Computers and Electrical Engineering 41 (2015) 325–341

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Performance evaluation of WirelessHART networks using a new network simulator 3 module ☆

Marcelo Nobre *, Ivanovitch Silva, Luiz Affonso Guedes

Federal University of Rio Grande do Norte, Department of Computer Engineering and Automation, Campus Universitario, Lagoa Nova, 59078-900 Natal, RN, Brazil

ARTICLE INFO

Article history:
Available online 6 June 2014

ABSTRACT

Nowadays, wireless communication is a tendency in industrial environments, enabling the addition of new applications and saving resources when compared to their wired counterparts. In this context, the WirelessHART specification is emerging as a solution for the last mile connection. Despite its high degree of applicability, a WirelessHART network faces some challenges. Some of the most challenging problems are its reliability, energy consumption, and interference from the environment, issues which involve the lower layers. In order to enable performance evaluations in a low cost and scalable way, we propose a new NS-3 module for the WirelessHART physical layer including an error model (Gilbert/Elliot), station positioning, signal attenuation and energy consumption. Furthermore, the module permits configuring each link with different fault probabilities, allowing more accurate simulations. For validation, the scheduling and routing approaches were statically configured assuming different topologies and probabilistic error scenarios already validated in the literature.

1. Introduction

Traditionally, distributed applications in industrial environments have been supported by wired communication networks [1]. However, the industry has recently shown an interest in moving part of the communication infrastructure from a wired to a wireless environment in order to reduce the costs related with installation, maintenance, and the scalability of the applications. In this context, Industrial Wireless Sensor Networks (IWSN) actually represent the best candidate for adoption as the communication solution for the last mile connection in process monitoring and control applications [2]. Among its many advantages, the absence of a wired infrastructure enables IWSN to extract information in a simpler way than the traditional monitoring and instrumentation techniques [3].

The use of wireless technologies in industrial environments has always been viewed with great skepticism by plant managers. These concerns were created primarily by the unreliable nature of the communication channel [4]. This was aggravated by the fact that the equipment is installed in areas subject to the influence of external agents (noise, interference, adverse weather, natural obstacles), which can lead to higher error rates than when using wired technologies [5]. Other errors in the communication channel are due to signal attenuation (mainly caused by the power loss due to the distance between the transmitter and receiver) and the problem of multiple paths (producing reflection, diffraction, and scattering of the transmitted signal: multiple copies of the data can interfere with one another constructively or destructively). In

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. S. Smys.

* Corresponding author. Tel.: +55 84 3215-3771; fax: +55 84 3215 3738.
E-mail addresses: marcelonobre@dca.ufrn.br (M. Nobre), ivan@imd.ufrn.br (I. Silva), affonso@dca.ufrn.br (L.A. Guedes).

http://dx.doi.org/10.1016/j.compeleceng.2014.05.005

general, errors in wireless communications are transient: that is, the communication channel is bad for a while and then returns to normal [6].

Nowadays, WirelessHART is emerging as a standard solution for IWSN, providing a simple, reliable and secure communication protocol [7]. Despite its high degree of applicability, a WirelessHART network faces some challenges. Some of the most challenging problems are its reliability, energy consumption, and interference from the environment. Thus, the lower layers assume a major role in addressing such challenges in the sense that radio and the frequency hopping mechanism have a direct influence on the reliability, energy consumption and issue of interference. A fundamental concern is the evaluation of these layers using flexible, scalable and precise procedures.

Concerning the design issues for this kind of network, two types of tools could be used for evaluation purposes [8]: testbeds and simulations. Testbeds provide more accurate results although they require more configuration, building, and financial efforts, which are aggravated by scaling issues. Simulation, on the other hand, minimizes these efforts but obtains less accurate results. In many situations, both approaches are needed in order to make an adequate performance evaluation and reliability analysis of the network [9].

This paper treats the simulation approach. Thus, with the object of contributing to the performance evaluation and reliability analysis of WirelessHART networks, we here present a new Network Simulator 3 (NS-3) Module for this kind of industrial wireless sensor network. This module for the NS-3 simulator mainly addresses the functionalities of the physical layer. The NS-3 simulator was chosen due to the data presented in [10], in which NS-3 obtained the best overall performance among the simulators. Also, [11] concludes that NS-3 is reaching a level of maturity that makes it suitable to replace NS-2 in various applications and scenarios.

The remainder of this paper is organized as follows. In Section 2, a brief overview of the parts of the WirelessHART specification relevant to the developed model is presented. Section 3 describes the state of the art. Then Section 4 describes the components of the simulation module, focusing on the simulator NS-3 core, the Gilbert/Elliot error model, and the upper layer abstraction. Section 5 presents the experiments performed with the developed module and the results obtained. Finally, concluding remarks are provided in Section 6.

2. WirelessHART

WirelessHART is an extension of the HART protocol to support wireless communication. In September 2008, the WirelessHART specification (HART 7.1) was approved by the International Electrotechnical Comission (IEC) as a publicly available specification (IEC 62591) [12]. WirelessHART was the first industrial wireless communication technology to attain this level of international recognition [7].

WirelessHART defines eight types of devices, as presented in Fig. 1: network manager, network security, gateway, access point, field device, adapter, router, and handheld device. All devices that are connected to the wireless network implement basic mechanisms to support network formation, maintenance, routing, security, and reliability.

WirelessHART has a physical layer based on IEEE 802.15.4, but it implements its own medium access control (MAC) sublayer. The MAC is based on a TDMA (Time Division Multiplexing) scheme that uses *superframes*. Superframes are composed

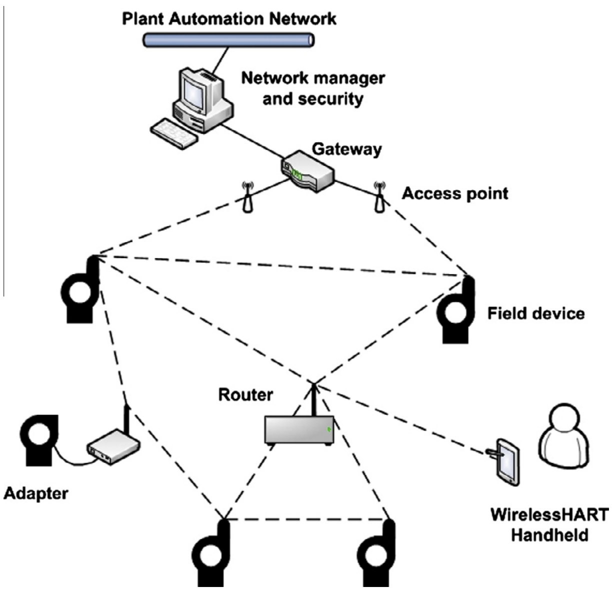

Fig. 1. WirelessHART devices.

of slots, and the number of slots indicates the periodicity of the superframe. In order to support different transmission intervals, a WirelessHART network can use multiple superframes with different numbers of slots. Each slot has a fixed duration of 10 ms, which is enough time to transmit a packet and receive an acknowledgment message (the maximum packet size is 133 bytes including headers). The organization of the time slots and how they are related to the superframe is shown in Fig. 2. As the protocol is TDMA-based, the source and destination stations have specific times to transmit and sense the medium. More details about the values of those times can be found in [12].

Additionally, slots can be dedicated or shared. The use of dedicated slots is more common. Shared slots are used for transmission retries and advertising indication during the join procedure. A slot supports up to 15 channels. Thus, theoretically 15 devices can simultaneously transmit in the same slot time. The standard uses a mechanism of frequency hopping and a blacklist channel to minimize the influence of noise in network operation and consequently increase the reliability of communication.

