

## Article

# Reliable Link Level Routing Algorithm in Pipeline Monitoring Using Implicit Acknowledgements

Carlos Egas Acosta. <sup>1,2</sup>, Felipe Gil-Castiñeira <sup>2,\*</sup>, Enrique Costa-Montenegro <sup>2</sup> and Jorge Sá Silva <sup>3</sup>

<sup>1</sup> GI-IoE Internet of Every Things Research Group, Escuela Politécnica Nacional, Quito E11-253, Ecuador; carlos.egas@epn.edu.ec

<sup>2</sup>atlanTTic Research Center for Telecommunication Technologies, Universidade de Vigo, Information Technologies Group, 36310 Vigo, Spain; kike@gti.uvigo.es

<sup>3</sup> Department of Electrical and Computer Engineering, INESC Coimbra, University of Coimbra, P-3030 Coimbra, Portugal; sasilva@deec.uc.pt

\* Correspondence: xil@gti.uvigo.es; Tel.: +34-986-818665

**Abstract:** End-to-end reliability for Wireless Sensor Network communications is usually provided by upper stack layers. Furthermore, most of the studies have been related to star, mesh, and tree topologies. However, they rarely consider the requirements of the multi-hop linear wireless sensor networks, with thousands of nodes, which are universally used for monitoring applications. Therefore, they are characterized by long delays and high energy consumption. In this paper, we propose an energy efficient link level routing algorithm that provides end-to-end reliability into multi-hop wireless sensor networks with a linear structure. The algorithm uses implicit acknowledgement to provide reliability and connectivity with energy efficiency, low latency, and fault tolerance in linear wireless sensor networks. The proposal is validated through tests with real hardware. The energy consumption and the delay are also mathematically modeled and analyzed. The test results show that our algorithm decreases the energy consumption and minimizes the delays when compared with other proposals that also apply the explicit knowledge technique and routing protocols with explicit confirmations, maintaining the same characteristics in terms of reliability and connectivity.

**Keywords:** access protocols; industrial communication; wireless sensor networks; linear networks



**Citation:** Acosta, C.E.; Gil-Castiñeira, F.; Costa-Montenegro, E.; Silva, J.S. Reliable Link Level Routing Algorithm in Pipeline Monitoring Using Implicit Acknowledgements. *Sensors* **2021**, *21*, 968. <https://doi.org/10.3390/s21030968>

Academic Editor: Ionut Anghel

Received: 31 December 2020

Accepted: 28 January 2021

Published: 1 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Many industrial applications require the deployment of sensor networks that follow straight lines. For example, in applications such as monitoring water, oil, or gas pipelines, roads, tunnels, borders, etc., wireless communication technologies provide clear advantages (such as self-organization, rapid deployment, cost, or flexibility) for the creation of highly reliable and self-healing industrial systems that can respond to new events [1]. This problem has been studied in recent years, and different authors proposed the deployment of Linear Wireless Sensor Networks (LWSN) [2], characterized by sparse node deployment along the linear infrastructure. It is assumed that nodes cannot move after being deployed. One of the main constraints in linear topology is the reduced number of neighbor nodes that limits the feasible transmission routes [2]. Such limitations make traditional solutions for Wireless Sensor Networks (WSNs) inapplicable to LWSNs [3]. For example, complex routing protocols may adversely affect the performance of the sensor network [4], and the limited number of transmission routes may increase the data loss probability [5].

Nevertheless, there are still important challenges that must be addressed. For example, in order to avoid changing batteries (which may be a complex or expensive procedure), power consumption must be optimized [6]. Such infrastructures may cover hundreds or thousands of kilometers, so it is necessary to consider end-to-end delay, node placement, and routing.

Different proposals have been developed to optimize the power consumption of the nodes and minimize delays. Many studies have focused on minimizing energy consumption during transmission and reception [7], while others have focused on minimizing the processing of information in the sensor node [8], a task which can also increase the delay.

In LWSNs, nodes may be placed in a thin (one dimension, forming a line) or thick (two or three dimensions around a line) configuration, and can have identical (one-level) or differentiated (multiple-level, creating a hierarchical network) functionalities [9]. In the applications such as pipeline monitoring, a thin one-level network is cheaper and simpler to build and maintain, this being the configuration considered in this paper to propose solutions for the minimization of energy consumption and delays (while maintaining reliability and connectivity) in LWSNs. Nevertheless, a direct transposition of classical protocols on LWSN is not efficient, as linear topologies are the worst cases for hierarchical networks (spoilage) and for stochastic distribution (time for reparation) [10]. Hence, routing protocols designed for nonlinear WSNs should not be directly applied to LWSNs with hundreds or thousands of nodes due to their special topology and application requirements [11]. For example, IEEE 802.15.4 nodes can use ACKs to confirm the reception of a frame. This technique is known as explicit confirmation (eACK), but the transmitting node can also receive an implicit confirmation (or iACK) when it receives the same information after the receiving node retransmit it to the next hop.

By using implicit acknowledgements (iACKs), we can address two challenges: reducing energy consumption and latency. This technique helps to decrease the packet exchange between nodes, and to reduce the communication stack by removing most of the functions usually implemented at network level layer (OSI layer 3). This is possible because in a wireless network when a node transmits a frame, all the active nodes in the coverage area receive it and in addition, in a linear topology, the routing is very simple, as packets can only go in one of the two directions of the linear infrastructure, and nodes just have to decide if they have to retransmit a packet or not. Thus, networking functions that are typically implemented in a separated layer, can now be integrated within layer 2 (link) functionalities. The proposed algorithm for LWSN was validated through real-world testbeds and contributes to the following aspects:

- Reliability of the transmission, using iACK.
- Reduction of the delay time caused by using acknowledgement frames.
- Alternative nodes are selected to transmit when failed nodes or links occurs, without using routing.
- Reduction of computation in the node thanks to the elimination of routing tasks.
- Automatic assignation of addresses to nodes using link level processes.
- Reduction of the energy consumption during the end-to-end reliability transmission process.
- Improvement of the scalability with a large number of sensor nodes in LWSN, allowing for the insertion of new nodes automatically.
- Maximization of the entire system's lifetime due to the energy efficiency in each node.

The paper is organized as follows: Section 2 discusses the related research; Section 3 presents the proposed algorithm; Section 4 analyzes the energy and the transmission time; and Section 5 describes a real implementation. We conclude our paper in Section 6.

## 2. Related Work

WSNs have been widely studied, and there are different protocols that have been designed to support a wide range of network topologies (star, mesh, etc.), for example, 6LowPAN and ZigBee, which can use IEEE 802.15.4 for the lower layers. Nevertheless, linear networks have very specific requirements which are not well addressed by generic protocols, as shown in [12], where the authors review and compare different aspects of routing protocols (addressing policies, discovery mechanisms, routing, etc.) designed specifically for LWSNs or for more traditional WSNs, by analyzing metrics such as end-to-end delay, reliability, or deployment requirements. Therefore, we can conclude that LWSNs

require specifically designed communication stacks. In this sense, there are different proposals for new MAC layer protocols for LWSN. LC-MAC [13] is a long-chain MAC protocol where the forwarding nodes schedule the transmission of packets to reduce the delay of an end-to-end extended sensor network without sacrificing energy efficiency. CH-MAC, a delay-aware MAC protocol with implicit control frame Request to Send (RTS) employed in the medium access control, is proposed and compared with eACK in [14]. The authors conclude that it prevents the “avalanche” effect, and the use of iACK optimizes the use of energy. In [15] a token-based MAC protocol for linear sensor networks has been proposed to manage the access to the medium and improve the network performance. Also, many protocols have been presented to optimize the energy consumption of large WSNs. For example, in [16] MI-MAC, a new link level protocol, is proposed. Finally, reference [17] includes an analysis and comparison of different MAC protocols for linear LWSN such as Dual mode MAC, Disc-MAC, LC-MAC, MFT\_MAC, CMAC-T, AC\_MAC/DPM, and WiWi protocols.

Different routing protocols have also been suggested to meet the specific LWSN needs. In [10] a robust discovery, addressing and routing protocol for dynamic LWSN is proposed. The application of LWSN to rail transport is considered in [18], and energy efficient routing protocols for linear wireless sensor networks are also studied. The authors recommend two energy efficient routing protocols, Minimum Energy Relay Routing (MERR) and Adaptive MERR (AMERR). A Chain-based Anonymous Routing (CAR) unicast-based broadcast to send data is studied in [19]. In this proposal, each node can only sense the adjacent nodes on the same chain, and the traceability of the system has been enhanced; thereby increasing network security. In [20] a new routing algorithm is proposed for wireless sensor networks for post-disaster road monitoring. For this approach, the nodes are arranged in a linear topology. It includes route establishment, packet forwarding, and route maintenance. This proposal assumes that the nodes are previously identified in sequence. Finally, the AODV Routing Algorithm in Linear Topological Wireless Sensor is studied in [21].

Transport and application protocols have been also developed for WSNs to guarantee reliability, flow control, and congestion control. However, large-scale networks require a complex implementation complicated by the limited processing and power resources of the nodes. For pipeline monitoring the main issue is reliability, since controlling the monitoring period helps to minimize other problems such as congestion or flow control. For example, a comparative analysis of the transport and application protocols CODA, ESTR, RMST, PSFQ, GARUDA, and ATP, proposed for large-scale WSN networks, is presented in [22]. The comparison includes the use of end-to-end ACK and NACK, but the studied protocols do not use eACK or iACK at the link level. The Rate-Controlled Reliable Transport (RCRT) protocol is presented in [23]. It uses a base station or receiver to discover missing packets and explicitly requests the packets from the sensor nodes for end-to-end recovery. The Stream Control Transmission Protocol-WAN (SCTP-WSN) [24] adopts the concept of Multihoming and adopts the SCTP transport protocol combined with the AODV routing protocol to provide network reliability with eACK. GARUDA [25] studies the reliability of the delivery of point-to-multipoint data in the downstream direction, from a sink to the sensor nodes using NACK. The Aggregation and Transmission Protocol (ATP) transport protocol [26] includes an end-to-end feedback control algorithm assisted by the receiving node and the network to counteract packet loss, using selective ACK.

