Document downloaded from:

http://hdl.handle.net/10251/181293

This paper must be cited as:

Vera-Pérez, J.; Todoli Ferrandis, D.; Sempere Paya, VM.; Ponce Tortajada, R.; Mujica, G.; Portilla, J. (2019). Safety and Security oriented design for reliable Industrial IoT applications based on WSNs. IEEE. 1774-1781. https://doi.org/10.1109/ETFA.2019.8869204

The final publication is available at

https://doi.org/10.1109/ETFA.2019.8869204

Copyright IEEE

Additional Information

Safety and Security oriented design for reliable Industrial IoT applications based on WSNs

José Vera-Perez
Instituto Tecnológico de Informática
Valencia, Spain
jvera@iti.es

David Todoli-Ferrandis
Instituto Tecnológico de Informática
Valencia, Spain
dtodoli@iti.es

Víctor Sempere-Payá
Universidad Politécnica de Valencia
Valencia, Spain
vsempere@dcom.upv.es

Rubén Ponce-Tortajada
Instituto Tecnológico de Informática
Valencia, Spain
rponce@iti.es

Gabriel Mujica
C Centro de Electrónica Industrial
Madrid, Spain
gabriel.mujica@upm.es

Jorge Portilla
C Centro de Electrónica Industrial
Madrid, Spain
jorge.portilla@upm.es

Abstract—**Internet of Things based technologies are enabling the digital transformation in many sectors. However, in order to use this type of solutions, such as wireless sensor networks, in scenarios like transport, industry or smart cities, the deployed networks must meet sensible safety and security requirements. This article describes a Wireless Sensor Network design that applies multi-layered mechanisms and tools to ensure security, safety and reliability while maintaining usability in Rail and Industrial IoT scenarios. The proposed solution provides guidelines for choosing the best implementations given usual restrictions, offering a modular stack so it can be combined with other solutions.**

Keywords—WSN, Safety and security, IoT, IEEE 802.15.4e, SCOTT

I. INTRODUCTION

The Internet of Things (IoT) is one of the key enablers of the digital transformation, and it is already extended in many real-world applications. From smartphones and tablets to vehicles, intelligent systems that connect these devices can be found across the world. This has brought various benefits to society and industry, which require more and more connectivity every day.

One of the responses to improve this connectivity and digitalization new necessities are the Wireless Sensor Networks (WSN), which number of deployments is rising for their reduced time and cost of installation, and their ability to bring together Operational Technologies (OT) and Information Technologies (IT) worlds together in industrial scenarios. Nevertheless, this popularity has arisen important concerns about security and privacy when using this type of technology and devices. Especially when talking about critical applications in industry and transport, where safety and security requirements are stricter, often with mandatory regulations.

As attention turns towards security vulnerabilities and safety breaches, several efforts and initiatives, such as the SCOTT (Secure Connected Trustable Things) project (https://scottproject.eu/), aim to build trust in wireless solutions by providing end-to-end secure, trustworthy and dependable connectivity and interoperability communications.

To address these critical requirements, this article presents a layered safety and security design, proposing an end-to-end protocol stack for operating a WSN in safe and secured applications.

This solution is presented in next sections as follows: section 2 introduces relevant related work. Section 3 proposes an end-to-end protocol stack as a base to build Industrial IoT secure applications. Section 4 presents implemented security mechanisms for WSN at each layer of the selected stack, whilst Section 5 describes safety related approaches. The paper ends with conclusions, acknowledgments, and references.

II. RELATED WORK

Security has always been necessary in every type of communication system. During the last few years, as WSNs have gained popularity, many researchers have turned their attention to analysing specific issues with this kind of wireless networks, so there are many surveys available in related literature.

In [1] and [2], authors provide layer-by-layer types of attacks and possible approaches for protection, analysing vulnerabilities, risk assessments, security requirements and solutions. Another point of view is shown in [3], where authors propose a model that makes use of a Software-Defined Networking (SDN) approach to manage a WSN in order to solve most of the security issues. Finally, [4] describes the impact of security mechanisms implementation in energy consumption for WSNs devices, which is another interesting and key issue, especially when limitations of the available energy have to be taken into account. These works show that IIoT faces many threats that are often not taken into account, and that this issue can be engaged following different policies. The security mechanisms to address common threats vary depending on the technology selected, but also on the environment and the application requirements. There is no "one solution fits all".

The following description of a secure design and implementation of a WSN aims to address several of the issues described in the aforementioned works, by using some of the proposed solutions, which are dependent on the technology and protocols chosen, and other tools to provide a robust, secure and operational WSN meeting industrial grade requirements.