For the scheduling process, the WirelessHART specification [12] only suggests an algorithm that meets the requirements defined by the specification. This algorithm suggestion states that an alternate route should be used, if possible, besides the main route from the data source to the data destination. For each packet to be transmitted by a station, two time slots must be reserved for the scheduling purpose: the first for the transmission and the later one for a possible retry (both through the main route). If an alternate route is defined for that link, a third copy packet must be sent through that alternate route. For the sake of understanding the scheduling approach, an simple example is presented in Fig. 3. In the first timeslot of the example described in Fig. 3, the data is produced at the source station 'S' and sent to the destination station 'D' using the main route S–1–D and the alternate route S–2–D. According to the aforementioned scheduling algorithm, the two first time

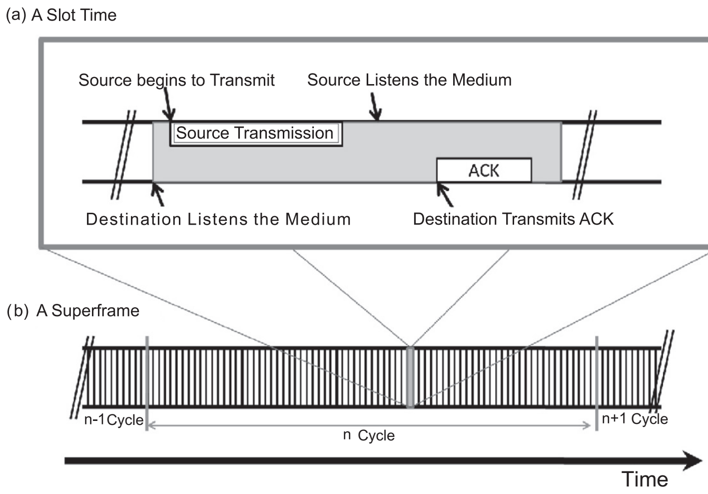

Fig. 2. Communication mechanisms for the WirelessHART data link layer.

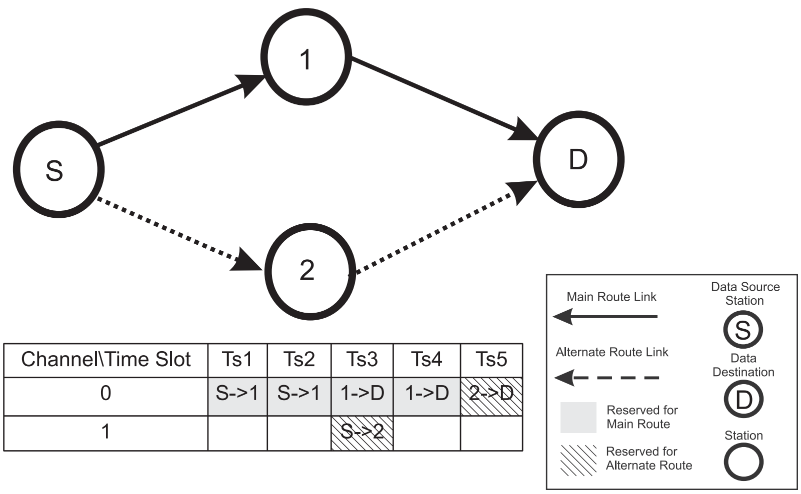

Fig. 3. An example of scheduling on a simple topology.

slots (Ts1 and Ts2) are reserved for the packet's transmission and retransmission through the main route on channel 0. On the other hand, Ts3 reserves channel 1 for the transmission through the alternate route. Again, channel 0 in Ts3 and Ts4 are reserved for the main route's transmission and retransmission. Finally, in Ts5, channel 0 is reserved for the transmission of the alternate route's sole try.

The scheduling must be built over defined routes. WirelessHART presents two main approaches for routing packets: graph routing and source routing. *Graph* in the WirelessHART context means a set of edges that connects the devices on the network. In graph routing, the edges are created by the network manager and downloaded to each device. The sender device writes a graph id into the header of the message to send a packet. As the packets arrive at a station, that station forwards (or consumes, if the station is the final destination) the packet according to the previously stored data. Each intermediate device can be configured with multiple neighbors to help the packet's forwarding. In source routing, a single fixed path of devices is written by the source in the header of the packet. Then each device in the route forwards the packet to the next specified device until the destination is reached. There are no alternate routes in this mode, so if any device fails on a path, the whole path fails too. This is mostly used for network diagnostics. The WirelessHART protocol defines three types of graphs: broadcast, uplink, and downlink.

A broadcast graph links all devices downwards from the gateway to all devices on the network for the transmission of common messages and configuration commands. An uplink graph is a graph that connects all devices upwards to the gateway for transmitting process data. Finally, there is a downlink graph for each device in the network in order for the gateway to send unicast messages to each of them.

In this paper, only graph routing and uplink communication will be simulated. The reasons for this restriction include the robustness of graph routing in comparison with source routing and the importance of monitoring the data from network devices. However, we would like to stress that the proposal can be extended so as to treat source routing and the downlink and broadcast communication directions.

3. Related work

There are already some papers in the literature that have dealt with simulations for the WirelessHART protocol. In this section, we will discuss some of them that are relevant to the present paper.

The papers described in [13,14] present a WirelessHART simulator applied to control systems. The authors have extended the TrueTime simulator in Matlab's Simulink tool. The WirelessHART MAC protocol was implemented using C++ classes with the corresponding MATLAB Mex-interfaces. Those C++ classes were used for the purpose of obtaining a better simulation speed. [13] compares the performances of WirelessHART and ZigBee networks. The tests were realized with a fixed number of lost packets. Although the simulator package is available for download, it is not open source for the community, and it does not address communication modeling aspects, such as model error.

[15] developed a simulator to provide a verification of transmission conflicts and possible incompletions on the network graph. Those verifications are based on the devices' transmission features and the communication/processing times. The simulator is also implemented on the TrueTime simulator for Matlab's Simulink tool. The simulator does not focus on the physical layer and radio communications, nor does it implement noise interference and multi-hop communication. It uses only one radio channel.

[16] proposed a simulator intended to deal simultaneously with WirelessHART, traditional IEEE 802.15.4, and IEEE 802.11.b networks. The simulator was implemented using the OMNet++ simulator. The objective was to study coexistence issues and optimal network parameters. The networks were implemented independently and are not aware of each other. The interference is simulated by a common element named *Interference Module*. From coincident transmissions of the involved networks and the signal energy level, the simulator calculates the packet error rate. A Bernoulli trial is used to determine if the packet is rejected or accepted, but does not correlate the different errors. This model has been substantially improved by [17] which proposes the use of a co-simulation framework based on the interaction between TrueTime and OMNet++ simulators. The idea was to improve the overall coexistence management, however no energy consumption analysis was conducted.

Finally, [18] presents a WirelessHart simulator developed over the NS-2 simulator. The project is quite complete and includes the network manager as well as the whole stack of the standard. It has adopted the physical layer from native implementation NS-2 and adapted the data link layer from the IEEE 802.15.4 mode of the NS-2. The upper layers (Network and Application layers) were implemented in order to allow analyzing the number of required communications for node joining and connection establishment. In spite of its having been very well developed, however, it has not addressed special attention to the functionalities of the lower layers, such as the more accurate error and energy consumption models that are the main focus of our proposal. We believe that this is essential for carrying out a performance evaluation of this type of wireless network in more realistic industrial scenarios. In addition, NS-2 is harder to use than NS-3 [19] and only the NS-3 allows emulation experiments where simulation and real devices are used together.

Therefore, when compared with previous works, our proposal aims first at modeling the physical layer along with the features of attenuation, positioning, and battery consumption. Also, for a more realistic simulation, the proposal implements the Gilbert/Elliot error model. This model calculates the packet error rate based on the packet size and the probabilities of losing a bit of information and will be given in detail in further sections. The parameters of the Gilbert/Elliot error model can be configured for each link in order to apply different error probabilities, in order to model communication scenarios more realistically. The routing and scheduling algorithms suggested in the WirelessHART specification also have been implemented. Currently, they are statically configured for each simulation scenario.

4. Simulation module

A four-part architecture is planned in order to provide a simulation environment. Those parts are: the Topology Configurator, the Route Generator, the Schedule Generator, and the Simulation Module itself. The flow between these parts is displayed in Fig. 4.

The Topology Configurator consist of a module that supports a user friendly input for the topology and provide those topologies to the following module. The Route Generator takes a topology as input and sets the routes between the stations based on selected algorithms or in a user defined manner. Those routes are passed on as input for the following module.

The Schedule Generator is responsible for, based on the provided routes over a topology, providing the time-channel association of the transmission schedule for each channel among the simulated stations, according to the WirelessHART specifications or in a user defined manner.