Nevertheless, using end-to-end mechanisms for reliability has also drawbacks, such as a higher delay because of the end-to-end retransmissions. In [27], the authors study the advantage of using hop-to-hop, instead of end-to-end acknowledgment, and in [28,29] the authors study the advantage of using iACK, instead of eACK in wireless sensor networks. Based on these conclusions, our proposal decreases the energy consumption and minimizes the delays using iACK for the communication. It provides the same reliability, end-to-end connectivity, and scalability as previous proposals that implement high level routing protocols and eACK mechanisms.

We take advantage of the previous work and we propose a new cross-layer algorithm for LWSNs, which unifies MAC, routing, and address assignment. This algorithm also provides end-to-end reliability and solves the hidden nodes problem without requiring ACK, RTS, or CTS signaling. Nevertheless, our proposal combines several advantages compared with previous proposals, such as the presented in [12], where the authors review different LPWAN protocols considering static and mobile nodes.

Table 1 compares our proposal with the protocols presented for static nodes and some other proposals specifically designed for large scale linear infrastructures. It is important to note that all the studied protocols at the link level use eACK, while our proposal uses iACKs because of the benefits provided for linear infrastructures.

Thus, our work is essential to facilitate the deployment of simple and cheap LWSNs, pave the way for the implementation of new services and applications [30] and take advantage of new paradigms such as fog computing [31], cloud computing [32], and the analysis of large amounts of data [33] also in LWSNs.

Table 1. Comparison of LPWAN protocols.

	Address Autoconfig.	Same Net. and Link Addr.	eACK	iACK	Linear Topology	Clustering	End-to-End Reliability	Topology Discovery	Energy Efficient	Considers End-to-End Delay	Large Scale	Access Channel	Protection against Link and Node Failure
This proposal	yes	yes	no	yes	yes	no	yes	yes	Yes	yes	yes	CSMA/CA	yes
Application and transport level													
RCRT [23]	no	no	yes	no	no	No	yes	no	No	yes	no	n.a.	yes
GARUDA [25]	no	no	yes	no	no	Yes	yes	no	n.a.	no	yes	n.a.	yes
ATP [26]	no	no	yes	no	no	No	yes	no	No	yes	yes	n.a.	no
SCTPWSN [24]	no	no	yes	no	no	No	yes	yes	n.a.	no	yes	n.a.	yes
Network level													
SimpliMote [34]	no	yes	no	no	yes	No	no	yes	yes	no	yes	n.a.	yes
MERR [35]	no	no	no	no	yes	No	no	no	yes	no	yes	n.a.	no
LDG [36]	no	no	no	no	no	Yes	no	yes	yes	yes	yes	n.a.	no
DiscoProto [10]	yes	yes	no	no	no	Yes	no	yes	no	yes	yes	TMAC	yes
ACO&GA [37]	no	no	no	no	yes	No	no	no	yes	no	yes	n.a.	no
LBHRP [2]	no	no	no	no	no	Yes	no	yes	no	no	yes	TDMA	yes
ROLS [6]	yes	no	no	yes	yes	Yes	no	no	no	no	yes	n.a.	yes
WTDP [29]	yes	no	no	no	no	No	no	yes	no	no	no	aloha	no
Link level													
PRIMAC [38]	no	no	yes	no	yes	No	no	yes	no	no	yes	n.a.	no
SA-MAC [39]	no	no	yes	no	yes	Yes	yes	yes	no	yes	no	TDMA	no
RP MAC [40]	no	no	yes	no	yes	No	no	no	yes	yes	no	n.a.	no
HEPRMAC [41]	no	no	no	no	no	No	no	no	yes	yes	no	TDMA	no
Token [42]	no	no	yes	no	yes	Yes	no	no	n.a.	yes	no	Token	no
LCMAC [13]	no	no	yes	no	no	No	yes	no	yes	yes	yes	n.a.	no
MIMAC [16]	no	no	yes	no	yes	No	no	no	yes	yes	yes	CSMA/CA	no

### 3. Proposed Algorithm

Our algorithm is specifically designed to address the capture of data and its transmission in linear infrastructures. Although many infrastructures are not strictly linear (e.g., a pipeline or a road may have branches), they can be separated in linear segments. Therefore, we make the following assumptions:

- We consider a linear structure, without branches, with  $n$  nodes and two border nodes or base stations,  $v_0$  and  $v_{n+1}$ , which are responsible for sending information to the central monitoring using long-range communication protocols.
- Each node works with the IEEE 802.15.4 standard in beaconless mode (CSMA/CA).
- Each node has a coverage area of 50 m (a reasonable distance for an IEEE 802.15.4 network in the 2.4 GHz band [43]), without obstacles and transmission conditions in the line of sight, and is associated with four nodes, two on the left and two on the right.
- The nodes are deployed with a separation of 25 m. Therefore, a node has at least 4 nodes in range.
- The participating nodes use a predefined PAN identifier to prevent the connection to other nodes that are not part of the linear infrastructure.
- Each node has an identifier assigned sequentially and automatically using processes at the link level, as indicated in [44].
- The bits of the header identified for future use are used to implement the control messages required by the proposed algorithm.

We propose using LWSN topologies as the one shown in Figure 1 which consists of  $n + 2$  nodes, sequentially identified from 0 to  $n + 1$ . The sequential assignment of the identifiers is an essential feature on this research approach and solved as detailed in [44].

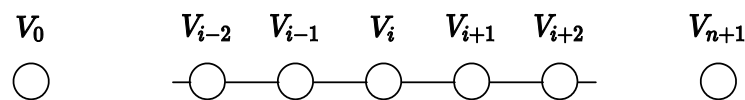


Figure 1. Linear Wireless Sensor Networks.

Nodes  $v_0$  and  $v_{n+1}$  are connected to the control center and do not have any limitation on energy or communication technologies, while nodes  $v_1$  to  $v_n$  are nodes with sensors that capture information and also retransmit the frames of the neighboring nodes. Sensor nodes generate traffic when an event happens (e.g., the rupture of oil pipelines, unauthorized entry into borders, etc.). The information generated by node  $v_i$  is transmitted to the neighboring nodes, creating two traffic streams simultaneously,  $P_l$  (left direction) and  $P_r$  (right direction), so it can reach the control center through  $v_0$  and  $v_{n+1}$ . This feature can be used to introduce redundancy in the network, but the operation with both flows is similar, so we are going to study only the  $P_r$  flow. The nodes are in fixed locations, and therefore, the topology does not change in time. Modifications to the topology will only be made when a node or several nodes fail, or a link failure occurs. This feature allows the use of simpler routing algorithms and facilitates the location of nodes.

#### 3.1. Algorithm Implementation

Each node includes four buffers ( $buffer_{tx}$ ,  $buffer_{rx}$ ,  $buffer_{ok}$ , and  $buffer_{adj}$ ) and will always check if the received frame is stored in one of them to perform the appropriate actions.  $buffer_{ok}$  stores the transmitted frames that have successfully arrived at the destination node;  $buffer_{tx}$  stores the transmitted frames from which a successful transmission has not been confirmed;  $buffer_{rx}$  stores the frames received; and  $buffer_{adj}$  stores the frames that have been sent by an adjacent node. The node  $v_j$  is adjacent to node  $v_i$ , when  $|j - i| \leq 2$ .

As shown in Figure 2, the transmitting node  $v_i$  is called TXN, node  $v_{i+1}$  is called INTN (intermediate node) and node  $v_{i+2}$  is called RXN (reception node). When a TXN node transmits a frame, a  $timer_{tx}$  is started. When an INTN node receives a frame, it starts a

$timer_{int}$  whose minimum value is equal to the time necessary for the RXN node to receive and retransmit the frame. When the node  $v_i$  detects an event, it sends information to the control center, acting as a TXN node, transmitting a broadcast frame. When node  $v_{i+2}$  receives a broadcast frame with flow  $P_r$  (towards  $v_{n+1}$ ) and if source address is  $v_i$ , the node  $v_{i+2}$  it will work as a RXN node. The frame will include the variable  $FailN$  that will have been activated if the node  $v_{i+2}$  fails, and the variable  $ChangeF$  that indicates a change in the data flow; both variables are initialized with "FALSE".

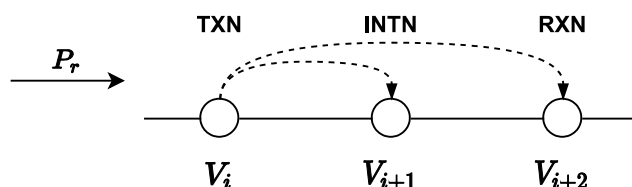


Figure 2. Node identification.