III. End-to-End Protocol Stacks

Designing an end-to-end secure solution comprises the selection and configuration of the protocol stack running in each element of the application's architecture. In this case, the involved elements are the WSN sensor nodes, the Gateways for WSNs, the core data bus implemented by a MQTT broker (Message Queuing Telemetry Transport [5]), and the remote clients that subscribe to sensor data and set configurations for the deployed devices.

As can be seen in Fig.1, two main choices define the possible security mechanisms to be implemented. On one hand, the MAC (Medium Access Control) layer and network protocols for the WSN nodes, which are IEEE802.15.4e and RPL (IPv6 routing protocol for Low Power and Lossy Networks), which represent the wireless part susceptible of unauthorized access and manipulation of data. On the other hand, the application layer protocol, with publish/subscribe MQTT as selected solution.

Another proposed candidate for the application layer is OPC-UA (OPC Unified Architecture) [6], a standard that is becoming popular because its relationship and integration with the industry classic OPC (Object Linking and Embedding for Process Control) protocol that is widely spread as a solution for connecting legacy devices such as automatons or distributed control systems (DCS). OPC-UA includes several security mechanisms comprising transport and application layer, but from now on, in the presented design, MQTT is selected as the candidate for IIoT communications due to its higher scalability and flexibility, seamless integration, low complexity and small footprint and need of resources.

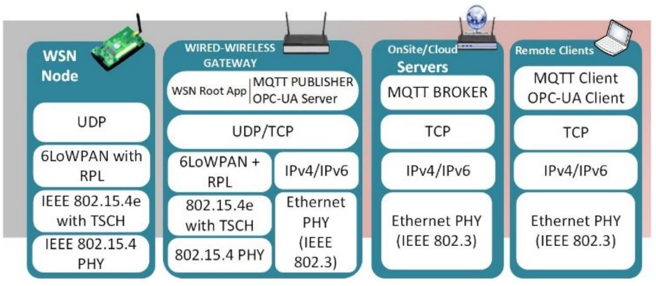

Fig. 1. End to end protocol stack for each participating device.

IV. Security Mechanisms for WSN

From a security point of view, mesh or multi-hop WSNs are not different to any other wireless network. These networks are vulnerable to passive espionage attacks and active manipulation since physical access to the cable is not required to participate in the communication. In addition, the very nature of ad hoc networks imposes some additional security constraints, which may make such networks more difficult to protect.

These devices are usually low cost and have capacity limitations in terms of computing power, available storage, and power consumption. In addition, network nodes establish infrastructure-less communications based on short-term relationships between devices that may have never communicated with each other before.

All these factors are quite limiting when choosing cryptographic algorithms and security protocols, as well as influencing when defining the security architecture, since the establishment and maintenance of communications between devices must be carried out with care.

There are solutions in the market based on dedicated cryptochips with enhanced electrical protection for key storage and cryptography issues [7]. They are prepared for side channel attack protection and manipulation avoidance. They are prepared to create a root of trust to ensure security in the embedded systems.

A. MAC layer with IEEE802.15.4e

WSNs based on IEEE 802.15.4e define different levels of encryption that can be used at different MAC layer, including the medium access method TSCH (Time Slotted Channel Hopping). The level of encryption can be configured using two bits that will be included in the "Security Level" field within the IEEE 802.15.4 frames. Currently, AES (Advanced Encryption Standard) encryption mechanisms [8] are supported, using the CBC-MAC (cipher block chaining message authentication code) technique for constructing message authentication codes.

Table I shows the different levels of encryption for IEEE 802.15.4, and the extra bytes overhead that each level adds to the communication, and Fig.2 shows how the different MAC frames are build according to the level selected.

In addition, by using Contiki OS to develop nodes' firmware it is possible to define a level of encryption for each type of message individually. This provides the option to assign a level of security to beaconing messages, another level of security for recognition messages (ACK) and yet a different level of security for the rest of messages.

In this way, it is possible to create a greater combination of wireless networks of secure sensors, being able to add security only to those frames with a particular interest for protection. In fact, when using TSCH MAC access method, securing beacon messages, which are needed to synchronize and join the network, ensures that only valid nodes with correct keys are able to join the network. Moreover, in order to avoid eavesdropping, the rest of messages must also be encrypted.

Given the limitations often found in terms of data payload, selecting the best type of MAC security is limited by the maximum size of data expected in each message, taking into account that the IEEE802.15.4 payload has a maximum size of 127 bytes.