Finally, the simulation module includes the simulation of the iterations among the stations using the NS-3 simulator. This later module is the main focus of this present work. All the features and specifications are described in this section.

The features of the developed module for NS-3 are presented in Fig. 5. As the WirelessHART physical layer is the primary focus of this paper, the main features are the ones related to that layer and the characteristics of the transmission medium (propagation, attenuation, positioning, and error probabilities). The features of the upper layers are abstractions for emulating the behavior of those layers, with the purpose of validating the physical layer implemented.

Positioning, attenuation, and propagation are native to the NS-3 implementation (for details, see [19]). The calculation of the energy consumption is based on the model presented in [20,21]. Further details on the model can be found in [22], where the preliminary idea for the module was proposed, a propagation model test was performed, the idea for the energy consumption was introduced, and the influence of the packet size on the error model was analyzed. The behavior of the physical layer is described by the state machine described in the WirelessHART specification [12]. The error model and the features of the upper layers will be given in detail in the following subsections.

4.1. Core

Network Simulator 3 (NS-3) is an open source discrete-event network simulator for Internet systems. It is the successor of NS-2, the most used network simulator in academia, although NS-3 is not backwards compatible with NS-2 [19]. For further

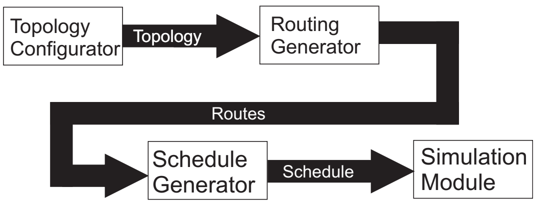

Fig. 4. The data flow between the modules.

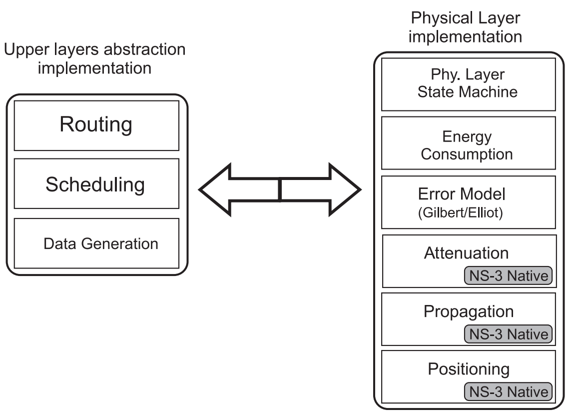

Fig. 5. The functionalities of the module implemented.

reference, much documentation is available on the NS-3 project website. NS-3 works with the C++ language (the Python language can also be used) and has been implemented using a object oriented structure, providing a modular environment in which various models can be re-used, making the software user-friendly. In [10], it is demonstrated that NS-3 has the best overall performance for memory usage and runtime of the various current network simulators (OMNet++, SimPy, and Jist), even surpassing NS-2. On top of that, the possibility of integrating different wireless technologies into the same simulated environment for a further coexistence study also influenced our choice of NS-3.

The core structure implemented in NS-3 can be seen in Fig. 6. The classes NetDevice and Channel are standard classes of NS-3, and must be extended for the implementation of the upper layers and the medium characteristics, respectively. The class WHartNetDevice stands for the implementation of a WirelessHART device, including the network and data link layers. The WHartPhy class maintains the modeling of the radio transmitter of the simulated device; it includes characteristics such as the signal detection level, the transmitter power, and the model for the mobility of the device (NS-3 native). The energy consumption model is also included in the WHartPhy class. Finally, the WHartChannel class models the characteristics of the medium, which include the 15 channels supported by the WirelessHART specification, the propagation loss, and the error models (NS-3 native).

It is important to emphasize that scope of the present paper deals only with established networks, and does not focus on the join process.

4.2. Propagation model

The propagation model determines the propagation loss through a transmission medium, calculating the receive power based on the positioning of the receiver/transmitter and the transmitter radio power. NS-3 natively has 11 different loss models included in the simulator library [19]. The models are categorized into three groups as follows: abstract propagation loss models, deterministic path loss models, and stochastic fading models.

In the first group, the propagation loss is configured in a simplistic way. Fixed Received Signal Strength (RSS), Matrix Loss Model, Maximal Range, and Random Propagation Loss are the abstract propagation loss models supported by NS-3. According to [23], these models are more adequate when the signal transmitted is spread throughout the simulation area without attenuation. On the other hand, in the second group, the propagation loss models assume a deterministic path loss depending on the distance from the sender to the receiver. The NS-3 supports the following deterministic path loss models: COST-HATA, Friis, Log Distance, Three Log Distance, and Two Ray Ground. A simulation study [23] has shown that although the latter group is more computationally complex than the former group, deterministic path loss models support more realistic scenarios than abstract models. Finally, in the third group, the propagation loss models assume fading issues. In this category NS-3 includes the models Nakagami and Jakes [19]. Despite their increasing the accuracy of the propagation loss, fading models require a high computational effort when compared to previous models [23].

In this paper we employ deterministic path loss models because their support for more realistic scenarios when compared with other techniques. The radio range was evaluated in a previous paper [22], in which the signal strength declines at a distance of around 200 m for the Friis model, 160 m for the Two Ray Ground model, and 20 m for the Log Distance model. All the models used a wavelength of 0.125 m and unitary antenna gains.

4.3. Error model

The ErrorModel class presented in Fig. 6 is native for the NS-3 core. It provides an error model for the physical layer but we choose to implement a more accurate approach which is based on the classical Gilbert/Elliot error model [24]. The model is mapped as a Markov chain which calculates the packet error rate (PER) value through the bit error rate (BER), providing different error probabilities for different packet sizes. Once the PER threshold is calculated, a random number is generated

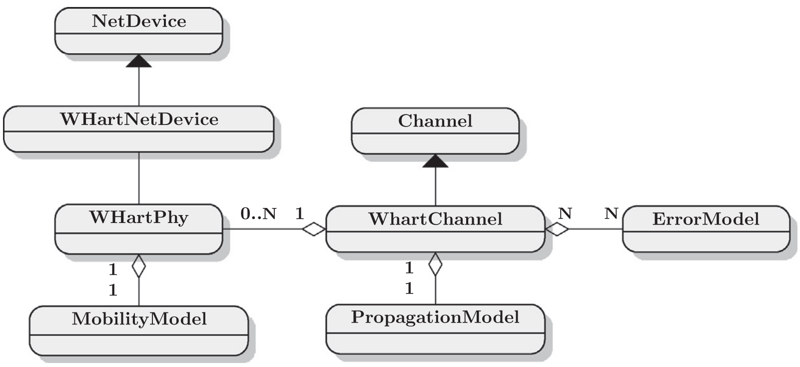

Fig. 6. Class diagram of the WirelessHART simulation module.

and if this number is less than the threshold itself, the packet is dropped. This approach is presented in Fig. 7. The initial state is the good state, and for each new bit, the channel can remain in this state or change it. The transition probabilities are also presented in Fig. 7, where p is the probability of being in a good state and remaining in it, and q is the probability of being in a bad state and remaining in it after a transition.

The steady state probabilities for good and bad states (P_g and P_b, respectively) are

$$P_g = \frac{1-q}{2-(p+q)}, \quad (1)$$

$$P_b = \frac{1-p}{2-(p+q)}. \quad (2)$$

Finally, according to [24], the PER for an n-byte message is

$$\mathrm{PER}(n) = 1 - (P_g p^{8n} + P_b(1-q)p^{8n-1}). \quad (3)$$

For more realistic modeling, each link between two stations has its own independent error probability. Then we must configure each link with its p and q probabilities, allowing us to simulate more complex scenarios.

According to [25], we can also calculate the time spent in each state of the Gilbert/Elliot model in steady state as described in Eqs. (4) and (5). Tg indicates the time spent in the good state and Tb indicates the time spent in the bad state. These variables are useful when it is desired to configure the interference based on the duration of the noise, in preference to using the fault probabilities.

$$T_g = \frac{1}{1-p}, \quad (4)$$

$$T_b = \frac{1}{1-q}. \quad (5)$$

The presented error model improves over the model presented in the NS-2 simulator since the latter provides a simple probability input and a generated random number determines if the transmission has succeeded, also not considering of influence of the packet size. Furthermore, the implemented model present more flexibility once it supports individual error probability values for each link and the alteration of those values during the simulation.

