunctioning, railways systems have been typically designed to work in isolation without - or with limited - interaction with other systems, internal or external. The result is that the train control and management environment are developed as vertical solutions, hence they do not exchange data or cooperate each other or with external entities.

This isolation is in contrast with the smart city concept. The smart city evolution envisions a city in which all its systems are closely entangled to interoperate and coordinate each other. The result is a close ecosystem of services that share data and information to offer better services to the general public [1]. Consequently, novel services can be created on top of existing ones. For instance, an omni-comprehensive travel companion application can be developed to help citizens move around the city offering mixed solutions (e.g. exploiting a bike sharing system together with bus transportation, and so on). Until now, railway stations have been excluded from this ecosystem, as they are mainly working in isolation without interacting with other public systems, mainly due to security concerns.

As the smart city ecosystem expands, the need to evolve current train stations into smart railway systems arises. The application of the most recent technologies from the communication fields is expected to foster the evolution of such systems through the Internet of Things (IoT) paradigm [2]. IoT Communication technologies such as wireless communication standards or Power Line Communication (PLC) are widely adopted today in industrial environments [3]. Considering that industrial applications are characterised by the same strict requirements in terms of security and reliability of railways systems, the adoption of those recent wireless and PLC technologies is expected to be now mature to implement the IoT paradigm on a large scale and to replace current wired based communication solutions in the railways sector.

In this context, the STINGRAY project is carrying out a set of activities that aims at renewing the railway station system, by applying Industrial IoT technologies. One of the core activity of the project is to design a horizontal monitoring and control system in which existing vertical solutions can be easily integrated. In order to ease such integration in a rapid manner, and with minimal additional costs, a mix of wireless technologies and PLC systems are employed to interconnect all the train station entities into a single communication infrastructure.

The reliability of such integrated infrastructure is considered paramount to ensure continuity of crucial services. For this reason, it is important to study the impact of potential failures in both the communication infrastructure and the sensors, to implement solutions that aim at mitigating the impact of failures in the effectiveness of the monitoring and the control services. In this work, we analyse the architecture of future IoT-based railway systems in order to study failure management strategies. Specifically, we focus on one crucial system, the rail-road switch heating control system, by studying failure detection and mitigation strategies considering two important components: (i) the communication infrastructure and (ii) the environment sensors, such as the temperature sensors, whose data is exploited to regulate the heating system for the rail-road switches. In our work, we first study a solution to detect failures in the communication network, and react by

introducing a distributed control mode that is automatically activated to ensure service continuity. Subsequently, we propose a strategy to analyse the data collected by temperature sensors and detect and react to malfunctions in the temperature sensors via outlier detection. The latter strategy is evaluated by means of simulations and it is demonstrated to be effective in mitigating the effects of the failure of the temperature sensors.

The paper is organised as follows: in Section II we present the project STINGRAY with its goals and main activities, in Section III the general architecture of the rail-road switch heating control infrastructure is presented. In Section IV we introduce the proposed management strategies to handle failures in the communication infrastructure, in Section V we present the results of the performance evaluation of the proposed strategies, while in Section VI we draw our conclusions.

II. The STINGRAY project

STINGRAY (https://stingray.isti.cnr.it/) is a project co-funded by the Tuscany Region under the POR FESR Toscana 2014-2020 program. It started on July 2018 and its duration is 24 months. The project involves four industrial partners, i.e. ECM S.P.A. as project leader, DMG Engineering S.R.L., ELFI S.R.L., C.T. Elettronica S.R.L. and two academic organisations, ISTI-CNR jointly with the University of Pisa and the University of Florence. The objective of STINGRAY is to renew the role of the railway station, traditionally seen as a meeting point for a city, in order to enhance its importance and integration in the Smart City of the future. Railway stations are a primary point of aggregation in every urban centre, but traditionally they are isolated from the urban context. They have a private energy distribution and communication system, mainly to prevent blackouts and unauthorised intrusions. However, we emphasise two major drawbacks of such isolation. First, there is no integration with the so-called Smart Cities, where information between different transport systems (i.e. bike sharing, car-sharing, urban transport) is meant to be synergically exploited. Second, the station system is excluded from modern techniques of energy saving and failure management.