TABLE I. Security Types In MAC layer

Extra bytes	Type	Description
0 B	No security	Data is not encrypted or authenticated
6 B	AES-CBC-MAC-32	Data is authenticated but not encrypted
10 B	AES-CBC-MAC-64	Data is authenticated but not encrypted
18 B	AES-CBC-MAC-128	Data is authenticated but not encrypted
4 B	AES-CTR	Data is encrypted but not authenticated
6 B	AES-CCM-32	Data is authenticated and encrypted
10 B	AES-CCM-64	Data is authenticated and encrypted
18 B	AES-CCM-128	Data is authenticated and encrypted

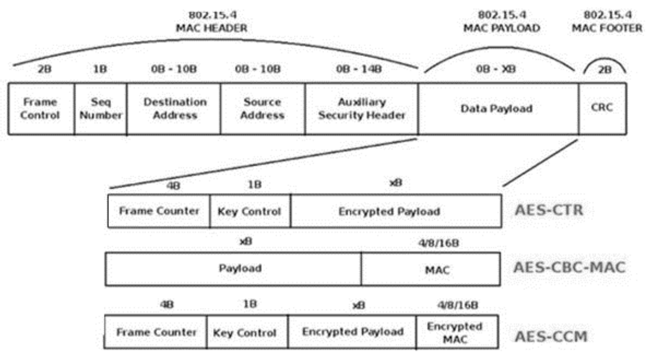

Fig. 2. IEEE802.15.4 frame for different security implementations from (www.sensor-networks.org)

B. Network Layer with RPL – IP layer

This type of ad hoc networks allows connections to be established with neighbouring nodes for short periods, thus creating decentralized meshed networks. To support these multi-hop networks, it is necessary to integrate a routing mechanism for exchanging messages. The current proposal for routing in 6LoWPAN networks consists in using the RPL protocol, described in RFC 6550 [9]. This protocol is specially designed to work with constrained devices in terms of computing, memory and battery, so it has become a de facto standard in WSN and IoT.

Using this protocol, a tree topology is constructed, named as DODAG (Destination-Oriented Directed Acyclic Graph), in such a way that the information is routed to a single destination, thus ensuring that no loops are produced between the nodes belonging to the topology.

Different types of ICMPv6 (Internet Control Message Protocol for IPv6) control messages are used to build the RPL topology. DIO (DODAG Information Object) messages contain all the necessary information for a new node to join the RPL topology and establish ascending routes. DIS (DODAG Information Solicitation) are sent by new nodes to trigger the sending of a DIO message with the updated information of the RPL tree. DAO (Destination Advertisement Object) messages are used to establish downwards routes.

All these control messages have a secure variant to implement different levels of security, such as integrity, confidentiality, replay protection and delay protection. Each of the RPL secure messages includes different configuration fields to choose the security level and both the encryption and signature algorithms.

The Security Algorithm field allows choosing the algorithm that is used in the whole network to carry out the task of encryption, messages authentication codes (MAC) and signature. Table II shows the possible different values for this field. According to the information reflected in the RPL specification, only a single configuration is possible.

The cryptographic mode of operation in RPL is based on CCM (Counter with CBC-MAC) and AES-128 to provide both authentication and confidentiality. In this case, that signature is also included; the mechanism used is RSA (Rivest–Shamir–Adleman) with SHA-256 (Secure Hash Algorithm) to provide integrity and data authenticity.

TABLE II. SECURITY ALGORITHM FIELD IN RPL [9]

Algorithm	Encryption/MAC	Signature
0	CCM with AES-128	RSA with SHA-256
1-255	Unassigned	Unassigned

Another parameter included in the security fields is the Key Identifier Mode (KIM). This parameter indicates whether the specific key used to secure the control messages is determined implicitly or explicitly. This configuration uses both Key Source and Key Index fields included in each control message to be able to use different key types, such as peer-to-peer keys, group keys, and signature keys. Table III shows the possible values of the KIM field.

Depending on the value that the KIM parameter takes, different levels of security can be configured, as shown in table IV. This parameter can be different for each type of RPL message and allows different levels of authenticity and, optionally, for the data confidentiality.

All these security mechanisms, described in the RFC, allow protecting the RPL protocol from different types of routing attacks.

Different studies have been carried out detailing the security problems of the RPL protocol, and the possible types of RPL attacks [10] [11].