4.4. Energy consumption model

For further studies of the energy consumption, a radio model has been developed based on the model presented by [20]. We also employ the data from the cc2500 radio transmitter [26]. The radio state machine model is described in Fig. 8. That model presents the power consumption and current values of each state, and the energy and time spent in each transition.

The modeled states and their corresponding energy consumption, according to [26], are:

- *Tx*: the radio is transmitting (37.8 mW).
- *Rx*: the radio is receiving (27 mW).
- *Idle*: the clock is turned on and the radio is ready to switch to the *Tx* or *Rx* state (2.7 mW).
- *Sleep*: the clock is disabled and the radio is waiting to be turned on (1.62 μW).

Therefore, the overall energy consumed by a device i can be calculated as follows:

$$Energy(i) = Tx_{pwr} \times Tx_{time} + Rx_{pwr} \times Rx_{time} + Idle_{pwr} \times Idle_{time} + Sleep_{pwr} \times Sleep_{time}, \quad (6)$$

where $\{Tx, Rx, Idle, Sleep\}_{pwr}$ represents the power consumed in each state whereas $\{Tx, Rx, Idle, Sleep\}_{time}$ is the time spent in each one. Note that the simulation time is given by the following sum: $Tx_{time} + Rx_{time} + Idle_{time} + Sleep_{time}$. Additionally, knowing the charge and the terminal voltage of the battery used for device i, it is possible to calculate its lifetime as follows:

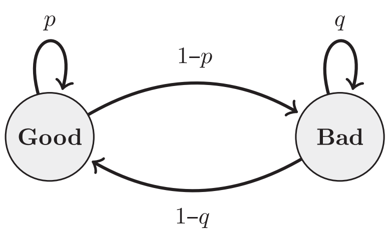

Fig. 7. Gilbert/Elliot error model.

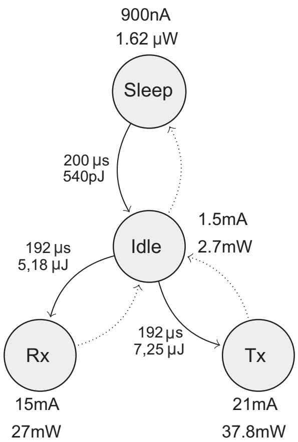

Fig. 8. Energy model of the radio CC2500.

$$lifetime(i) = \frac{(charge \times terminal\_voltage) \times simulation\_time}{Energy(i)}. \tag{7}$$

The times spent in each state of the energy model are influenced directly by the data link layer, more precisely by the scheduling algorithm. In a given slot time, a device can transmit and/or receive a packet. If there is no task to carry out, the device can simply sleep in that slot time. For the sake of understanding these issues, consider the structure of a WirelessHART slot time in more detail as described in Fig. 9 (properties of the data link layer are given in Table 1).

From the source device point of view, the state *Idle* is visited during the periods of TsCCAOffset (preparing the slot), TsRxTx (switch from reception to transmission or vice-versa), and TsRxAckDelay (waiting for preparing of the acknowledge (ACK) packet), whereas the state *Tx* is visited only during the transmission of a packet. On the other hand, the state *Rx* is visited during the clear channel assessment (TsCCA) and the reception of an ACK packet (TsTxAckDelay – TsRxAckDelay + TsAck). If the reception of the ACK packet fails, the source device keeps the state *Rx* active until the timeout (TsAckWait). In any other event, the source device remains in state *Sleep*.

The same analysis can be made from the destination device point of view. The state *Idle* is visited only during the start of slot (TsRxOffset) and the generation of an ACK packet (TsTxAckDelay). On the other hand, the state *Rx* is visited in two cases: when the packet has not arrived (TsRxWait) or during the reception of a packet (TsTxOffset – TsRxOffset + transmission time). The state *Tx* is visited only during the transmission of an ACK packet (TsAck). In any other event, the destination device remains in state *Sleep*.

4.5. Upper layers

The implemented architecture has as its main focus the development of a solid physical layer and the features of the transmission channel. The upper layers were abstracted in such a way that the network routing and scheduling were implemented as entries for the simulation, allowing the validation of the lower layers. The details are described in Fig. 10.

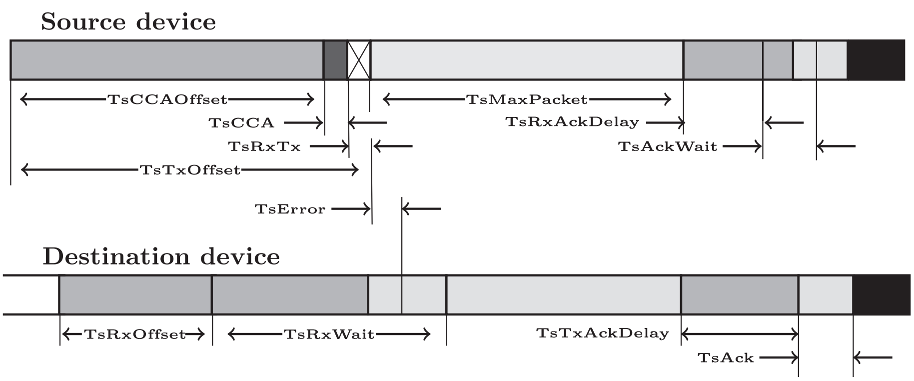

Fig. 9. Slot time structure.

Table 1
Data link layer properties used in the energy consumption model.

Variable	Definition	Value (ms)
TsCCAOffset	Start of a slot	1.800
TsCCA	Clear channel assessment	0.128
TsRxTx	Switch from reception to transmission or vice-versa	0.192
TsTxOffset	Start of preamble transmission	2.120
TsError	Synchronization reference	–
TsMaxPacket	The duration of transmitting the longest possible packet (133 bytes)	4.256
TsRxAckDelay	Listening for ACK	0.800
TsAckWait	Minimum time to wait for an ACK	0.400
TsRxOffset	Preparing to listen to a packet	1.120
TsRxWait	The limit to the wait for the start of a message	2.200
TsTxAckDelay	Duration of generating an ACK	1.000
TsAck	Time for transmitting an ACK	0.832

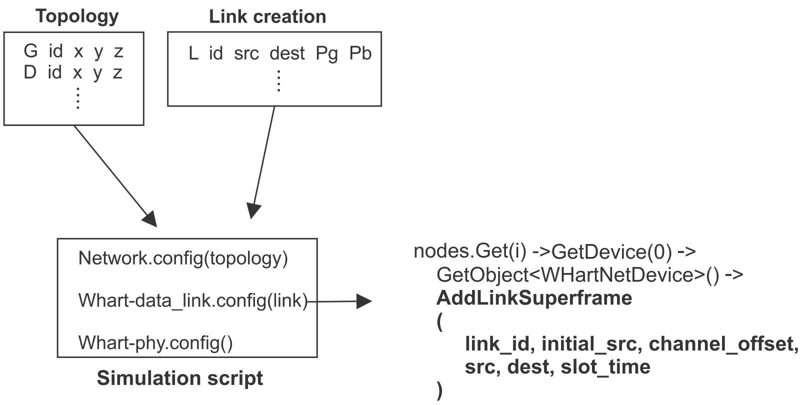

Fig. 10. Upper layers configuration.

Regarding the routing, the user must provide beforehand the information about the network topology. The configuration file for the network topology requires the following tuple: *G/D* – gateway or field device, *id* – identification, *x/y/z* – Cartesian coordinates. After that, the hop-by-hop routes are created in the link creation file based on following tuple: *L* – link, *id* – identification, *src* – source, *dest* – destination, P_g and P_b – error model. The user must identify a main route from each device to the next device on the route. The packets will be forwarded through such routes with a retry attempt if necessary. Optionally, a second route can be assigned as an alternate. If the alternate route is set, one sole attempt at transmission must be made through this route. There is no retry in this case. This strategy of routing is indicated in the WirelessHART specification (graph routing) [12].