There are two options to acknowledge the reception of the frame by the RXN node: using explicit ACKs (eACK) or implicit ACKs (iACK). In the first case, when the RXN node successfully receives a frame from TXN, it responds with an ACK, as shown in Figure 3. This approach increases the usage of energy (to explicitly transmit the ACK) and may even increase the latency if we wait for the ACK before transmitting a new frame. In 802.15.4, the additional time that node  $v_i$  must wait to transmit another frame when an ACK is used is defined by the time to pass from transmission mode to reception (TTA), the duration time of the ACK frame (which involves the additional back off time in the node  $v_i$  associated with the retransmission of the frame to the  $v_{i+2}$  node), and the time for the Clear Channel Assessment (CCA).

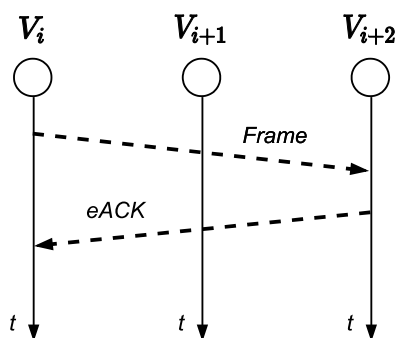


Figure 3. eACK.

Nevertheless, when the RXN node sends again the frame towards the destination ( $v_{n+1}$ ), the nodes in the surrounding ( $v_i$ ,  $v_{i+1}$ ,  $v_{i+3}$  and  $v_{i+4}$ ) will receive the information. Therefore, TXN will know that its frame has arrived at node  $v_{i+2}$  because it has retransmitted it. This works as an implicit acknowledgment (iACK) of the frame sent by node  $v_i$  and avoids the need for the transmission of an ACK from  $v_{i+2}$  (RXN), as shown in Figure 4.

In order to summarize, Figure 5 presents a typical scenario. When node  $v_i$  detects an event, its objective is to transmit this information to the central station, so it transmits the broadcast frame to the node  $v_{i+2}$ , and the node  $v_{i+1}$  also receives the frame. The node  $v_i$  stores the frame in  $buffer_{tx}$ . Nodes  $v_{i+1}$  and  $v_{i+2}$  store the received frame in  $buffer_{rx}$  and  $v_{i+2}$  also verifies if it comes from an adjacent node to store it in  $buffer_{adj}$ .

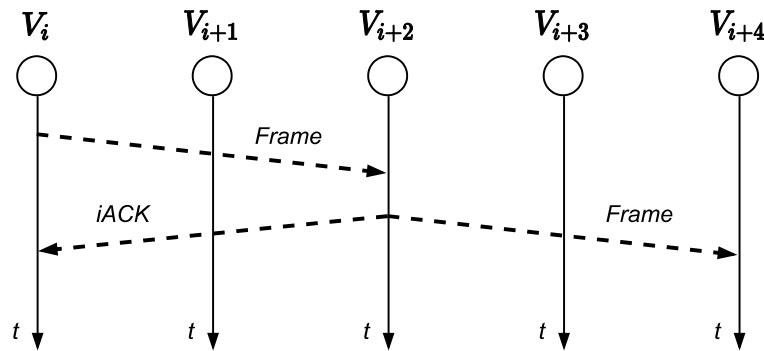
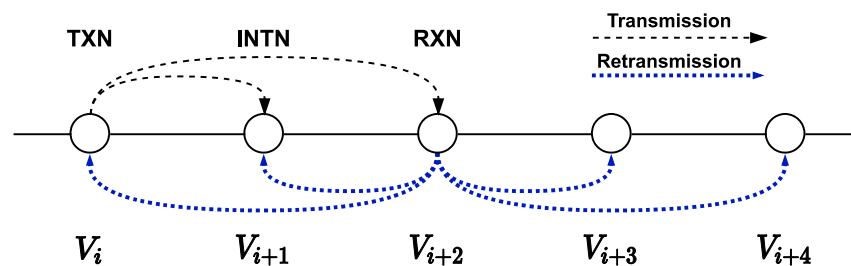


Figure 4. iACK.

Figure 5. Node  $v_i$  transmits to node  $v_{i+2}$ .

The frame retransmission from node  $v_{i+2}$  to node  $v_{i+4}$  is also received by nodes  $v_i$ ,  $v_{i+1}$  and  $v_{i+3}$ . When  $v_i$  receives the retransmission from node  $v_{i+2}$  to node  $v_{i+4}$ , it checks if the frame is stored in one of its buffers. If it is stored in  $buffer_{tx}$ , the node concludes that its transmission was successful without the need of an ACK from  $v_{i+2}$ , so it stores the frame in  $buffer_{ok}$  and eliminates the  $buffer_{tx}$  frame. If the frame was stored in  $buffer_{ok}$ , it is discarded because it is confirmed that the frame was received by  $v_{i+2}$ . When node  $v_{i+1}$  receives this frame, it verifies if the frame is already stored in any of the buffers. If the received frame is stored in  $buffer_{rx}$ , node  $v_{i+1}$  stores the frame in  $buffer_{ok}$  and removes it from  $buffer_{rx}$ . If it is stored in the  $buffer_{ok}$ , it discards it because it is confirmed that the frame was received by  $v_{i+2}$ . Since this frame was not transmitted or retransmitted by the  $v_{i+1}$ , it will not be stored in the  $buffer_{tx}$ .

### 3.2. Algorithm Robustness

A noisy link prevents the signal from being picked up by the receiving node and therefore the frame cannot be recovered and retransmitted to the next node. In 802.15.4 the unslotted mechanism may cause collisions. However, the unsynchronized behavior of the back off mechanism reduces the possibility of simultaneous transmissions, causing the Rx-to-Tx turnaround to be the main reason behind the collisions. The non-negligible Rx-to-Tx turnaround time in the unslotted mechanism may cause collisions between data frames [45].

The behavior of the node in noisy links will depend on whether it is a TXN, INTN, or RXN node. We present some examples that depict how our algorithm solves link problems. Consider the case where  $v_{i+2}$  does not retransmit the frame to  $v_{i+4}$  because it did not receive it or received it with errors. So, the  $v_i$  or  $v_{i+1}$  node will wait a given time for one of them to retransmit the frame to  $v_{i+2}$ . The node  $v_{i+1}$  will wait a time given by  $timer_{int}$ , as seen in Figure 6, to retransmit the unicast frame to  $v_{i+2}$ , storing the frame in  $buffer_{tx}$  and placing the node address  $v_{i+2}$  in the destination direction of the frame. The node  $v_{i+1}$  retransmits first, as  $timer_{tx} > timer_{int}$ , to take advantage of the highest level of the signal that reaches the node  $v_{i+2}$ . This retransmission will be heard by node  $v_i$  concluding that the frame it previously sent did not reach  $v_{i+2}$ , and it will wait until node  $v_{i+2}$  retransmits the frame.



Additionally, when node  $v_{i+2}$  verifies that its address is in the destination address of the frame, it will understand that it is a retransmission made by the INTN node of the frame that TXN node sent to RXN node with broadcast address.

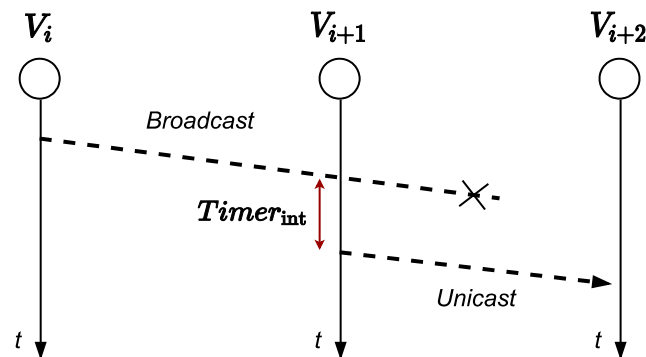


Figure 6. Node  $v_{i+1}$  retransmits to node  $v_{i+2}$ .

When the frame sent by  $v_i$  does not reach  $v_{i+2}$  (Figure 7),  $v_i$  starts a  $timer_{tx}$  to retransmit the frame to  $v_{i+2}$ . After this time,  $v_i$  will understand that its transmission to  $v_{i+2}$  was not successful, regardless of whether the node  $v_{i+1}$  retransmitted or not the frame to  $v_{i+2}$ . Therefore, node  $v_i$  will retransmit the frame to  $v_{i+2}$  a maximum number of times (set by the  $aMaxFrameFRameRetries$  variable) using broadcast destination address in order to make information reach the nodes defined as INTN and RXN. If the maximum number of retransmissions is reached, node  $v_i$  will understand that node  $v_{i+2}$  could not retransmit the frame and consequently will proceed to execute the damaged node or link procedure: changes the  $FailN$  variable to "TRUE" to indicate that the  $v_{i+2}$  node is failing and transmits a unicast frame to the node  $v_{i-1}$  indicating to act as a TXN node so it can transmit the information to node  $v_{i+1}$ . In case node  $v_{i+1}$  also fails,  $v_{i-1}$  assigns the value "TRUE" to the  $FailN$  variable and transmits the frame to the node  $v_{i-1}$ .

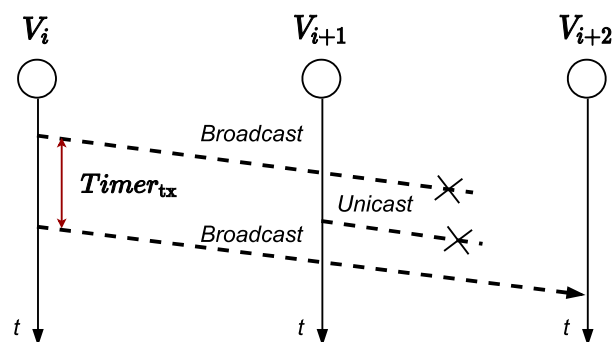


Figure 7. Frame does not reach node  $v_{i+2}$ .