TABLE III. KEY IDENTIFIER MODE IN RPL [9]

KIM	Meaning	Key Identifier Length (octets)
0	Group key used. Key determined by Key Index Field. Key Source is not present. Key Index is present.	1
1	Per-pair key used. Key determined by source and destination of packet. Key Source is not present. Key Index is not present.	0
2	Group key used. Key determined by Key Index and Key Source Identifier. Key Source is present. Key Index is present.	9
3	Node's signature key used. If packet is encrypted, it uses a group key, Key Index and Key Source specify key. Key Source may be present. Key Index may be present.	0 / 9

TABLE IV. SECURITY LEVEL IN RPL [9]

KIM	LVL	Attributes	MAC/Sig Len
0,1,2	0	MAC-32	4
	1	ENC-MAC-32	4
	2	MAC-64	8
	3	ENC-MAC-64	8
	4-7	Unassigned	N/A
3	0	Sign-3072	384
	1	ENC-Sign-3072	384
	2	Sign-2048	256
	3	ENC-Sign-2048	256
	4-7	Unassigned	N/A

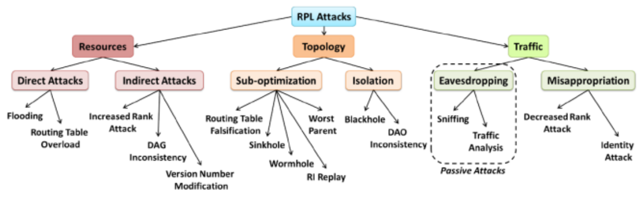

Fig. 3. Different types of routing attacks in RPL [10]

Fig. 3 summarizes some of these attacks. Using the security fields described above, RPL protocol can be configured to use three different security modes:

- Unsecured: this is the default mode of operation in RPL, in which security is not applied to any of the control messages. In this mode DIS, DIO and DAO messages do not have the security fields, therefore the control messages are sent without any security.
- Preinstalled: in this second security mode, all the control messages use its secure version. If a new node wants to join the RPL network, it must have a pre-installed key to be able to use any of the security levels described above. In this mode, the new node can be a final node or perform routing tasks.
- Authenticated: in the third security mode, all the control messages use its secure version. If a new node wants to join the RPL network, it must have a pre-installed key to be able to use any of the security levels described above. However, unlike the previous mode, the node cannot act as a router. In order for the node to perform routing, a second key must be requested from a key authority.

Although all of these security mechanisms are described in the RPL specification, a real implementation does not have to include the different security features. As indicated in the RFC, an implementation may only include a subset of these security specifications. In fact, the Contiki OS does not include any security mechanism for RPL. Some described mechanisms, such as signature or key management, may include too much complexity for constrained devices with few computing resources. Some security implementations have been made for RPL to validate its performance in Contiki OS [12].

In addition, different methods of encryption and authentication can be used in a network that uses the IEEE 802.15.4e MAC protocol. The reason is that the ICMPv6 messages used in RPL are embedded in IEEE 802.15.4 messages. If both the RPL frame and the IEEE 802.15.4 frame were encrypted and/or authenticated, the security mechanism would be redundant.

C. Transport and Application layer with TCP and MQTT

One of the protocols that are supported within the application layer is MQTT, which is a popular implementation of a publish/subscribe solution for exchanging messages. This protocol has multiple advantages, among which are ease of use, reliability and different authentication methods that add a greater degree of security to communications. The MQTT protocol supports different authentication strategies that can be used according to the needs of the application to be implemented, thus providing greater versatility to the protocol. The types of authentication that the protocol supports are:

TABLE V. SECURITY TYPES FOR MQTT PROTOCOL

Type	Description
No security	Default mode, in which anonymous and unauthenticated users can use the broker without any restriction.
Basic	The broker maintains a list of allowed users to make a connection and use. The broker will maintain a list of login-password pairs. Clients that connect to the broker must provide this information
TLS	TLS encryption is used for the connection and exchange of information. The broker must maintain a certificate that guarantees its authenticity.
TLS with client certificate verification	TLS encryption is used for the connection and exchange of information. In this case, both ends (client and broker) guarantee its authenticity by means of two certificates. It is the most secure mode of communications that the protocol provides.

Credentials sources for authenticating a client can be implemented in several ways, with LDAP (Lightweight Directory Access Protocol), a SQL (Structured Query Language) database, a file, or a web service, depending on the use case. For instance, inside SCOTT project some use cases draw on a LDAP server to authenticate publishers and subscribers, added on top of a TLS (Transport Layer Security) with client's certificate connection type, which offers a very reliable and secure communication.