For the scheduling, the configuration must be done as if the own user was the network manager (using the function *AddLinkSuperframe*). In order to set a transmission, the user must provide the channel offset, the number of the slot time (inside the superframe), and the source and destination devices. Additionally, the information about the creator of the datum transmitted (*initial_src*) has been encapsulated into the packet to distinguish overlap flows. Note that a device can only be assigned to listen or transmit once per timeslot. The functions *Get(i)* and *GetDevice(0)* were implemented to get a node *i* and its corresponding data link layer. The configuration method of the schedule presented provides enough flexibility so the Channel Blacklisting and the Frequency Hopping features of the WirelessHART specification can be implemented by selecting the channels that will be used in each transmission in the schedule. Also, as far as we know, the majority of the blacklisting on real WirelessHART equipment is manually inputted, coping with our implementation.

For the sake of clarity, let us describe the configuration of the upper layers used in Fig. 3. The details are presented in Fig. 11. Note that according to the WirelessHART specification, the transmission channel is based on following formulae: $(slot\_number + offset)\%15$. The topology at hand is composed of three field devices (*D*) and a gateway (*G*). The Cartesian coordinates are irrelevant for this example. Regarding the routing approach, four links were defined in the network. The error model parameters are also irrelevant for the example. Finally, seven links were configured, three of them to device *S* and four links to device 1. The former used the offsets 14, 13 and 13 whereas the latter was configured with the offsets 14, 13, 12 and 11. Note that the link with *id* = *1* was statically configured to the slot 3 (*AddLinkSuperframe (1, S, 13, S, 2,* **3***)*), however, its communication channel is automatically configured to 1 because (3 + 13)%15 = 1.

Finally, regarding the data packet production, the user can configure a single packet to be transmitted from a source station or choose to set a device to periodically produce packets. All the packets are uplink forwarded (to the gateway) and the payload value can be configured.

```
Topology                          Scheduling (only for the devices S and 1)
G   D   100  100  0               Device S
D   S   0    100  0               nodes.Get(S)->AddLinkSuperframe(0,S,14,S,1,1)
D   1   50   150  0               nodes.Get(S)->AddLinkSuperframe(0,S,13,S,1,2)
D   2   50   50   0               nodes.Get(S)->AddLinkSuperframe(1,S,13,S,2,3)

Link creation                     Device 1
L   0   S   1   0.5 0.5           nodes.Get(1)->AddLinkSuperframe(0,S,14,S,1,1)
L   1   S   2   0.5 0.5           nodes.Get(1)->AddLinkSuperframe(0,S,13,S,1,2)
L   2   1   D   0.5 0.5           nodes.Get(1)->AddLinkSuperframe(2,S,12,1,D,3)
L   3   2   D   0.5 0.5           nodes.Get(1)->AddLinkSuperframe(2,S,11,1,D,4)
```

Fig. 11. Configuration of the upper layers used in Fig. 3.

5. Results

In this section we will present the results obtained by using the proposed simulation module to evaluate some WirelessHART networks over different scenarios. Our main goal is to demonstrate the module's key features, such as: support for all WirelessHART topologies, scheduling and routing, energy consumption evaluation (device lifetime), reliability, transient faults (noise injection), a realistic and flexible error model (individual error probabilities per link and packet-sized dependent). We would like to stress that the attenuation and positioning issues are contemplated by our module but they were addressed in our previous work [22].

The main assumptions for our simulations are listed below:

- *Simulation scenarios*: we have used line, star and cluster (a particular case of mesh topology) topologies.
- *Attenuation*: if two devices are neighbors, then we assume that they are always inside each other's radio range. We have adopted the NS-3 native Friss propagation model.
- *Mobility*: the devices have fixed positions.
- *Error probabilities*: the probabilities used as parameters for the Gilbert/Elliot error model are listed in Table 2. Four scenarios were defined. Case I is a very optimistic scenario, and the ratio between T_g and T_b is around 100. The same ratio metrics for Cases II, III and IV (the most pessimistic ones) are, respectively, 20, 20, and 8. Note that the larger is this ratio, the greater is the time spent in the good state of the Gilbert/Elliot model.
- *Interference injection*: the error probabilities for each channel in a given link may be individually set. These configuration probabilities can be scheduled to occur at a user defined simulation time.
- *Routing and scheduling*: since the WirelessHART does not define an algorithm for routing and scheduling, but provides an algorithm suggestion instead, we have adopted this suggestion.
- *Superframe*: we only consider one superframe, with a 10 s length (user defined).
- *Update rate*: each source device produces only one piece of information per superframe cycle.
- *Reliability*: the reliability metric is based on the ratio between the amount of information produced by a source device and the amount of information that reaches the gateway. No redundant data is taken into account.
- *Path stability*: this metric is related to the quality of a link (1-hop pair). It is the ratio between the number of packets sent by the source hop and the number of packets received by the destination hop.
- *Battery duration*: we simulated batteries with a 1200 mAh capacity, as described in [21].
- *Packet size*: we considered a common industrial packet size (90 bytes) according to [27] and a native 9 bytes ACK packet.

As a matter of the reliability of the simulation, we obtained a confidence level of 99.8% with a confidence interval of 1% or 20,000 s of simulation time (for all the realized simulations). The values 3 as *Seed* and 7 as *Run Number* were used to initialize the random number generators in NS-3.

5.1. Star topology

The first assessed scenario was the star topology. Consider an application that is monitoring the temperature of four boilers. A field device is installed in each boiler as described in Fig. 12. The information is produced by the different source nodes portrayed in gray and also the data routes are represented by the arrows. All the data produced by the sources is destined to the gateway. The redundancy strategy is that if there is a link failure in the first transmission, a retransmission is realized in the next timeslot reserved for the scheduling for that station. If the retransmission fails, the packet is dropped. Note that if the first transmission is successfully delivered but the acknowledge packet is dropped, then the retransmission link should be used.

For this experiment we have configured each link with a different error probability (as described in Table 3) in order to demonstrate the module's capability of setting different error probabilities for each link.

According to Table 3, the more pessimistic the model is, the more energy is consumed, reducing the expected lifetime. Comparing the links 4-GW (error model – Case IV) and 1-GW (error model – Case I), it is possible to note that the former

Table 2
Error probabilities for the communication channels.

Scenario	Pg	Pb	Description
Case I	0.9999918	0.999184	Outdoor environment [28]
Case II	0.9999	0.998	Error burst scenario [6]
Case III	0.999	0.98	Error burst scenario [29]
Case IV	0.995	0.96	Indoor environment [30]

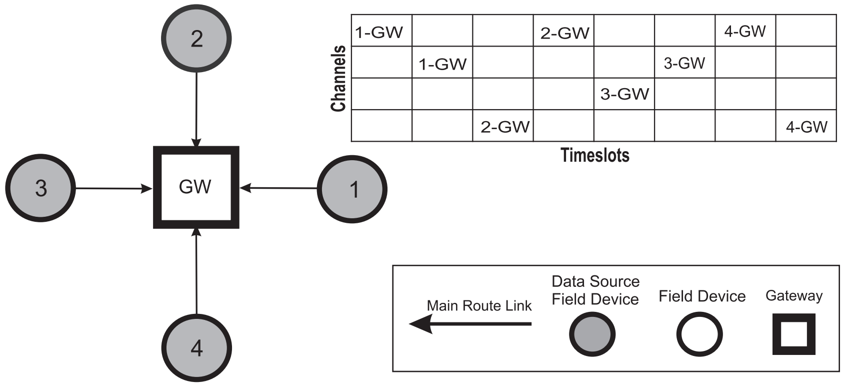

Fig. 12. Star topology.

Table 3
Path stability, reliability and lifetime values for the star topology simulation.

Error (link)	Path stability (%)	Reliability (%)	Lifetime (days)	Lifetime (ratio) (%)
Case I (1-GW)	98.06	100.00	2423	–
Case II (2-GW)	87.73	98.15	2148	11.34
Case III (3-GW)	40.74	64.70	1567	35.32
Case IV (4-GW)	1.35	2.70	1358	43.95

one consumes around 44% more energy than the latter. This behavior can be explained by the fact that devices which communicate through a noisier medium have more packets dropped and consequently need more retries in order to try to accomplish the data transmission.