When the frame transmitted by  $v_i$  does not reach  $v_{i+1}$  but it reaches  $v_{i+2}$ ,  $v_{i+2}$  retransmits the frame to  $v_{i+4}$ , so this frame is also received by  $v_{i+1}$ . This node discards the frame because it has determined that the flow is  $Pr$  and that the transmitting node is  $v_{i+2}$ , concluding that the frame goes from  $v_{i+2}$  to  $v_{i+4}$  with destination node  $v_{n+1}$ .

When the frame sent by node  $v_{i-2}$  reaches nodes  $v_i$ , and the  $FailN$  variable is "TRUE", it assumes that nodes  $v_{i+1}$  and  $v_{i+2}$  are damaged and notifies the control station using the opposite route ( $P_1$ ), and assigns "TRUE" to the variable  $ChangeF$ .

If the retransmitted frame by node  $v_{i+2}$  to node  $v_{i+4}$  does not reach  $v_i$  (the situation would be similar for  $v_{i+1}$ ),  $v_i$  waits  $timer_{tx}$  and retransmits the frame. The receivers verify that it has already been stored in  $buffer_{rx}$  or  $buffer_{ok}$  and therefore they discard it. Nevertheless, node  $v_{i+2}$  will retransmit again the frame which will be discarded by the nodes  $v_{i+1}$ ,  $v_{i+3}$  and  $v_{i+4}$  and node  $v_i$  will store it in  $buffer_{ok}$  and then eliminate it from  $buffer_{tx}$ .

When a node  $v_i$  sends information to the control station in the direction of the flow  $P_r$ , and there are at least two damaged nodes, as seen in Figure 8, the last node that works correctly ( $v_{k-2}$ ) will retransmit the frame in the flow towards the node  $v_0$ . At that moment, when the frame passes through node  $v_i$ , this node, knowing that the variable *ChangeF* is “TRUE”, will change the flow direction, starting to use  $P_l$ , as the flow  $P_r$  is interrupted. When a frame arrives at the central station with *ChangeF* set to “TRUE”, it means that two adjacent nodes are damaged.

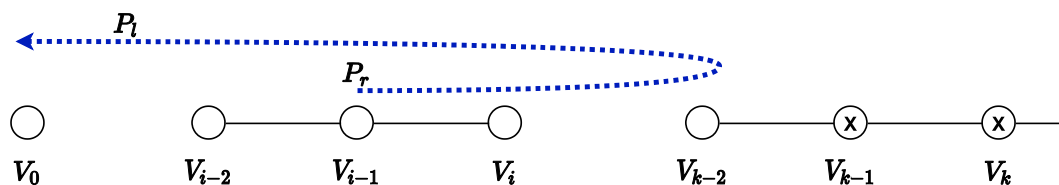


Figure 8. Flow  $P_r$  interrupted.

It might also happen that the node is in a network segment that has no connection with the end nodes  $v_0$  and  $v_{n+1}$ , because the nodes  $v_g$ ,  $v_{g+1}$ ,  $v_{k-1}$  and  $v_k$  are damaged, as seen in Figure 9. Therefore, the frame would be circulating through the flows  $P_r$  and finally by  $P_l$  to be later discarded by  $v_{g+2}$ .

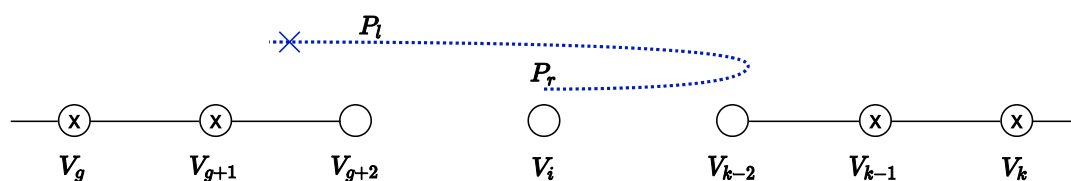


Figure 9.  $P_r$  and  $P_l$  flows are interrupted.

When two nodes want to transmit simultaneously and the nodes are separated by two or three nodes, the problem of the hidden nodes may appear. The collision of the frames due to the problem of the hidden nodes is resolved according to what is indicated in the noisy link scenario.

### 3.3. Traffic Minimization

In linear structures, there is the possibility that several adjacent nodes detect the same event, up to five nodes that are located consecutively. Due to the value of the distance in which the nodes are separated, an event that occurred at a specific location in the linear infrastructure may cause the adjacent nodes to generate the same alarm, at the same moment in time. Therefore, they will try to notify the alarm generated to the central station, by sending a frame using the same predefined data flow,  $P_r$  for example. Our proposal prevents adjacent nodes from sending repeated alarms from the same event. Before sending information related to an event, the nodes will check if that information was already transmitted by an adjacent node, confirming that the frame has already been stored in the  $buffer_{adj}$  and avoid retransmitting it.

### 3.4. Pseudocode and State Diagram

The Algorithm 1 that determines the operation of a node is presented below. All nodes that are part of the linear infrastructure will run the same code.

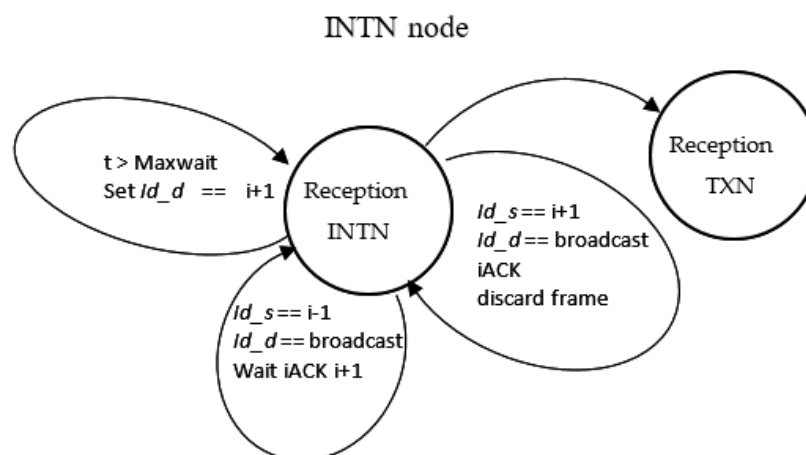
**Algorithm 1** Pseudo-code for our proposal

```

1 Input:  $id_s$  is source address,
2 Input:  $id_d$  is destination address,
3 Input:  $Maxwait$  is maximum wait time for an iACK from of node TXN or INTN,
4 Input:  $FailN$  node is on
5 Input:  $ChangeF$  The direction of traffic flow has not change
6 if frame received then
7 if frame is stored in  $buffer_{adj}$  or  $buffer_{ok}$  then frame is discarded
8 else
9 case  $id_s == i-1$ , and  $id_d == broadcast$ : the node is INTN, and waits iACK from node  $i+1$ :
10 case  $id_s == i+1$ :
11 case  $id_d == broadcast$ : the node behaves as INTN and discards the frame clean  $buffer_{rx}$  and
stores frame in
 $buffer_{ok}$ :
12 case  $id_d == unicast$ : ( $i+3$  node fail) node works as TXN and retransmits the frame with  $id_d =$ 
broadcast,
stores the frame in  $buffer_{tx}$ :
13 case  $id_s == i-2$  and  $id_d == broadcast$ : behaves as an RXN node and retransmits the frame
with  $id_d = broadcast$ , store the frame in  $buffer_{tx}$ :
14 case  $id_s == i+2$  and  $id_d == broadcast$ :
15 if  $FailN == F$  then it has received an iACK, stores frame  $buffer_{ok}$  and transmits the following
frame else
16 case  $ChangeF == F$ , set  $ChangeF = T$ , transmit broadcast frame:
17 case  $ChangeF == T$ , discard the frame:
18 end if
19 end if
20 else
21 case node INTN: retransmit unicast frame to  $i+1$ :
22 case node TXN:
23 if retransmission >  $Maxwait$ . then
24 transmit unicast frame to node  $i-1$  and set  $FailN = T$ 
25 else retransmit frame broadcast
26 end if
27 end if

```

Figures 10–12 also show the state diagram for nodes TXN, RXN, and INTN.



**Figure 10.** INTN node state diagram.

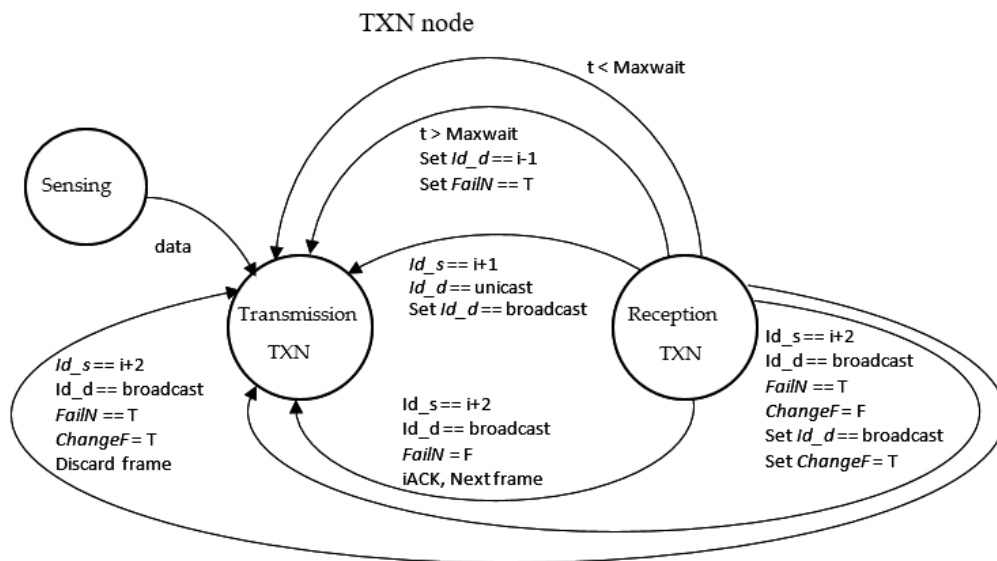


Figure 11. TXN node state diagram.