Additionally, the majority of MQTT brokers also support the use of access lists (ACL's) that relate authenticated users and the actions allowed by broker regarding the publication and/or subscription of certain topics. In this way, authentication is complemented with authorization, thus increasing security over the whole system, and both can be managed directly by the LDAP server.

To check the different effects that each of the authentication modes offered by the protocol can cause, several tests have been carried out to evaluate how they affect performance in some of its variants: connection quality, speed, bandwidth, or quality of service. Among all the possibilities, we have opted for the design of two types of tests:

- Effects on the connection time: in which different number of clients are instantiated and the average time between sending the connection request with the broker and receiving the established connection confirmation message is checked.
- Effects on the rate and delivery of messages: in which different number of clients and rates of periodic message sending are instantiated and it is verified how the broker is able to manage and address these messages through the queue of departure to the points of destiny.

Results show that the type of security selected affects primarily the connection time, because it is in this stage when the secure channel is established. During the exchange of data messages, security does not introduce relevant delays and communication is mainly affected by the number of connected clients and the publishing rate, because certificate exchange is done only when establishing the connection (see Fig. 4). It is important to highlight that to assume this TLS handshake overhead negligible, TCP (Transmission Control Protocol) connections should be long living, maintaining the session open. More information about the use and best practices of TLS with MQTT can be found in [13].

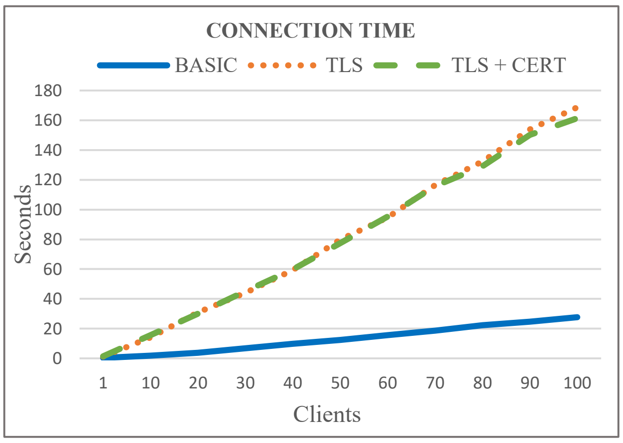

Fig. 4. Impact in connection establishment time for different security types and number of clients in MQTT.

Another key element to be considered that requires security measures is the User Interfaces (UI). The described MQTT security can be extended for its use in UI, as it can be the provider of the data to be shown. A proposed implementation is a UI that requires user authentication (user/password). These credentials are then used for connecting to a MQTT broker plus LDAP, allowing or banning the user to become a trusted client (either publisher or subscriber). Furthermore, a guideline to deploy MQTT consistently with the NIST (US National Institute of Standards and Technology) Framework for Improving Critical Infrastructure cybersecurity can be found on [14].

D. *Device provisioning*

Another key aspect when deploying a WSN securely is device provisioning. The possibility of restraining adding new devices to floor-plant network and registering these events for security analysis is an important addition for the secure design of an IIoT application.

For this purpose, a provisioning method, involving physical access to devices, installing operator credentials and maintenance logs, is proposed. Sensor nodes include an NFC tag (protected against writing and encrypted), which contains ID information related to the device. When scanned with a tablet or smartphone, the online deployment UI, managed by the gateway, is launched directly to allow the installer to validate the operation. Using NFC and web UI authentication results in a more secure process, that can be considered as OOBA (Out of band authentication) [15].

On one hand, the UI is triggered with a node token that needs to be authorized. The user must be logged on the UI; otherwise, credentials are required by the interface before granting access to the UI. If the user has correct permissions to install the type of device scanned, the sensor node is added to the device list, and the operator can then finish configuring it.

On the contrary, if the user logged in the UI does not have the correct credentials, the sensor is not added to the list and therefore it is forbidden from sending data. Fig. 5 shows a diagram illustrating this mechanism.

This operation is also registered in a maintenance log. This recorded information comprises the identity of the deploying operator and device id, timestamp, and configuration loaded in the device. This enables the traceability of any problem arisen during installation time.

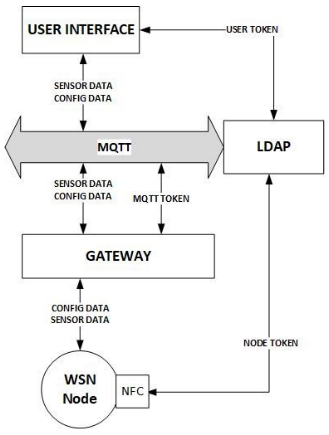

Fig. 5 Authorization diagram for WSN nodes provisioning with NFC

Another approach to guarantee a secure authentication and validation of a sensor node during the in-field deployment phase is the use of cryptographic hardware elements within the embedded platform architecture, which allows the support of an additional degree of security among the participant nodes, prior to starting the WSN data communication phase.