Additionally, it was verified (as expected) that the reliability for each device is slightly higher than the path stability. In a star topology there are only single hops. Thus, a way to overcome interference is by transmitting the data multiple times. If the path stability is bad, retransmissions are still likely to improve the reliability of devices. For example, the difference between the reliability and the path stability to the link 2-GW is around 10%. This occurred because of the retransmission mechanism. If there was no such mechanism, the path stability would be similar to the reliability.

One of the most important features implemented in our simulation module is the capability of injecting interference for different simulation times. In order to evaluate this feature, we assumed the same star topology described in Fig. 12, however using only a field device and the gateway. The evaluation metric is the instantaneous reliability, which measures the continuous reliability for the respective field device at instant t. Links with three error scenarios were assumed: Case I, Case III, and interference injection. The third error scenario is a mix of Case I and Case III. In this scenario the communication link is started with Case I, however, between 5.000 and 9.000 s, the link is configured as in Case III. The results are described in Fig. 13.

As expected, the instantaneous reliability for the interference injection scenario decreases at instant 5.000 s. In other words, the communication link was good (Case I) until 5.000 s and after that it received the influence of interference (Case III). At the time 9.000 s, the interference ceased (the link configuration went back to Case I) and the instantaneous reliability grows continually.

5.2. Line topology

Line topology is a typical solution used for monitoring pipeline applications. In this case, the information is relayed hop-by-hop until the gateway. Fig. 14 describes an example of a line topology. For the sake of clarity, we assume that only the last field device on the line is the data source. The other devices only relay the information generated.

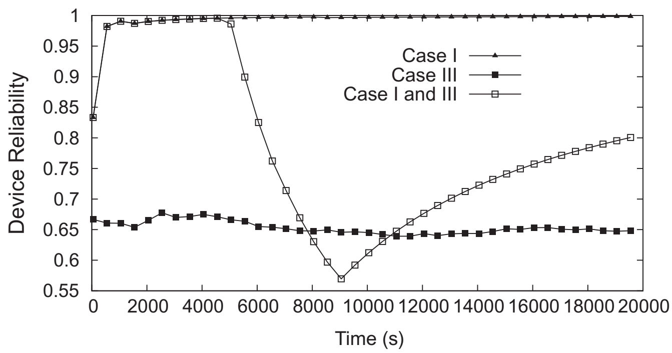

Fig. 13. Influence of interference injection for device reliability.

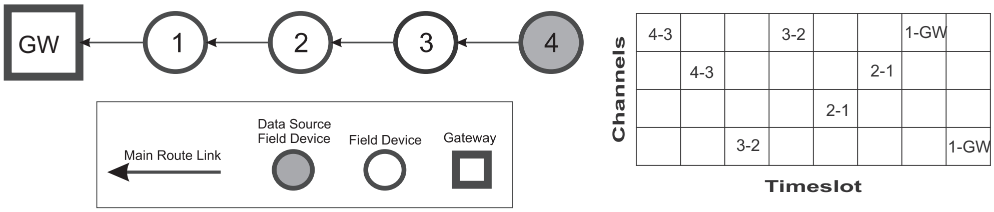

Fig. 14. Line topology.

For the line topology, we intend to conduct two types of experiments. The first one is related to the reliability evaluation for the tail device of the line topology. On the other hand, the second experiment analyzes the influence of the error model on the energy consumption and reliability.

A fundamental requirement for any network is to identify the most critical device. In a line topology, the tail device is considered the most sensitive because it depends on all the other devices to reach the gateway. Thus, the reliability of the tail device was evaluated, assuming different network sizes. The results are described in Fig. 15. All links were configured with the error scenario of Case II. As expected, the reliability of the tail device is sensitive to the network size. Its reliability decreases when the network size increases, following an exponential trend. This can be explained by the fact that with more hops, the packet will be more likely to be affected by interference since it will be transmitted through the medium more times.

The study about the impact of the interference over the network links is fundamental to industrial applications. Given a specific scenario, the network designer may estimate actions during the early design phase to guarantee the dependability requirements. Thus, the second experiment conducted in the line topology has the goal of evaluating such an impact. We assume the same topology described in Fig. 14. The links were configured with the error probability of Case I (the most optimistic one). To evaluate the impact of interference, four sub-scenarios were adopted: each one has a poor quality link (configured with the error probability of Case IV). The results are described in Fig. 16 and Table 4.

According to Fig. 16, the transmitter of the poor quality link always has a shorter battery duration than the other devices in the network. This behavior can be attributed to the fact that these devices realize more retransmissions in harsh environments and consequently consume more energy. Such behavior is presented in a more moderate way on the receiver of the poor quality link, since it receives more packets (due to the retransmissions) than do the other devices.

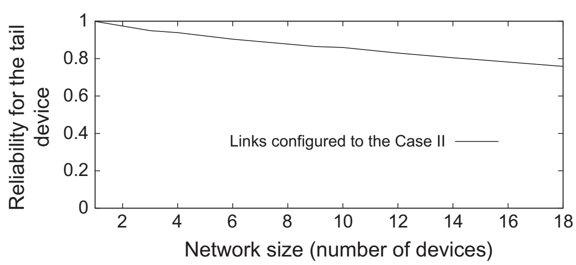

Fig. 15. Reliability evaluation for the tail device.

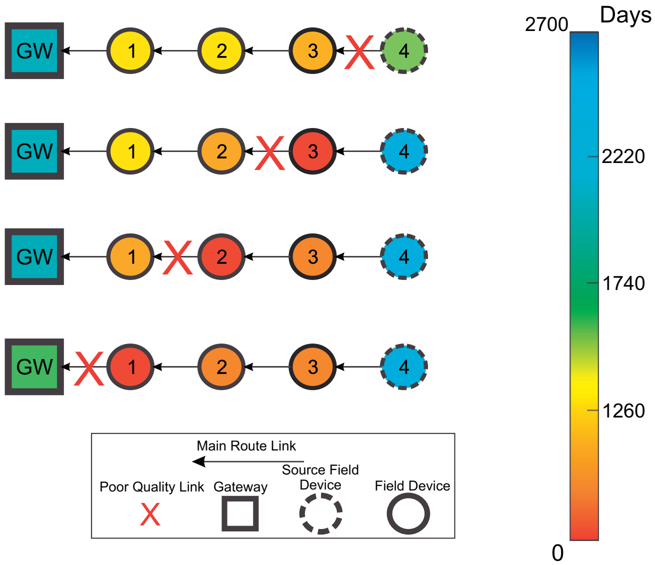

Fig. 16. Influence of a poor quality link on the energy consumption.

Table 4
Influence of poor quality link on the reliability of tail device.

Poor quality link	$Fd_4 \rightarrow Fd_3$	$Fd_3 \rightarrow Fd_2$	$Fd_2 \rightarrow Fd_1$	$Fd_1 \rightarrow GW$
Reliability	63.85%	63.80%	64.10%	64.25%

In the line topology experiment, the source field device has a slower rate of energy consumption than the other devices since it does not receive any data packets: it only transmits them. The gateway has a similar behavior, for it only receives data packets, it does not transmit them. The devices between the poor quality link and the gateway have a lower consumption since the packets barely make their way through the faulty link, causing these devices to transmit and receive less. Finally, the devices between the poor quality link and the source field device maintain the same energy consumption pattern. However, it is greater than the energy consumption beyond the poor quality link. This can be explained by the fact that the devices placed between the poor quality link and the source field device are responsible for routing packets, thus consuming more energy.

The second experiment also evaluated the influence of interference on the reliability of the tail device. The results are shown in Table 4. Poor quality links are inserted alternately in the topology until reaching the gateway. Due to the symmetry of this approach, it was verified (as expected) that there was the same influence of poor quality links in all four sub-scenarios. Note that the results presented in Table 4 are considered to be the same, since they are within the confidence interval (1%).

In Fig. 15, it was verified that the reliability of tail device for a network with four field devices is around 94%. Thus, comparing this result with the one described in Table 4, it is possible to note that the interference injection reduced the reliability by around 33%. This result shows how important it is to adopt mechanisms to overcome the influence of interference on the network.