STINGRAY is focused on these challenges, which are meant to be addressed by the study, design and development of an integrated monitoring and control infrastructure. Such horizontal system aims at integrating all the existing vertical solutions into a single system through a single communication infrastructure, integrating PLC and wireless technologies. Specifically, the goals of the STINGRAY project are the following:

1) develop a local network over the station plants using PLC and wireless technologies;
2) create a single control and monitoring system for the station equipment;
3) create, on top of this system, added value services for both customers and railway staff, also tailored to integrating the station system within the Smart City.

STINGRAY represents an opportunity for a more efficient management of machinery and energy resources. It promotes a technological innovation, obtained without building new expensive infrastructures, thanks to the exploitation of power lines, and results in a lower environmental impact.

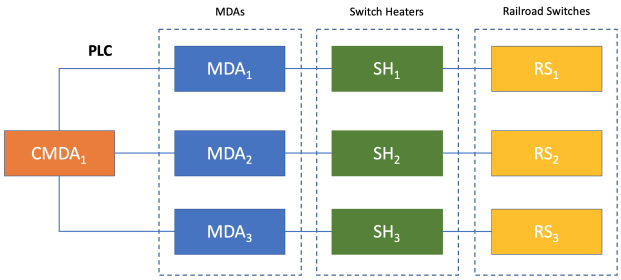


Fig. 1: Rail-road switches control architecture

III. Rail-road Switch Heating Control System

Rail-road switches are mechanical devices that allow railway trains to be moved from one track to another. The switches play a crucial role in operating the railway system, as the routing of trains strongly depends on their correct operation. The proper working of such switches is crucial, as malfunctions can cause train derailments or train collisions, with catastrophic consequences for passengers.

One of the main cause for rail-road switches malfunctioning is the low temperature, along with snow and ice. In the past, it was common to have workers specifically employed by railway companies to keep the switches clear by sweeping the snow away. Recently, manual workers have been substituted by automatic heaters installed in the proximity of the switches. Such heaters are operated automatically to keep the temperature in the surroundings of each switches above the freezing point. Heaters may be powered by gas, water circulation, steam or electricity. Electric heaters are mainly used worldwide today, consequently we concentrate on them in this paper.

In the Italian railway system considered in the Stingray project, heaters are operated remotely following a policy that switches on/off the heater based on the current conditions of the surrounding environment, e.g. temperature and humidity. The overall architecture of the control system, implemented on a single railway station, is illustrated in Fig. 1. The overall system that controls all the switches of a railway station adopts a hierarchical structure with two different modules at different levels for control/decision making: (i) one module for each rail-road switch, named MDA (Module for Data Acquisition), that locally controls the switch through a direct connection, and (ii) one centralised module, named CMDA (Coordinator MDA), that is responsible for supervising and coordinating the operations of all MDAs, or a subset of them in large stations that employ more than one CMDA.

In normal operation conditions, each switch heater SH_i is controlled by its locally connected MDA_i. The latter is equipped with a local temperature sensor and, before activate/deactivate its corresponding switch heater, sends the measured temperature T_i every Δt minutes to the CMDA and waits until the CMDA responds with an *on*/*off* message. The CMDA compares the received T_i with a threshold value T^{thr} that is chosen taking into account additional information

provided by other systems. For instance, by considering the dew point based on the current air relative humidity, or by retrieving data from the weather forecast, e.g., the probability of snow falls. The fine-grained tuning of the T^{thr} performed by the CMDA is important to reduce the energy consumption of the overall system, as it guarantees that the switch heaters are turned on only when strictly necessary.

Actions are taken every Δt minutes in order to guarantee hysteresis, and the value of Δt is selected according to estimates of the temperature change velocity of a specific region.