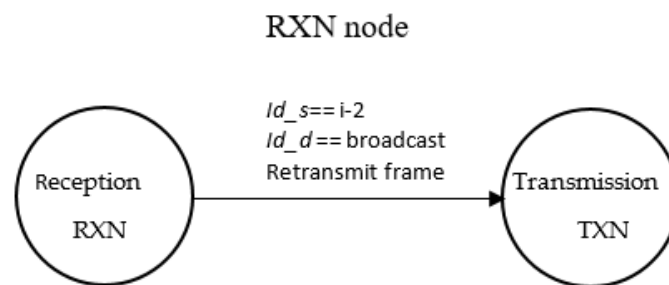


Figure 12. RXN node state diagram.

## 4. Energy and Transmission Time Analysis

### 4.1. System Model

To build the model, we consider a linear structure described in Section 3. For the analysis, ideal conditions are assumed: no collision, no overhead for turning on the transceiver, and prompt response in the MAC layer. Note that these assumptions are especially realistic for many LWSN networks that monitor some parameters to know the state of the infrastructure (pressure, vibrations, leaks, etc.) in vast areas: small amount of traffic, few nodes within the coverage area, environments where the presence of interference is minimal (e.g., in networks for pipe monitoring, road borders, etc.). Our proposal does not support sensors that require large bandwidths, such as cameras, radars, or lidars, that could be used to monitor especially relevant areas. Nevertheless, a separate network could be easily deployed for such traffic in the required locations.

Although the processing of the frame for its forwarding is not negligible, we do not consider that time for the analysis because the delay caused is the same for scenarios that use iACK and eACK. Other considerations are presented below.

- All the frames have the same length, so the transmission times are fixed.
- Distance between transmitter and receiver is 50 m.
- Probability of error when transmitting a frame in each link is the same. We include a model for comparing the latency added by a retransmission using eACK or iACK. Nevertheless, for the sake of simplicity, we did not consider retransmissions in the end-to-end analysis.
- ACK frames have a constant length.
- If the transmitted data collides at the receiver, the *macAckWait* duration (an IEEE 802.15.4 attribute dependent on constants and attributes of the physical layer, which

defines the maximum number of symbols to wait for an ACK frame) for the transmitting node is the sum of the turnaround time of 12  $T_s$  and the ACK duration of 22  $T_s$ , a total of 34  $T_s$  considering  $T_s = 16 \mu s$  and 1 symbol is 4 bits.

- A CCA is successful when no transmission is detected from any node, at the time of initiation of its CCA. In the IEEE 802.15.4 standard, the channel state is averaged over 8  $T_s$ .

#### 4.2. End-to-End Transmission Time

It is important to know how much time is necessary to transmit information from a sensor to the control station. We can approximate the end-to-end delay by the sum of the time between the arrival of information at a node  $v_i$  and the arrival at the  $v_{i+2}$  node [46]. In this scenario the information (alerts) is just transmitted using a single frame, and the probability that other nodes that are part of the lineal infrastructure generate alarms at the same time is very low. Therefore, this simplification will be realistic in many situations.

Then, it is necessary to determine the time to transmit a frame when using explicit ACK, Equation (1), or implicit ACK, Equation (2):

$$de(x) = T_{BO} + T_{CCA} + T_{fra}(x) + T_{TA} + T_{ACK} + T_{IFS} + \tau \quad (1)$$

$$de(x) = T_{fra}(x) + T_{IFS} + T_{TA} + T_{BO} + T_{CCA} + \tau \quad (2)$$

where  $x$  represents the number of bytes that have been received from the upper layer,  $T_{BO}$  is the back off period,  $T_{CCA}$  means Clear Channel Assessment time,  $T_{fra}(x)$  is Transmission time for a payload of  $x$  byte,  $T_{TA}$  is Turnaround time from TX to RX,  $T_{ACK}$  is Transmission time for an ACK,  $T_{IFS}$  (Interframe Space time) is the Frame Processing time,  $\tau$  is Propagation delay.

The back off period is calculated as the product of the number of back off slots and the time for each slot (20 symbols or 320  $\mu s$ ). The number of back off slots is a random number uniformly in the interval  $(0, 2^{BE} - 1)$ , with BE the back off exponent which has a minimum of 3. As we only assume one sender and a Bit Error Rate (BER) of zero, the BE value will not change. Hence, the number of back off slots can be represented as the mean of the interval:  $(2, 2^3 - 1)$  or 3.5. Regarding IFS, Short IFS (SIFS) is used when the Media Access Control Protocol Data Unit (MPDU) is smaller than or equal to 18 bytes (192  $\mu s$ ). Otherwise, Long IFS (LIFS) is used (640  $\mu s$ ). Then, the transmission time of a frame with a payload of  $x$  bytes can be formulated as:

$$T_{fra}(x) = 8 \left( L_{phy} + L_{MACHDR} + L_{address} + x + L_{MACFTR} \right) / R_{data} \quad (3)$$

where  $L_{phy}$  is the Length of the PHY and synchronization header in bytes (6 bytes),  $L_{MACHDR}$  is Length of the MAC header in bytes (3 bytes),  $L_{address}$  is the length of the MAC address info field,  $L_{MACFTR}$  is the length of the MAC footer in bytes (2 bytes), and  $R_{data}$  is the raw data rate (250 kbps).

The transmission time for an acknowledgment can be formulated as:

$$T_{ACK} = \left( L_{phy} + L_{MACHDR} + L_{MACFTR} \right) / R_{data} \quad (4)$$

So, in summary, the delay can be expressed as:

$$de(x) = a(x) + b \quad (5)$$

In Equation (5),  $a$  and  $b$  depend on the length of the data bytes (SIFS or LIFS) and the length of the address used (64 bit, 16 bit, or no addresses). Then, the delay in the communication between a node and the edge node is a linear function of the number of bytes in the payload and the number of nodes. The MPDU is set to a maximum of 127 bytes, but the maximum number of payload bits depends on the usage of short or long addresses.

Figure 13 shows the time required by the RXN node to retransmit a message, and the time required by TXN to send a second frame using eACK and iACK,  $B_0 = 3$  and short addresses scheme.

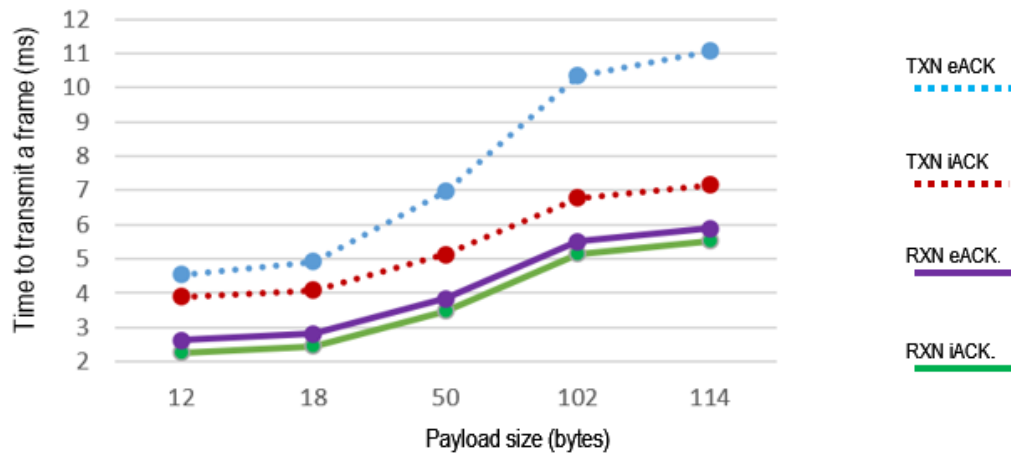


Figure 13. Delay according to the payload size.

The results indicate that in the different scenarios the delay is lower when iACK is used. With a payload of 114 bytes and eACK, the delay in RXN is 5.88 ms while with iACK it is 5.53 ms. With a 12-byte payload and eACK, the delay is 2.62 ms and with iACK it is 2.27 ms. Thus, delays increase with payload length, but they are always lower with iACK.

Figure 14 shows how the delay decreases with the use of iACK instead of eACK, in percentages that depend on the payload, the length of the node's address and the mode of operation of the node, TXN or RXN. The delay reduction by using iACK is higher in the TXN node. In RXN, the percentage is smaller when the payload decreases.