By integrating a dedicated hardware security module besides the main processing core of the node (optimized in terms of energy and computational resources, and specifically intended to assure the end-to-end communication chain between the remote sensor node and the base-station of the ad-hoc network) allows providing trustability on the edge against a broader spectrum of possible wireless and low-level attacks.

This strategy can also enhance the operational and remote reconfiguration/reprogramming and maintenance phases where securing the control packet exchange process is crucial to assure the trustability chain.

V. Safety Mechanisms for WSN

The term safety is very easy to overlap with security. It is used to refer to the condition of being protected from aspects that are likely to cause harm. It can be used to refer to controlling or preventing the risk of being harmed, even if it is unintended. Therefore, applied to IoT scenarios, it can be seen as protecting against external or internal problems that may cause malfunctions, broken equipment, material losses or human damage.

Given the characteristics of the proposed WSN, safety measures must be oriented to provide robust operation during critical moments such as connectivity problems, battery shortage, or other application related events and alarms that can lead to loss of relevant data or actuation commands.

To address these safety concerns during the WSN operation, the proposed design relies in three aspects: developing robust, resilient and self-healing MAC and routing layers that optimizes connectivity quality, implementing different operation modes in the WSN nodes so they can react to unexpected or troublesome conditions, and prevent data loss at Gateway level with redundant databases and backups.

A. Safety MAC & Routing layer:

The robustness of the wireless communication links can be improved with the medium access mechanism defined in the amendment of IEEE 802.15.4e standard. TSCH method is specially defined to work in scenarios with interferences and possible fading due to shadowing and multipath effects. This protocol usually works at a frequency of 2.4 GHz. This frequency band, commonly referred to as ISM (industrial, scientific and medical) radio band, is a medium shared by many other technologies and applications, so it is prone to have interferences.

TSCH mechanism allows performing a frequency hopping each time a new packet is transmitted. The exchange information will suffer fewer losses as it mitigates the impact of interferences in a given channel.

In the IEEE 802.15.4e standard, 16 channels are defined for the 2.4 GHz band. A device that works at 2.4 GHz may use all or a subset of channels. Some channels allow coexistence with other wireless technologies like IEEE 802.11 (Wi-Fi). Fig. 6 shows how some IEEE 802.15.4 channels do not overlap with Wi-Fi channels at 2.4 GHz. Current implementation of protocol stack is configured to use a subset of four channels that improves the communication exchange.

RPL protocol has been designed to be an auto-configurable and self-healing routing protocol. This routing protocol uses different objective functions, in such a way that the nodes of the network are able to choose their preferred parent with the most robust communication link. These objective functions use different rules and quality metrics to select the preferred parent among all its close neighbours. Whenever the quality of the link decreases, the device automatically switches from parent to improve its link quality.

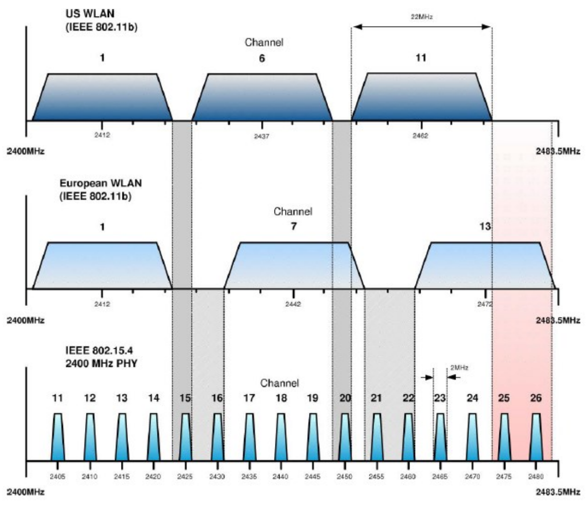

Fig. 6 Coexistence of IEEE 802.11 and IEEE 802.15.4 at 2.4 GHz [16]

The current implementation of the protocol stack has an objective function with multiple quality metrics, such as RSSI (Received Signal Strength Indicator) and ETX (Expected transmission count). These two routing metrics allow evaluating the received signal level in each channel individually, as well as the packet error rate. Each of these quality metrics is filtered using an EWMA (Exponentially Weighted Moving Average) filter.