The CMDA communicates with the assigned MDAs mainly through PLC or, in small scale installations, via wireless links. PLC is very popular since it exploits for communication the electric network that is already installed for power distribution, thus reducing significantly the installation costs. Both the technologies, however, are characterised by network outages. Wireless technologies due to interference on the communication frequency, whereas PLC due to the significant reduction in the communication reliability that can frequently occur with high level of humidity or low temperatures.

When the communication between the CMDA and the MDAs is interrupted, each MDA switches to a different policy: it compares the value of T_i with a default threshold, e.g., $5°$C, chosen to guarantee that, in the selected region, the drop of T_i from the threshold to $0°$C in less than Δt minutes is extremely unlikely. When the communication is re-established, the MDAs switch back to the normal policy.

IV. Failure Management Strategies

As highlighted in Section III, long-lasting outages in the communication infrastructure can lead to malfunctions. Network outages, however, are not the only source for the system misbehaviour. The accuracy of the temperature sensor installed in each MDA is crucial in order to ensure the reliability of the heating control operations. Sensors, however, can reduce their accuracy or eventually broke over time. For this reason, in order to ensure the proper functioning of the whole system, it is crucial to monitor the data from the temperature sensors to spot in a timed manner temperature sensors that provide inaccurate readings.

In order to mitigate the effects of both network outages and temperature sensor inaccuracy, specific failure management strategies can be introduced with the goal of reducing the inaccuracy of the operations performed by each MDA and quickly detect faulty sensors and exclude the usage of the information provided by them. In the following we present two different failure management strategies that aim at handling such failures: in Section IV-A, we present a failure management strategy to mitigate the effects of long-lasting outages of the network infrastructure, while in Section IV-B, we present a strategy to identify malfunctioning of the temperature sensors and properly react.

A. Communication Infrastructure

The most popular technology adopted for the communication among MDAs and the assigned CMDA is PLC. PLC networks are known to have significant variations of the communication channel quality over time, which might result in periods in which the network is partitioned.

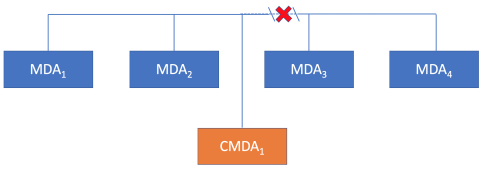


Fig. 2: Network failure example.

Fig. 2 illustrates an example of possible network fault that results in a network partition. In this example we assume that a failure in the power link connecting MDA_3 with $CMDA_1$ occurs, thus resulting in a network partition. Specifically, MDA_1 and MDA_2 continue to communicate with the $CMDA_1$ and therefore can continue to operate normally; MDA_3 and MDA_4, instead, remain isolated. Although isolated, the communication between them can take place and the exchange of data among them can still occur.

The possibility of communication between MDAs has not been exploited so far, as all the functionalities are implemented on the CMDA. In this work, we propose to exploit the local communication between MDAs, which can be still available during network failures, to implement distributed coordination functionalities among them. This local communication can be exploited to compensate the impossibility to communicate with the CMDA. In details, we propose an approach to: (1) detect network failures and partitions; (2) elect a local coordinator among the MDAs that can still communicate inside the network partition through a distributed election procedure; (3) let the local coordinator sets the T_i values of the MDAs to a safe values to avoid the formation of ice, and performs other security operation, e.g. the temperature sensor malfunctioning detection procedure described in Section IV-B.

In the following we describe in details the network failure detection procedure (Section IV-A1) and the local coordinator election algorithm (Section IV-A2).

1) Network failure detection: In order to detect failures in the communication network and consequently trigger the election of a temporary coordinator, the CMDA sends a *beacon* message to each MDAs available with period P. Each MDA replies with an *acknowledgement* message, to report the reception of the message and that it is working properly.

Each MDA has a timeout timer that is set to $3 \cdot P$. Every time a beacon message is received the timer is reset. If the timer expires, i.e., three consecutive beacon messages from the CMDA are lost, the MDA assumes that a network failure occurred and it is not possible to communicate with the CMDA. Consequently the process for the election of the local coordinator is triggered.