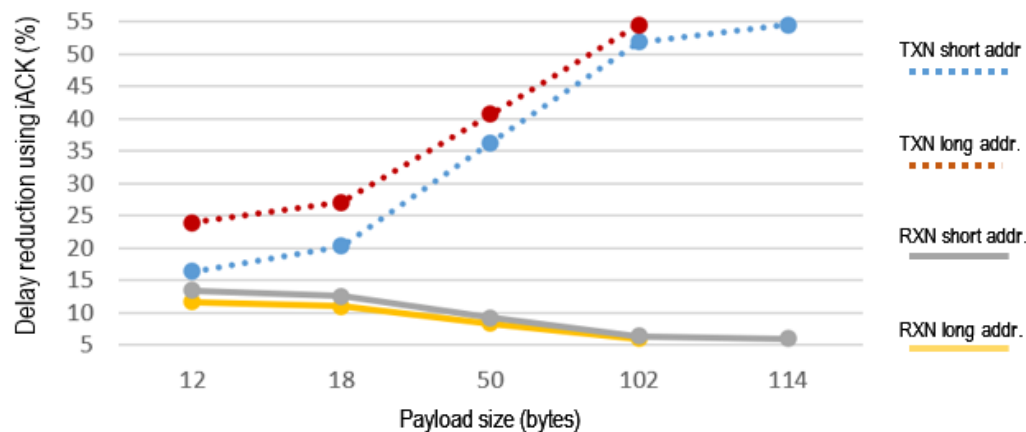


Figure 14. Delay reduction when using iACK compared to eACK.

When the TXN node sends a frame to the RXN node, the node expects to receive an eACK or an iACK. The node must wait a maximum time to receive these confirmations, before concluding that the sent frame did not reach its destination because of errors in the transmission. The channel error probability is independent of the acknowledgment method used, however when there are erroneous frames, the retransmission time of the frames varies according to the approach used. IEEE 802.15.4 defines the *MacAckWaitDuration* parameter, with a value of  $54 T_s$ , as the  $T_{weACK}$  time that the TXN node must wait to retransmit a frame when it does not receive an eACK.

$$T_{weACK} = 54 T_S \quad (6)$$

when iACK is used, the time that the TXN node must wait to start the retransmission process of the frame,  $T_{wiAck}$ , is equal to the time it takes for the RXN node to finish retransmitting the entire frame plus the  $T_{BO}$  and  $T_{CCA}$  times. This is because the TXN node must receive the frame retransmitted by the RXN node to validate the successful transmission.

$$T_{wiAck} = T_{BO} + T_{CCA} + T_{frame} \quad (7)$$

Even though the delay to retransmit the erroneous frame is greater with iACK, we must bear in mind that with iACK the retransmission of the frame in each node when there are no errors is  $34 T_S$  faster, as mentioned above. As such, the time that is lost with iACK when the frame is retransmitted is compensated in the following hops. The number of hops required to make up the time lost due to iACK retransmission is shown in Table 2. Having into account that a typical linear infrastructure includes hundreds or thousands of nodes, there will not probably be any extra delay caused by using iACK instead of eACK.

**Table 2.** Number of hops required to balance out a retransmission with iACK compared to eACK according to the payload size.

Payload	Required Hops
12	1
18	1
50	3
102	6
114	7

In linear structures with  $n$  nodes, the time to transmit information from node  $i$  to the extreme of the network, without collisions or lost frames in the transmission, is (for  $P_l$  and  $P_r$  flows):

$$de_l(x) = (i) de(x) \quad (8)$$

$$de_r(x) = (n - i) de(x) \quad (9)$$

Since each node knows its position or address ( $i$ ) within the linear infrastructure, it will select the shortest path to reach the extreme of the network, so it will select the  $P_l$  or  $P_r$  flow that minimizes the delay. Table 3 shows the time to transmit from a node to the extreme of the network with a variable number of intermediate nodes. By using the iACK mechanism the time can be reduced by 9%.

**Table 3.** Time for end-to-end transmission with 16 bits ID.

Intermediate Nodes	LWSN Length (Km)	eACK Delay (s)	iACK Delay (s)
100	3	0.33441	0.30721
1000	30	3.3441	3.0721
2000	60	6.6882	6.1442
3000	90	10.0323	9.2163
4000	120	13.3764	12.2884

#### 4.3. Energy Consumption

WSN nodes are composed of several functional modules that should be modeled in order to estimate the energy consumption: microprocessor, transceiver, sensor, and power supply. The most relevant is the communication model that considers the energy consumption of the transmitter and receiver [47]. Most sensor nodes support the states *Idle*, *Transmit* and *Receive*.

To model the energy consumed by transceiver, the average active ratio has been used. It is defined as the average active time divided by the total time spent [48]. It is a reasonable

measure since energy consumption for transmission and reception (including idle listening and channel sensing) are similar in many transceivers compared to the energy consumption for standby or sleep mode [49].

An analysis has been performed to evaluate the energy consumption using iACK or eACK. Considering typical scenarios for linear structures where only alarms are transmitted, making the traffic generated very low. Let  $P_t$ ,  $P_r$ ,  $P_i$  be the power consumed during transmission, reception and idle states respectively, and let the duration of a transmission be the sum of the duration of the data frame, ACK frame, back off, IFS, turnaround, CCA and TA.

When using eACK, the RXN node consumes an additional amount of energy to confirm the reception of the frame.

$$AE_{eACK} = P_r T_{ACK} \quad (10)$$

The percentage of additional energy consumed by the RXN node,  $AE_{RXN}$ , using eACK to receive the frame, transmit the ACK and be ready to retransmit the frame, can be calculated as, (Equation (11)):

$$AE_{RXN} = \frac{P_r T_{ACK}}{P_t T_{fra} + P_t T_{ACK} + P_i (T_{TA} + T_{BO} + T_{CCA})} \quad (11)$$

The energy consumed in the TXN node to know that the frame sent was received can be calculated for the eACK (Equation (12)) and iACK schemes (Equation (13)):

$$E_{eACK} = P_t T_{fra}(x) + P_r T_{ACK} + P_i (T_{IFS} + T_{TA} + T_{BO} + T_{CCA}) \quad (12)$$

$$E_{iACK} = P_t T_{fra}(x) + P_i (T_{TA} + T_{BO} + T_{CCA}) \quad (13)$$

The percentage of additional energy consumed by the TXN node,  $AE_{TXN}$ , using eACK to send the next frame, can be calculated as Equation (14):

$$AE_{TXN} = \frac{P_r T_{ACK} + P_i ((T_{waite} - T_{wait}) + 2T_{TA} + T_{BO} + T_{CCA})}{P_t T_{fra} + P_r T_{ACK} + P_i (T_{waite} + 4T_{TA} + T_{BO0} + T_{BO1} + 2T_{CCA})} \quad (14)$$

The percentage of additional energy consumed by the TXN node,  $AE_{TXNf}$  using eACK to retransmit the frame that did not reach the RXN node, can be calculated as Equation (15):

$$AE_{TXNf} = \frac{P_i (T_{waite} - T_{wait})}{P_t T_{frame} + P_i (T_{waite} + 2T_{TA} + T_{BO0} + T_{CCA})} \quad (15)$$

Many research works have evaluated sensor node energy consumption while transmitting, receiving, or idle. In most cases energy consumption is similar in transmission or reception mode (processing and waiting) [50,51] and slightly larger than the idle mode consumption. In our experiments, and according to the specifications of our hardware [52], our node uses 55.8 mW, 49.9 mW, and 12.3 mW for  $P_t$ ,  $P_r$ , and  $P_i$  respectively, in non-beacon mode. In all cases, these values depend on the particular sensor node.

Figure 15 shows how energy consumption in the TXN and RXN nodes is reduced when iACK is used in comparison with eACK. Percentages depend on payload, node address length, and mode of operation. The percentage decrease in power consumption at the RXN node is better with small payload as opposed to the TXN node, where it is better with large payload.



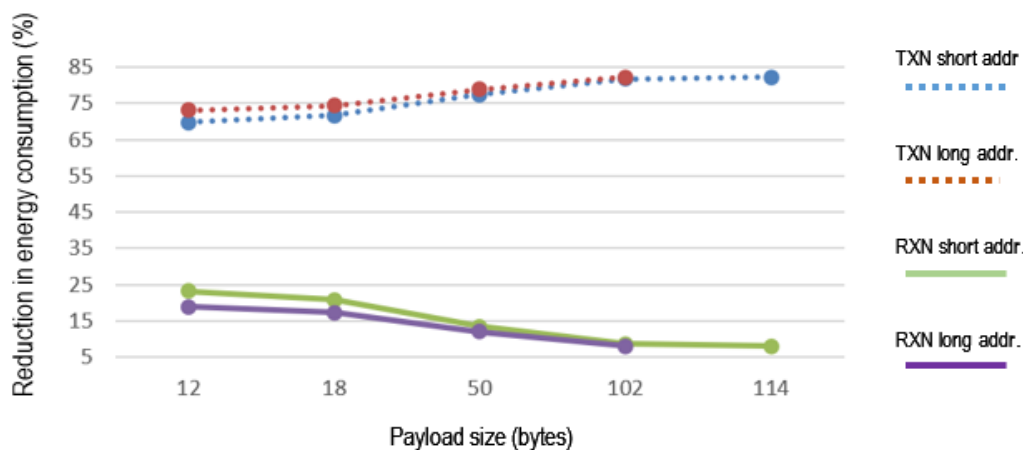


Figure 15. Reduction in energy consumption.

Table 4 shows the percentage of energy saved when using iACK compared to the use of eACK. The reduction in energy consumption in the RXN node using iACK to retransmit the frame depends on interframe spacing. With SIFS it is about 22.5%, and with LIFS is about 8%. In TXN mode, the reduction in consumption to transmit two frames using iACK depends on the payload. With a payload of 18 bytes, it is possible to save 36% of energy, and with a payload of 114 bytes about 14%. In the RXN node the use of iACK to detect the loss of frames implies an increase of energy consumption that depends on the payload. With a payload of 18 bytes the increase is about 8% of the total energy consumed by the node and with a payload of 114 bytes the increase is about 2%.