5.3. Cluster topology

A cluster topology is usually used when there is a need to partially segregate the network. Each cluster may assume specific duties, for example, monitoring specific equipment, traffic prioritization, providing redundancy (new paths), etc. Fig. 17 describes a typical WirelessHART cluster topology. For the sake of clarity, we assume that only the end devices are data sources. The clusters communicate with each other through router devices.

Based on the cluster topology, it is possible to evaluate an important feature presented in the WirelessHART standard: temporal redundancy. According to that standard, if there is an alternative route for forwarding the message, then the device should send the same packet through the main and alternate routes. Two slot times are reserved for the main route and one slot time for the alternate route. If no ACK is received through the main route (first slot time), the packet is retransmitted on the same route (second slot time). There is no retransmission of data packets in alternate routes. Note that the main and alternative routes of source field devices are highlighted in Fig. 17.

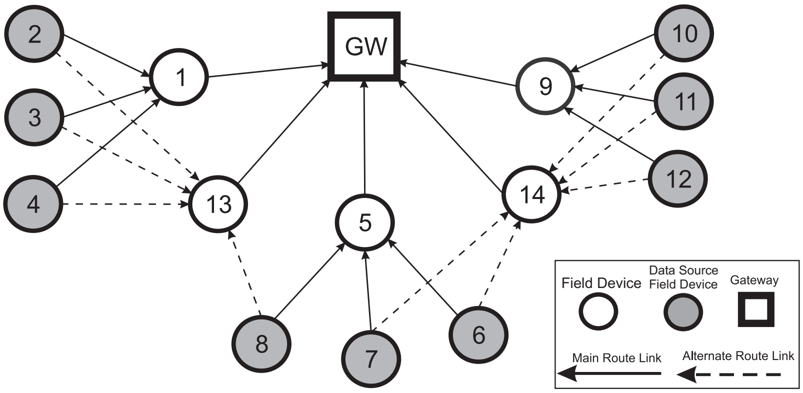

Fig. 17. Cluster topology.

We will use an example to clarify the notation described in Fig. 17. Assuming field devices 2 and 6 are the sources of the data, it is possible to generate the corresponding scheduling according to Table 5. By implementing this schedule we can demonstrate the flexibility of the proposed module in terms of supporting simultaneous transmission on different channels. Additionally, the usage of the main and alternate routes can also be noted on the schedules. For the field devices 2 and 6 we have:

- **Field device 2**
  - Main route: $Fd2 \rightarrow Fd1 \rightarrow GW$.
  - Alternate route: $Fd2 \rightarrow Fd13 \rightarrow GW$.
- **Field device 6**
  - Main route: $Fd6 \rightarrow Fd5 \rightarrow GW$.
  - Alternate route: $Fd6 \rightarrow Fd14 \rightarrow GW$.

The experiments conducted in this topology were intended for the evaluation of the impact of different interference patterns on the mechanism of redundant routes. The metrics adopted were energy consumption and the reliability of the devices. The topology studied was divided into three clusters according to Fig. 18. The links of Clusters I, II and III were configured with the error probabilities of Case I, Case III and Case IV, respectively. The error probability of each cluster was applied to all the links originating from a device belonging to that cluster. As the clusters are similar, then the same error probabilities will cause a also similar behavior among the clusters. Thus we chose to put three different error probabilities in order to show a diversity of behavior on the same experiment, making it easier to compare such behaviors. Additionally in a real scenario, e.g., refinery sites, it is possible to have different error probabilities along the plant, as is the case.

The results about the energy consumption are summarized in Fig. 18. In general, the data source field devices presented lower energy consumption than the router devices. This was because the former only transmit data packets but do not receive them, whereas the routers concentrate the data flow of the other devices, leading to greater energy consumption.

Another observation is that the energy consumption of the end devices in Clusters II and III is greater than for those of Cluster I. The explanation for this is the poor quality links configured in Clusters II and III, which cause a greater number of retransmissions.

Table 5
Scheduling for field devices 2 and 6.

Channel	Slot time										
	0	1	2	3	4	5	6	7	8	9	10
0	2 → 1		2 → 13				13 → 0			14 → 0	
1	6 → 5		6 → 14					5 → 0			
2		6 → 5			1 → 0						14 → 0
3		2 → 1				13 → 0			5 → 0		
4				1 → 0							

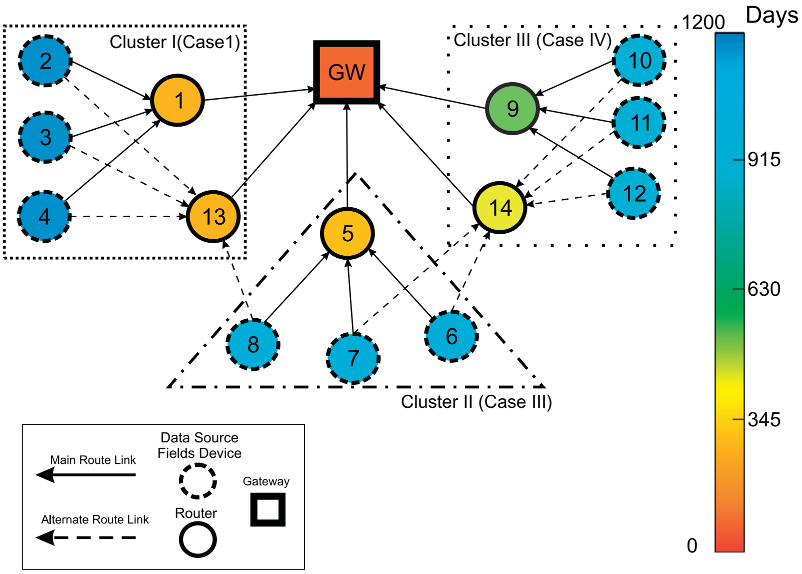

Fig. 18. Energy consumption evaluation for a WirelessHART cluster topology assuming different interference patterns.

Table 6
Reliability evaluation for the data source field devices.

Cluster	Data source field device	Reliability (%)	Error probability
1	2	100.00	Case I
	3	100.00	Case I
	4	100.00	Case I
2	6	44.30	Case III
	7	44.50	Case III
	8	64.90	Case III
3	10	0.05	Case IV
	11	0.10	Case IV
	12	0.00	Case IV

Considering only the energy consumption among the router devices, it is possible to note that routers 9 and 14 presented energy savings. Even though they receive packets to route, most of them come with errors due to the poor quality links configured in Cluster III and they are not forwarded to the gateway. Another interesting comparison is between routers 1 and 5. The former receives fewer retransmissions: however, it forwards more packets to the gateway (due to the quality of the link). The opposite behavior is attributed to the latter. Thus, both have a similar energy consumption. The energy consumption of routers 13 and 14 was also analyzed. Both receive packets originating from alternate routes. Despite having more slot times to use, router 14 has consumes less energy than router 13. This was, again, because of the quality of the links in Cluster III. The amount of corrupted packets in router 14 is greater than in router 13, thus the latter transmits many more packets.

The intent of the second part of the experiments was to evaluate the reliability of the data source field devices under interference. The results are described in Table 6.

The data sources in Cluster I presented a high reliability despite the 2-hop path to the gateway. This result is attributed to the alternate route which elevates the success probability of transmitting information and the excellent quality of the link. According to the results described in the star topology (1-hop path), it was expected that the reliability would decrease for a 2-hop path. This behavior only occurred in Clusters II and III because their error probabilities are very pessimistic.

An interesting result was observed in router 8. In spite of its links having a pessimistic error probability, it uses an alternate path to router 13. The path between router 13 and the gateway uses a very optimistic link, whose probability of delivering a packet is very high. Thus, the difference between the results for the routers in Cluster II justifies the importance of using an alternate path. The poor results from Cluster III occur due to the combination of a very pessimistic error probability and a two-hop path to the gateway. Note that the results of Cluster II had a better performance due to the error probabilities' being less pessimistic than Cluster III.