After the election, the local coordinator itself sends the beacon messages to detect further partitions in the network. Whenever a beacon message from the CMDA is received, the normal operations are resumed and the functionalities of the local coordinator are disabled.

2) Local coordinator election: Whenever a network partition is detected, an election procedure is triggered to select a local coordinator among the MDAs of the partition. The election procedure proposed in this work is inspired by the Paxos algorithm [4], a popular consensus algorithm widely used in many distributed systems. The election procedure aims at selecting the MDA with the highest priority among the ones available in the partition. For simplicity, it is assumed that each MDA has an unique ID and that the CMDA role is assigned to the MDA with the lowest ID, i.e., $ID = 0$. With this assumptions, the ID can be adopted as priority index for the election. It is worth nothing that a different index (e.g., the address) could be adopted to implement the election procedure with minimal modifications.

An example of the procedure is illustrated in Fig. 3. The left side of the figure illustrates the election procedure when messages are all correctly transmitted, the right side the procedure when some messages are dropped. The election is organised in rounds of fixed length. Each round is divided into two phases, a discovery phase first, second a candidate/confirmation phase. In the discovery phase, each MDA broadcasts a *discovery message* with its ID. In order to avoid collisions, a random backoff is introduced before transmitting the message. Every node that receives the discovery message from other MDAs, compares the received IDs with its own and updates the ID of the local coordinator candidate (i.e., the lowest ID received). In the candidate/confirmation phase, instead, each MDA sends out a *candidate message* with its own ID and the ID of its current local coordinator candidate. The other MDAs process this message and update their local coordinator candidate if needed. Each MDA terminates the election if in the current round the candidate local coordinator was not modified. When for a round the candidate local coordinator is not changed, the election procedure is considered terminated by an MDA. If an MDA is selected as candidate local coordinator it sends a *confirmation message* to the others, so the procedure can terminate for all.

After confirmation, the elected local coordinator starts to send out beacon messages in a periodic manner, to allow the MDAs of the partition to detect and react to further network outages. If a beacon message from the CMDA, instead, is received, the local coordinator stops operating as CMDA and the regular operations coordinated by the CMDA are resumed.

B. Temperature Sensors

As mentioned, the reliability on the accuracy of the measurements provided by the temperature sensors of MDAs is paramount. Indeed, the CMDA's logic for producing *on/off* messages is based on $\{T_i\}_i$ and external information. For this reason, in the following we propose an approach to detect possible faults in the sensors and to react to them.

The fault of a sensor is detected by comparing temperature readings from different MDAs. When a temperature sensor is not working properly, the readings can be significantly different from the values generated from other sensors in the same area. In particular, the faulty sensor usually reports

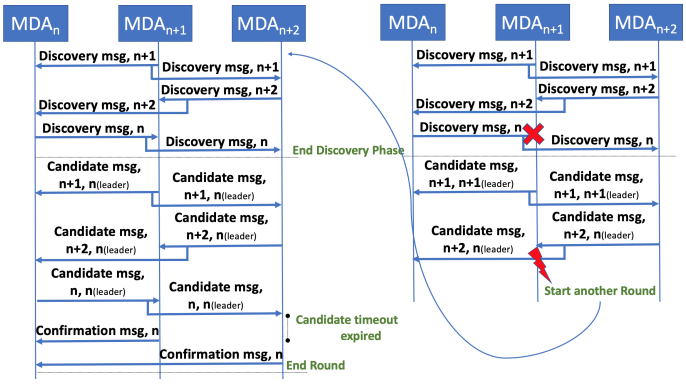


Fig. 3: Election procedure - message exchange example.

values that are excessively low or high in comparison with the ones obtained from correctly functioning sensors. This can be the consequence of calibration issues, which generate samples with a significant offset with respect to the actual values, or hardware issues, which can result in randomly generated readings.