This filtering enables smoothing the changes of each metric to avoid unnecessary parent changes (changing to frequently also hinders communication). Finally, both parameters are combined in a quality estimator using fuzzy logic and different weights for each metric. Further information about this quality estimator implementation can be found in [17].

B. Safety operation modes for WSN nodes

Following the safety requirements, even after the efforts destined to strengthen the networking to avoid issues due to connectivity failures, it should be taken into account that connectivity or power may fail. According to the conditions when the node is operating, WSN nodes implement the following modes:

- *Functional mode*: nodes work in functional mode when it is connected with optimal quality to the WSN network, remaining battery is in optimal level and there are not preconfigured alerts activated. In this mode of operation, the node is reading sensors data such as temperature, humidity, accelerometers, gyroscope, barometer, and power consumption, at a preconfigured rate. Then, the acquired data is transmitted with an asynchronous on change policy: data is only sent to the gateway when the newly measured value is different from the previous, with a configured hysteresis margin. Additionally, there is a keep alive mechanism to avoid false assumptions if connection is lost. This policy helps save energy and avoids network congestion. In addition, every time the node jumps back to functional mode it sends the current sensors and alerts information, and enqueued sensor data if it was stored locally.
- *Local-loop mode*: nodes enter Local-loop mode when connectivity is lost with the WSN network and the battery is still in optimal level. In Local-loop mode, the node works with the Functional mode configuration. The node is reading the sensors data, then the acquired data is saved in a local memory only when a configured threshold value for each sensor is exceeded. Once the node brings back the connection, the stored data is sent with the timestamp in which the measurement was taken.
- *Safety mode*: nodes enter safety mode when preconfigured alerts are activated or battery level is below 20%. In Safety mode, the node is able to operate according to the reason causing the state transition:
    - o When the transition is due to the battery level falling below the limits, the node disables the sensors that are not configured as critical, and disables the quality level LED indicators and

any other energy consuming peripherals in order to maximize battery lifetime.

- When the transition is activated by an alert, the node sends the alert information to the Gateway and saves the alert in local memory. In case the node is capable of actuation, it remains idle until the alert is acknowledged and deleted.

In both cases if the node is working offline, the data acquired and the activated alerts will be saved in local memory. When the node regains connectivity, alerts and sensor data will be sent to the WSN Gateway. It is worth noting that if the node suffers a complete power loss, next time it is powered up it must establish the secure channel described in security section.

Transitions between modes are described in Fig.7 and Table VI below.

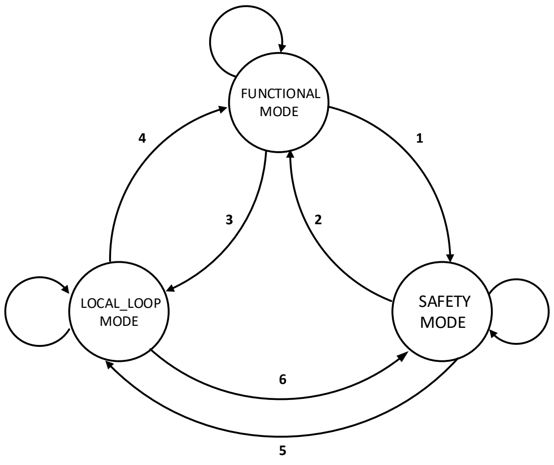

Fig. 7 Operation modes state machine

TABLE VI. STATE TRANSITION OPERATION

State Transitions		
States	***Jump***	***Description***
Func →Safety	1	Battery level is below 20% or configured alert is activated.
Safety →Func	2	Battery level is above 20 % after charge, configured alerts are deleted and there is connectivity with the WSN network
Func →L.L	3	The node loses the connection with the WSN network
L.L →Func	4	The node regains connectivity with the WSN network, battery above 20% and no alerts.
Safety →L.L	5	Battery level is above 20%, configured alerts are deleted and there is no connectivity with the WSN network
L.L →Safety	6	Battery level is below 20 %, configured alerts are enabled and there is no connectivity with the WSN network

C. *Local GW database and remote backup*

The GW integrates a mongoDB database that stores locally all data from sensors, in order to avoid information loss in case of connection failure with the broker. This provides a redundant data storage locally, which can avoid total loss of data in case remote databases fed by these sensors are no longer available.

It also runs a database backup/restore script that periodically creates a copy of the database. These backups can be uploaded to a secure repository where it can be downloaded by authenticated users.