6. Conclusion

In this article, we presented a WirelessHART simulator based on NS-3[1] and a more realistic physical layer. Aiming at validating the proposal, we conducted simulation experiments (under different error probabilities) to evaluate the energy consumption, end-to-end reliability, the behavior of the network upon the injection of noise, the impact of the number of hops on the reliability level, and the impacts of the existence of alternate routes. The scheduling and routing were implemented entirely based on the suggestions presented in the WirelessHART specification.

The ability to configure different error probabilities for each communication channel in each link at any given simulation time, allows the WirelessHART simulator to evaluate more realistic scenarios. All experiments are feasible for reproduction and may also serve as a basis for evaluating emerging industrial applications.

As future research, we intend to finish the architecture proposed in Fig. 4. Also we intend to develop new scheduling algorithms and submit them to the simulator in order to compare them to the current state of the art scheduling algorithms. Later, we intend to develop the interaction between the simulator and real WirelessHART devices in order to achieve more realistic simulations.

References

[1] Gungor VC, Lambert FC. A survey on communication networks for electric system automation. Comput Networks: Int J Comput Telecommun Network 2006;50:877–97.
[2] Zhao G. Wireless sensor networks for industrial process monitoring and control: a survey. Network Protoc Algorithms 2011;3:46–63.
[3] Johnstone I, Nicholson J, Shehzad B, Slipp J. Experiences from a wireless sensor network deployment in a petroleum environment. In: Proceedings of the 2007 international conference on wireless communications and mobile computing, IWCMC '07. ACM, New York; 2007. p. 382–7.
[4] Silva I, Guedes LA, Portugal P. Emerging technologies for industrial wireless sensor networks. IGI Global; 2013. p. 343–59.
[5] Bai H, Atiquzzaman M. Error modeling schemes for fading channels in wireless communications: a survey, communications surveys tutorials. IEEE 2003;5:2–9.
[6] Willig A, Kubisch M, Hoene C, Wolisz A. Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer. IEEE Trans Ind Electron 2002;43:1265–82.
[7] Song J, Han S, Mok A, Chen D, Lucas M, Nixon M, et al. WirelessHART: applying wireless technology in real-time industrial process control. In: Proceedings of the 2008 IEEE real-time and embedded technology and applications symposium, RTAS '08. IEEE Computer Society, Washington, DC, USA; 2008. p. 377–86.
[8] Coulson G, Porter B, Chatzigiannakis I, Koninis C, Fischer S, Pfisterer D, et al. Flexible experimentation in wireless sensor networks. Commun ACM 2012;55:82–90.
[9] Bergamini L, Crociani C, Vitaletti A, Nati M. Validation of WSN simulators through a comparison with a real testbed. In: Proceedings of the 7th ACM workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, PE-WASUN '10. ACM, New York, NY, USA; 2010. p. 103–4.
[10] Weingärtner E, vom Lehn H, Wehrle K. A performance comparison of recent network simulators. In: Proceedings of the IEEE international conference on communications 2009 (ICC 2009). IEEE, Dresden, Germany; 2009. p. 1–5.
[11] Font JL, I nigo P, Domínguez M, Sevillano JL, Amaya C. Architecture, design and source code comparison of ns-2 and ns-3 network simulators. In: Proceedings of the 2010 spring simulation multiconference, SpringSim '10. Society for Computer Simulation International, San Diego, CA, USA; 2010. p. 109:1–109:8.
[12] I.E.C.C. (IEC). IEC 62591: Industrial communication networks—Wireless communication network and communications profiles—WirelessHART; 2010.
[13] De Biasi M, Snickars C, Landernas K, Isaksson A. Simulation of process control with WirelessHART networks subject to packet losses. In: Automation science and engineering, 2008. CASE 2008. IEEE international conference on; 2008. p. 548–53.
[14] De Biasi M. Implementation of a WirelessHART simulator and its use in studying packet loss compensation in networked control. Master's thesis, KTH Electrical Engineering; 2008.
[15] Shah K, Seceleanu T, Gidlund M. Design and implementation of a WirelessHART simulator for process control. In: Industrial embedded systems (SIES), 2010 international symposium on; 2010. p. 221–4.
[16] De Dominicis C, Ferrari P, Flammini A, Sisinni E, Bertocco M, Giorgi G, et al. Investigating WirelessHART coexistence issues through a specifically designed simulator. In: Instrumentation and measurement technology conference, 2009. I2MTC '09. IEEE; 2009. p. 1085–90.
[17] Ferrari P, Flammini A, Rizzi M, Sisinni E. Improving simulation of wireless networked control systems based on WirelessHART. Comput Stand Inter 2013:605–15.
[18] Zand P, Dilo A, Havinga PJM. Implementation of WirelessHART in NS-2 simulator. In: Proceedings of the 17th IEEE international conference on emerging technologies and factory automation, ETFA 2012, Krakow, Poland. IEEE Industrial Electronics Society, USA; 2012. p. 1–8.
[19] The Network Simulator 3 (NS-3) website (<http://www.nsnam.org/>); 2013.
[20] Ramachandran I, Das AK, Roy S. Analysis of the contention access period of IEEE 802.15.4 MAC. ACM Trans Sensor Networks (TOSN) 2007;3:4.
[21] Casilari E, Cano-García JM, Campos-Garrido G. Modeling of current consumption in 802.15.4/ZigBee sensor motes. Sensors 2010;10:5443–68.
[22] Nobre M, Silva I, Guedes L, Portugal P. Towards a WirelessHART module for the ns-3 simulator. In: Emerging technologies and factory automation (ETFA), 2010 IEEE conference on; 2010. p. 1–4.
[23] Stoffers M, Riley G. Comparing the ns-3 propagation models. In: Modeling, analysis simulation of computer and telecommunication systems (MASCOTS), 2012 IEEE 20th international symposium on; 2012. p. 61–7.
[24] Han B, Lee S. Efficient packet error rate estimation in wireless networks. In: Testbeds and research infrastructure for the development of networks and communities, 2007. TridentCom 2007. 3rd International conference on; 2007. p. 1–9.
[25] Willig A, Ebert J-P. A Gilbert–Elliot bit error model and the efficient use in packet level simulation. Technical report TKN-99-002. Technical University Berlin; 1999.
[26] CC2500 Datasheet low-cost low-power 2.4 GHz RF transceiver. Texas Instruments; 2005.
[27] Silva I, Guedes L, Vasques F. A new AODV-based routing protocol adequate for monitoring applications in oil amp; gas production environments. In: Factory communication systems (WFCS), 2010 8th IEEE international workshop on; 2010. p. 283–92.
[28] Wang H, Moayeri N. Finite state Markov channel—a useful model for radio communication channels. IEEE Trans Vehicular Technol 1995;44:163–71.
[29] Bhagwat P, Bhattacharya P, Krishna A, Tripathi SK. Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs. Wireless Networks 1997;3:91–102.
[30] Fantacci R, Scard M. Performance evaluation of preemptive polling schemes and ARQ techniques for indoor wireless networks. IEEE Trans Vehicular Technol 1996;45:248–57.

[1] Source code available on http://www.dca.ufrn.br/∼marcelonobre.

Marcelo Nobre received his M.Sc. in Electrical and Computer Engineering from the Federal University of Rio Grande do Norte (UFRN). He is a professor at the Federal Institute of Education, Science and Technology of Rio Grande do Norte (IFRN). His research interests include industrial wireless networks and network simulations.

Ivanovitch Silva received the licenciate, M.Sc., and Ph.D. degrees in Electrical and Computer Engineering from the Federal University of Rio Grande do Norte, Natal, Brazil, in 2006, 2008, and 2013. Currently, he is an assistant professor in the Institute Metropole Digital of the Federal University of Rio Grande do Norte. His research interests include industrial wireless communications systems, dependability and performability modeling and evaluation, and Internet of Things applications for critical scenarios.

Luiz Affonso Guedes graduated in electrical engineering from the Federal University of Pará, Brazil, in 1987, and received a PhD in Computer Engineering from the State University of Campinas, Brazil, in 1999. Since 2003, he has been an associate professor in the Department of Computer Engineering at the Federal University of Rio Grande do Norte, Brazil. His main research interests include performance and reliability analysis of wireless sensor networks and industrial systems.