Table 4. Variation of energy consumption in the node with ID = 16 bits

Payload	$AE_{RXN}$	$AE_{TXN}$	$AE_{TXNf}$
18	22.5%	36%	−8%
114	8%	14%	−2%

Thus, by confirming the reception of the information using iACKs we can reduce the consumption of energy [28]. When the transmission reliability is left for superior layers, where the  $v_{i+1}$  node confirms the message reception to the node  $v_i$ , the energy consumption affects all the intermediate nodes that transmit the ACK confirmation.

## 5. Implementation

We have performed a practical validation of the proposed architecture using an Atmel RCB256RFR2 device [53], which includes a compatible IEEE 802.15.4 transceiver (Figure 16).



Figure 16. Atmel RCB256RFR2 node.

The MAC layer was implemented with Atmel Software Framework (ASF) [54], which simplifies the control of the contents of the payload and the header of the frame.

In our test, nodes operate as Full Function Devices (FFDs), and are arranged in a linear topology with seven nodes, as shown in Figure 17.

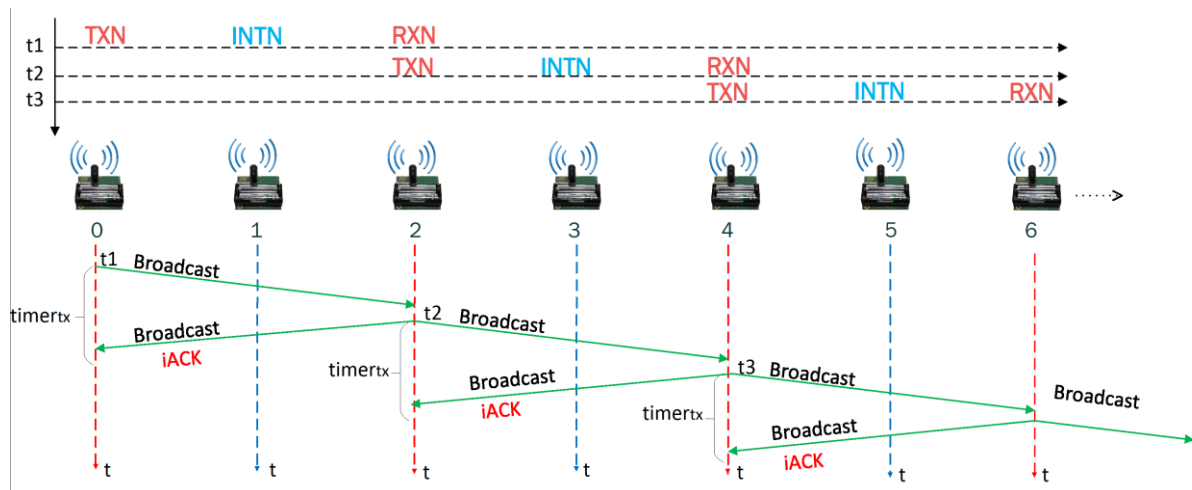


Figure 17. Tested network.

Figure 18a shows the hardware used in the experiments (Atmel RCB256RFR2 nodes). One of the environments where the measurements were completed is shown in Figure 18b). Previously, we validated the algorithm in the laboratory with three nodes working as TXN, RXN, and INTN, in a controlled environment, where sniffers were used to analyze the traffic among the nodes. Then, we used 7 nodes to build a larger structure. Although the expected communication range with IEEE 802.15.4 would be approximately 70 m, we placed each node 50 m (line of sight) from each other to ensure connectivity. The obtained measurements can be used to estimate the results in an infrastructure of hundreds of nodes, like the required for a scenario such as the shown in Figure 18c).

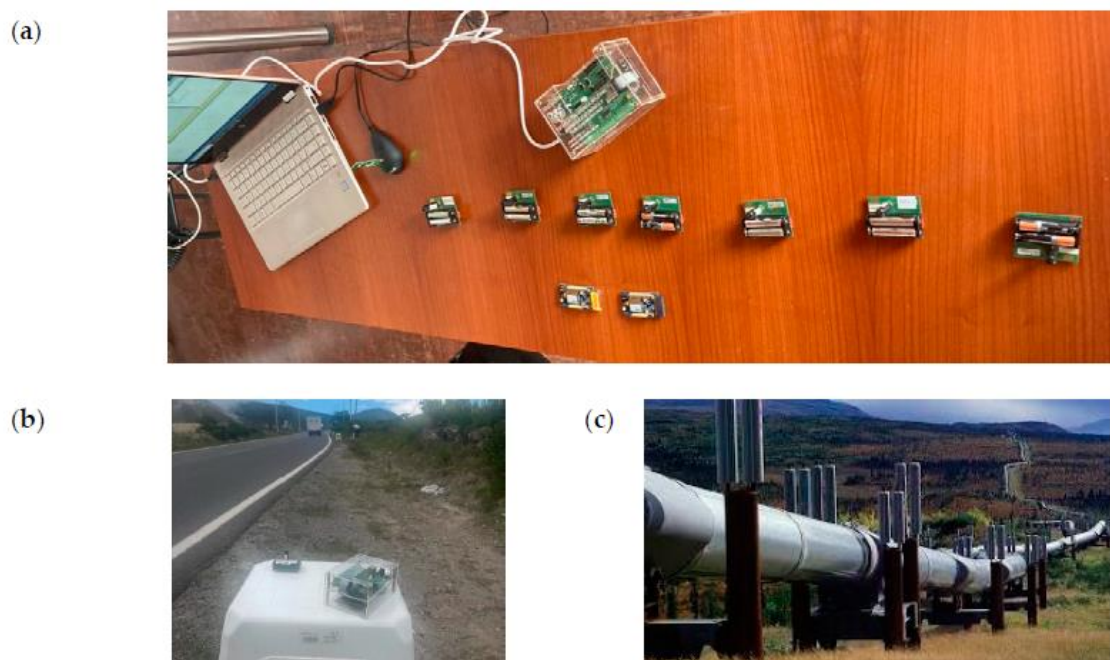


Figure 18. (a) Used hardware. (b) Scenario for the tests. (c) Typical pipeline implementation.

Table 5 shows the power levels and duration of the frames used in different the tests. A sniffer was used to capture IEEE 802.15.4 frames in order to check the correct operation of the algorithm to transmit information to the control center, even when there are retransmissions or interruptions due to damaged nodes, noisy links, or links that work intermittently. Such scenarios are forced by modifying the code of some nodes.

Table 5. Design parameters.

Parameter	Description
Transmission power	0.5 dBm
Distance between nodes	25 m
802.15.4 broadcast frame TXN	0.384 ms, 12 bytes

Failures in the link are simulated by forcing the nodes to not transmit the information when they should (according to the algorithm), and some nodes are disconnected to simulate a node that failed completely.

Different tests were completed to check the correct operation of the algorithm in those scenarios. For example, Figure 19 shows a test where the connectivity recovery is performed when a node is damaged. In this scenario, node  $v_0$  transmits data to node  $v_2$ , using a broadcast frame, node  $v_2$  retransmits it to node  $v_4$ , since node  $v_4$  is broken, node  $v_2$  does not receive an iACK and therefore retransmits the frame three times.

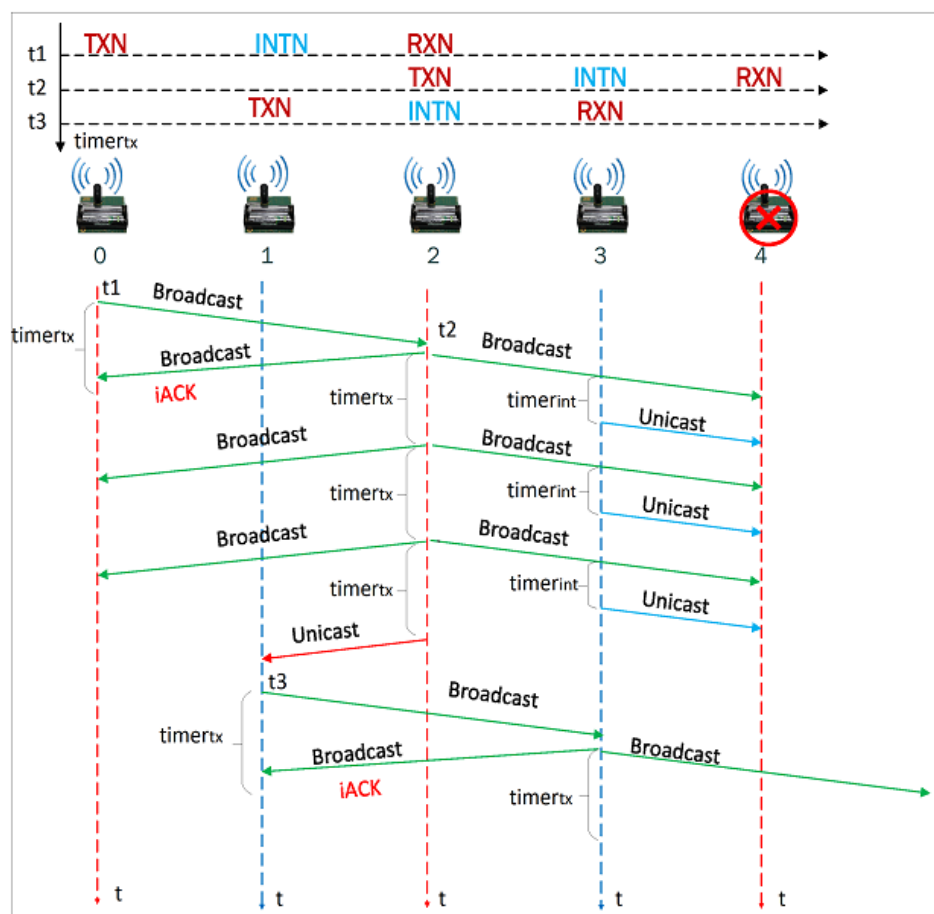


Figure 19. Test with a failing node.