In our approach, we propose to detect faulty sensors by comparing the sequence of samples collected from different MDAs. Specifically, the CMDA periodically receives temperature sensors from all the MDAs and compares the most recent values looking for outliers, i.e., a temperature value from an MDA that is significantly different from the others. A sensor that reports outlier values for a certain period is consequently marked as faulty. In order to detect outliers, the Dixon test [5], a specific test that can be exploited to highlight outliers on a statistic population, is used.

When a temperature sensor is marked as faulty, the CMDA notifies the railway personnel for replacement. In the meantime, however, a compensation strategy is put in place to ensure the proper management of the switch heater to minimise any potential safety risks. Specifically, the CMDA instructs the MDA to ignore the readings from the temperature sensor and use as current temperature $\tilde{T}_i = \frac{1}{n_{\text{ctrl}}-1} \sum_{j \neq i} T_j$, where ctrl is the number of MDAs controlled by the CMDA. In order to ensure that the faulty temperature sensors are detected also during network outages, such functionalities could be temporarily implemented by the local coordinator once elected, where $\tilde{T}_i = \frac{1}{|\Pi|-1} \sum_{j \in \Pi \setminus \{i\}} T_j$ and Π is the partition.

V. Performance Evaluation

A. Leader Election Procedure Evaluation

In order to evaluate the efficiency of the leader election procedure, a set of simulations has been carried out. To this aim, the proposed procedure has been implemented on an ad-hoc simulator written in Python using the library SimPy [1], a popular Python library that allows, in a rapid manner, to develop a discrete-event based simulator. The simulator has been written to simulate a network of N MDAs connected to a single CMDA in which a network outage occurs, thus forming a partition in the network, i.e., a subset of isolated

[1] SimPy Website https://simpy.readthedocs.io/en/latest/

MDAs unable to communicate with the CMDA but that can communicate each other.

The goal of our simulations is to measure the performance of the proposed leader election procedure, triggered right after the network outage. To this aim, the metrics we measured are the following:

1) *Election completion time*, defined as the time between the beginning of the network outage and the termination of the election procedure with the last MDA confirming the new local coordinator.
2) *Number of messages*, defined as the overall number of messages exchanged by the isolated MDAs during the execution of the local election procedure.

The simulation is configured to trigger the network outage at a time t that is randomly selected with a uniform distribution between 10s and 2.5h, while the election timeout is set to 20s. The network outage we simulated, isolates 30% of the MDAs in the network every time a partition occurs. An increasing number of MDAs from 10 to 50 is considered, while an increasing value of the packet loss probability P_{loss} is adopted to consider different channel conditions in the PLC communication. In order to obtain statistically sound results, each scenario is replicated 1000 times using a different seed. In the graphs, the average value of the metrics is reported along with its 95% confidence intervals.

In Fig. 4 we report the average election completion time with an increasing value of the number of MDAs N and different values of the packet loss P_{loss}. As expected, increasing values of P_{loss} result in increasing completion time. This can be explained with the fact that the loss of a message can delay the completion of the procedure for at least one MDA, since it can require another round of messages to obtain the proper information on the candidate leader. On the other hand, as N increases the election completion time decreases. This improvement is more noticeable when a higher loss probability is considered (e.g., 0.7, 0.8, 0.9). This can be explained considering that a larger N results in a larger number of isolated MDAs; a higher number of isolated MDAs, generated more replicas of the messages transmitted during the election procedure thus compensating for messages lost during transmission.

In Fig 5, instead, we report the average number of messages exchanged during the election procedure by the isolated nodes. As expected, the number of messages increases with N as more isolated nodes transmit more messages. In a similar manner, as the P_{loss} increases, the number of exchanged messages increases as well. This can be explained with the additional time required to complete the election in which additional messages are transmitted.

The results of our simulations highlight that the proposed election algorithm can successfully terminate in an acceptable amount of time, always below 350s in our simulations. The time for completion, however, is dependent on the number of MDAs installed in the system and, more noticeably, on the P_{loss} of the communication medium. For instance, if a rapid convergence is needed in all the configuration, e.g. below 120s, a $P_{loss} < 0.5$ is required.