VI. CONCLUSIONS

The proposed design of a WSN for IIoT applications stresses the importance of including security and safety concerns from early steps of the implementation, and provides solutions for each critical technology selection.

Regarding security aspects, after testing mechanisms for MAC layer some recommendations have been made in order to adopt a compromise in complexity to fit restrained devices without resources. It has also been analysed the impact of similar mechanisms for the routing RPL layer, which showed that it may be redundant after using MAC security mechanisms.

Application layer has also been studied in order to provide hints to use MQTT securely, analysing also the impact of security mechanism in this protocol. Finally, security in relation to deployment and human intervention has been addressed proposing a solution that requires physical access and authorization.

On the safety aspects, different modes of operation and other tools in order to minimize hazard due to loss of data have been proposed.

There may be other valid designs and implementations, but the presented work aims to provide a solid proposal and recommendations in each layer and protocol, taking into account many factor such as restrained resources of devices.

This means parts of the design can also be combined with other solutions in order to adapt to other application requirements, which may lead to simpler or more complex implementations.

ACKNOWLEDGMENT

This work has been supported by the SCOTT project (Secure COnnected Trustable Things) (www.scottproject.eu), which has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737422. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme, and from Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium and Norway. It has also been funded by Generalitat Valenciana through the “Instituto Valenciano de Competitividad Empresarial – IVACE”, and by the MCyU (Spanish Ministry of Science and Universities) under the project ATLAS (PGC2018-094151-B-I00), which is partially funded by AEI, FEDER and EU.

REFERENCES

[1] Ali S., Al Balushi T., Nadir Z., Hussain O.K. “WSN Security Mechanisms for CPS”. In: Cyber Security for Cyber Physical Systems. Studies in Computational Intelligence, Vol 768, 2018.

[2] Yosef Ashibani, Qusay H. Mahmoud. “Cyber physical systems security: Analysis, challenges and solutions”, in: Computers & Security, Vol. 68, 2017.

[3] S. W. Pritchard, G. P. Hancke and A. M. Abu-Mahfouz, "Security in software-defined wireless sensor networks: Threats, challenges and potential solutions," 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, 2017.

[4] A. Othman and D. Maga, "Relation Between Security and Energy Consumption in Wireless Sensor Network (WSN)," 2018 New Trends in Signal Processing (NTSP), Demanovska Dolina, 2018

[5] OASIS Message Queuing Telemetry Transport (MQTT) version 5.0 https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[6] OPC-UA Security How it works, Information Revolution 2014, Microsoft Conference Center. https://opcfoundation.org/wp-content/uploads/2014/08/11_OPC_UA_Security_How_It_Works.pdf

[7] Atmel White Paper, “Security for Intelligent, Connected IoT Edge Nodes”, 2015.

[8] Encryption Standard (AES) (FIPS PUB 197). https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[9] T. Winter et al. “RPL: IPv6 routing protocol for low-power and lossy networks”, RFC 6550, 2012.

[10] A. Mayzaud, R. Badonnel, I. Chrisment. “A Taxonomy of Attacks in RPL-based Internet of Things”. International Journal of Network Security, IJNS, 2016, 18 (3), pp. 459 – 473.

[11] P. Pongle and G. Chavan. “A survey: Attacks on RPL and 6LoWPAN in IoT”. International Conference on Pervasive Computing (ICPC), 2015.

[12] P. Perazzo et al. “An implementation and evaluation of the security features of RPL”. International Conference on Ad-Hoc Networks and Wireless, 2017.

[13] MQTT Security Fundamentals: TLS/SSL https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl

[14] MQTT and the NIST Cybersecurity Framework Version 1.0. Edited by Geoff Brown and Louis-Philippe Lamoureux. 28 May 2014. OASIS Committee Note 01. http://docs.oasis-open.org/mqtt/mqtt-nist-cybersecurity/v1.0/mqtt-nist-cybersecurity-v1.0.html.Advanced.

[15] Wu, Longfei, et al. “An out-of-band authentication scheme for internet of things using blockchain technology”. In International Conference on Computing, Networking and Communications (ICNC), pp. 769-773, 2018..

[16] NXP Application Note, “Coexistence of IEEE 802.15.4 at 2.4 GHz”, 2013.

[17] J. Vera-Pérez, D. Todolí-Ferrandis, S. Santonja-Climent, J. Silvestre-Blanes and V. Sempere-Paya. “Path Quality Estimator for 802.15.4e TSCH Fast Deployment Tool”. In Telfor Journal, Vol. 10, No.1, 2018