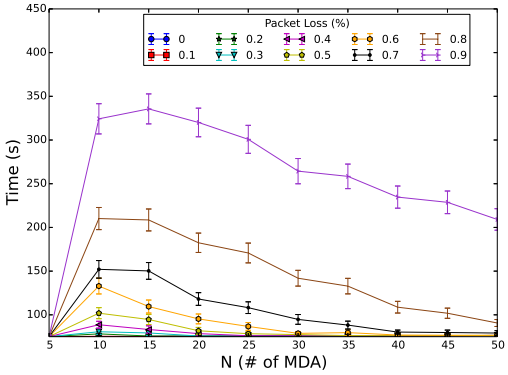


Fig. 4: Time required to complete the election procedure.

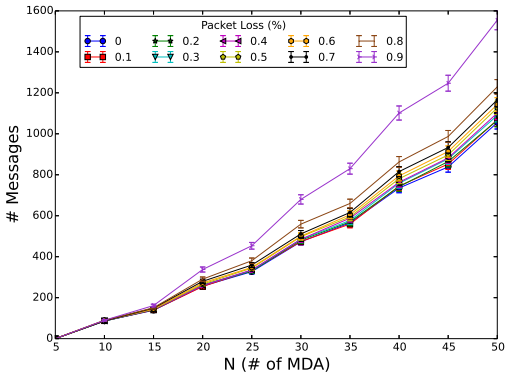
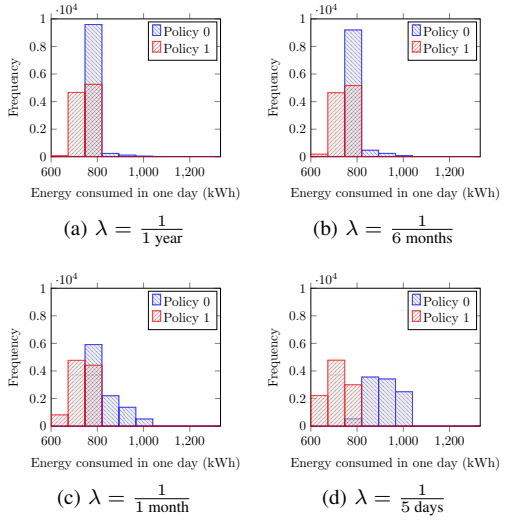


Fig. 5: Number of messages exchanged during the election procedure.

B. Temperature Sensor Failure Mechanism Evaluation

In order to evaluate the performance of the mechanism to detect and react to temperature sensor failures, we run another set of simulations. To this aim, a model of the railroad switch heating control system has been created on Möbius [6], a popular simulation engine tool. Specifically, the model



(a) $\lambda = \frac{1}{1 \text{ year}}$ (b) $\lambda = \frac{1}{6 \text{ months}}$

(c) $\lambda = \frac{1}{1 \text{ month}}$ (d) $\lambda = \frac{1}{5 \text{ days}}$

Fig. 6: Frequency distributions of the energy consumption (kWh) in one day at increasing of λ.

of the system has been expressed through Stochastic Activity Networks (SANs) [7], which are an extension of the Petri Nets formalism, based on the following primitives: plain and extended places, timed and instantaneous activities, input and output gates. The SAN primitives are defined by expressions or statements of the C++ programming language.

Through SAN, a general model of the system has been implemented, in addition to the logic of the proposed algorithm to detect sensor failures and to react to them. In order to ensure the reliability of the simulations, a set of real temperature traces has been exploited. Specifically, we employed a set of traces obtained from environmental sensors deployed in the city of Pisa between 2016 and 2019. Considering that we aim at evaluating the effectiveness of the temperature sensor failure mechanism, in our model we do not consider faults in the communication network.

The key performance indicator selected to measure the impact of the sensor failures in the system, and the potential benefit of the proposed approach, is the overall energy consumption in one day expressed in kWh. It is defined as the sum of the energy consumed by all the switch heaters in the system over one day, where each heater consumes 7.4 kW when on, and zero kW when off. This metric is expected to provide an overall indication on how the failure of a sensor can increase the overall power consumption, and how the proposed approach can be effective in mitigating the effect of the failure.

Each simulation analyses 24h and includes one CMDA and 18 MDAs. During each experiment, the failure of 1 temperature sensor is simulated. The failure occurs at a certain instant in the simulation that is modelled through an exponential random variable with average $1/\lambda$. After the failure is triggered, the sensor reports a fixed temperature value off-the-scale below zero that systematically triggers the switch heaters on. The values of λ are selected among: one over a year, one over six months, one over a month and one over 5 days. For each assignment, 10000 independent simulation batches have been executed in order to obtain statistically sound results.

In Fig. 6, the distribution of the energy consumption over the different independent replicas of the experiment is reported for the different configurations considered. The results obtained with the proposed detection procedure (*Policy 1*) are compared to the system default behaviour, with no sensors malfunctioning detection enabled (*Policy 0*).

As can be seen, in all the considered configurations, the proposed policy can guarantee less energy consumption than the default policy. This is the result of the fact that the proposed policy allows to identify faulty temperature sensors, whose misleading readings lead to unnecessary turn the heaters on for a long time. In the proposed policy, instead, the CMDA can rapidly spot faulty sensors and instruct the corresponding MDAs to ignore the readings and use the average value provided. Such value of the temperature is not expected to be accurate, however, it can reduce the likelihood of periods in which heaters are turned on unnecessary.

If we compare the results obtained with different values of λ we can notice that a higher average period before the fault of the sensors, results in a higher benefit for the proposed policy. Specifically, a lower λ results in a higher average period before sensor failure, thus increasing the benefit in highlighting the fault and adopt a mitigation strategy in terms of lower energy consumption.

In Figs. 6a and 6b, even though the mean energy consumed by *Policy 0* and *Policy 1* are close, the distributions are different, so the environmental impacts are different. In addition, being electricity price a piece-wise constant function of kWh, this means that a difference in distribution can lead to none or significant saving.

VI. Conclusions

In this paper, we analysed the architecture of future IoT-based railways systems in order to improve their reliability and security. To this aim, two different solutions to handle network and sensor failure in the control system of the switch heaters are proposed. Firstly, we proposed a solution to detect failures in the communication network and to react by introducing a distributed mode based on the election of a local coordinator. Secondly, we defined a mechanism to detect and react to the failure of the temperature sensors deployed on the local controllers. Both the mechanisms are evaluated by means of simulations and demonstrated to be effective in improving the reliability of the system.

As future work, we plan to extend the performance evaluation of the proposed mechanisms by considering different settings, such as a varying ratio of the isolated MDAs and different models both for the failure probability of the communication channel and for its duration.

Acknowledgments

This work was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the Crosslab project (departments of excellence) and by the Tuscany Region under the POR FESR Toscana 2014-2020 program in the framework of the STINGRAY project.

The authors would like to thank Filippo Guggino for his invaluable help in the definition of the procedures and the execution of the simulations.

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, 2014.

[2] P. Fraga-Lamas, T. M. Fernández-Caramés, and L. Castedo, "Towards the internet of smart trains: A review on industrial iot-connected railways," *Sensors*, 2017.

[3] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, 2014.

[4] L. Lamport, "The part-time parliament," in *Concurrency: the Works of Leslie Lamport*, 2019.

[5] R. B. Dean and W. J. Dixon, *Simplified Statistics for Small Numbers of Observations*. Anal. Chem., 1951., 1951.

[6] T. Courtney, S. Gaonkar, K. Keefe, E. W. Rozier, and W. H. Sanders, "Möbius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models," in *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*. IEEE, 2009.

[7] W. H. Sanders and J. F. Meyer, "Stochastic activity networks: Formal definitions and concepts," in *School organized by the European Educational Forum*. Springer, 2000.