Node  $v_0$  received the iACK from node  $v_2$  so it does not retransmit the information. After the third attempt, node  $v_2$  transmits a unicast frame to node  $v_1$  indicating that node

$v_4$  is damaged. Therefore, node  $v_1$ , which was operating as an INT node, will operate as a TXN node and retransmit the frame to node  $v_3$  that, in turn, transmits to node  $v_5$ , solving the problem of the failed node  $v_4$ . In case the node  $v_3$  is also broken, node  $v_1$  will discover that nodes  $v_3$  and  $v_4$  are failing and it will change the direction of the flow of the transmission.

In our approach, the delay for a frame in the node, with a payload of 12 bytes is 2.43 ms (without considering the frame processing time), the average time in the RXN node to retransmit a frame, operating with iACK is 3.4 ms. This value is similar to the average retransmission time in a TelosB node running TinyOS, which is 3.63 ms [55].

To measure the frame processing time, we used a RCB256RFR2 node, which includes an IEEE 802.15.4 transceiver. The MAC layer was implemented using Atmel Software Framework (ASF), which provides direct access to the payload and the header of the frame, without requiring an operating system in the node. This feature allowed us to measure delays using iACK and eACK with IDnode = IDPAN = 2 bytes, BO = 3 and with a frame processing time equal to LIFS 0.640 microseconds. Tables 6 and 7 show how there is also a small reduction in the processing time when using iACK both in the node working as TXN or the node working as RXN. This also helps saving energy. The difference in the frame processing time is related to the low-level operation of the software stack of the node.

**Table 6.** Frame processing time (RXN node).

Processing Time at Node RXN			
	Measured [ms]	Calculated [ms]	% error
eACK	3.78	3.32	12%
iACK	3.43	3.27	4.6%
Difference	0.35	0.05	

**Table 7.** Frame processing time (TXN node).

Processing Time at Node TXN			
	Measured [ms]	Calculated [ms]	% error
eACK	6.98	6.33	9.31%
iACK	6.66	5.82	12%
Difference	0.32	0.51	

Other protocols, such as AODV, HTR [56] COLBA, ABORT [57], RPL, or 6LowPAN [58] require higher times. For example, we can compare our proposal with 6LowPAN, a protocol that also provides address autoconfiguration, is robust against link failure or device unavailability, and that is designed for large multi-hop networks. Table 8 shows the end-to-end delay measured in [58] using 6LowPAN and using our proposal. 6LowPAN is a protocol designed for multi-hop networks, but it is not specifically adapted to large-scale multi-hop linear topologies, causing high delays because of retransmissions in higher transport layers.

It is also possible to place nodes closer than the maximum communication range to reduce the transmission power and reduce the probability to require retransmissions. For example, nodes can be placed in each one of the segments that are used to build the pipeline. Typically, those segments have a maximum length of 15 m.

**Table 8.** Comparison of the end-to-end delay between 6LowPAN and our proposal.

Hop	End to End Delay		
	Our proposal	6LowPAN	LWSN length
1	3.43 ms	13.88 ms	50 m
10	34.3 ms	138.8 ms	275 m
100	343 ms	1.38 s	2.525 km
1000	3.43 s	13.88 s	25 km
2800	9.06 s	38.86 s	70 km

## 6. Conclusions

This paper presents a new proposal for sensing linear infrastructures with WSN to send alerts. By taking advantage of the physical topology imposed by the infrastructure, we use implicit ACKs (iACKs) to reduce energy consumption and the time to transmit information to the edge of the linear segment (connected to the control center of the infrastructure). The routing also takes advantage of the particularities of the linear topology. We described the operation of the algorithm in a typical scenario and in the presence of problems such as noisy links or with inoperative nodes, showing how the algorithm can solve the problem to forward the information to the edge of the network.

After studying different existing proposals, we observe that our algorithm covers almost all the functions typically related to the network layer. Therefore, by using our algorithm at the link level layer allows for eliminating the network layer in a linear topology, leaving the few functions that are not performed, such as flow control, to the transport or application layer.

We have modeled the time required to transmit alerts from a node to the control center in order to check if this is a suitable algorithm for large infrastructures. The results show that our algorithm requires approximately 1 s to cross a 10 Km section, and that by using iACKs we can reduce the time by a 9% compared to using eACKs. Furthermore, the use of iACKs also decreases the energy consumption.

Finally, we have implemented our algorithm using Atmel RCB256RFR2 nodes to test its operation in a real scenario. We have completed different tests that included noisy links and failing nodes. We could check how our proposal provides reliable transmission and low latency for linear sensor networks, supporting failing nodes or noisy links, while simplifying the network layer. Thus, this algorithm is especially interesting for implementing pipeline monitoring applications over a 802.15.4 physical layer.

**Author Contributions:** Conceptualization, C.E.A.; Formal analysis, C.E.A.; Funding acquisition, F.G.-C.; Investigation, C.E.A., F.G.-C. and E.C.-M.; Methodology, F.G.-C., E.C.-M. and J.S.S.; Software, C.E.A.; Supervision, F.G.-C. and E.C.-M.; Validation, F.G.-C., E.C.-M. and J.S.S.; Writing—original draft, C.E.A., F.G.-C. and E.C.-M.; Writing—review & editing, C.E.A., F.G.-C., E.C.-M. and J.S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by Xunta de Galicia (GRC2018/053) and Escuela Politécnica Nacional (PIS-17-03).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The source code used for the practical validation of the proposed architecture is available at <https://github.com/cegas/routinglinklevel>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kocakulak, M.; Butun, I. An Overview of Wireless Sensor Networks towards Internet of Things. In Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 9–11 January 2017. [\[CrossRef\]](#)
2. Varshney, S.; Kumar, C.; Swaroop, A. Leach Based Hierarchical Routing Protocol for Monitoring of Over-ground Pipelines Using Linear Wireless Sensor Networks. *Procedia Comput. Sci.* **2018**, *125*, 208–214. [\[CrossRef\]](#)
3. Zhang, M.; Wang, N.; Chen, L. Sensing Technologies and Automation for Precision Agriculture. In *Women in Precision Agriculture. Women in Engineering and Science*; Khemais, T.H., Ed.; Springer Nature Switzerland AG: Cham, Switzerland, 2021; pp. 34–54. [\[CrossRef\]](#)
4. Singh, P.K.; Paprzycki, M. Introduction on Wireless Sensor Networks Issues and Challenges in Current Era. In *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's. Advances in Intelligent Systems and Computing*; Singh, P., Bhargava, B., Paprzycki, M., Kaushal, N., Hong, W.C., Eds.; Springer: Cham, Switzerland, 2020; Volume 1132. [\[CrossRef\]](#)
5. Joshi, H.; Ravindra, E. MAC Protocols for Application-Specific Wireless Sensor Networks: A Study. *Int. J. Res. Eng. Sci. Manag.* **2018**, *1*, 2581–5782.
6. Jawhar, I.; Mohamed, N.; Shuaib, K.; Kesserwan, N. An Efficient Framework and Networking Protocol for Linear Wireless Sensor Networks. *Ad Hoc Sens. Wirel. Netw.* **2009**, *7*, 3–21.
7. Raja, M.K.; Chen, X.; Lei, Y.D.; Bin, Z.; Yeung, B.C.; Yuan, X. A 18 mW Tx, 22 mW Rx Transceiver for 2.45 GHz IEEE 802.15.4 WPAN in 0.18  $\mu\text{m}$  CMOS. In Proceedings of the 2010 IEEE Asian Solid-State Circuits Conference, Beijing, China, 8–10 November 2010. [\[CrossRef\]](#)
8. Farhad, A.; Zia, Y.; Hussain, F.B. Survey of Dynamic Super-Frame Adjustment Schemes in Beacon-Enabled IEEE 802.15.4 Networks: An Application's Perspective. *Wirel. Pers. Commun.* **2016**, *91*, 119–135. [\[CrossRef\]](#)
9. Jawhar, I.; Mohamed, N.; Agrawal, D.P. Linear wireless sensor networks: Classification and applications. *J. Netw. Comput. Appl.* **2011**, *34*, 1671–1682. [\[CrossRef\]](#)
10. Sarr, M.D.; Delobel, F.; Misson, M.; Niang, I. Robust Discovery, Addressing and Routing Protocol for Dynamic Linear Networks. In Proceedings of the International Symposium on Wireless Communication Systems, Bologna, Italy, 28–31 August 2017. [\[CrossRef\]](#)
11. Varshney, S.; Kumar, C.; Swaroop, A. Linear Sensor Networks: Applications, Issues and Major Research Trends. In Proceedings of the International Conference on Computing, Communication and Automation, Greater Noida, India, 5–6 May 2015. [\[CrossRef\]](#)
12. Varshney, V.; Rajput, P.K.; Singh, A.; Varshney, G. Routing Techniques Used for Monitoring the Linear Structures Using Linear Wireless Sensor Networks: An Overview. In Proceedings of the International Conference on Computing, Communication, and Intelligent Systems, Greater Noida, India, 18–19 October 2019. [\[CrossRef\]](#)
13. Fang, C.; Liu, H.; Qian, L. LC-MAC: An Efficient MAC Protocol for the Long-Chain Wireless Sensor Networks. In Proceedings of the 3rd International Conference on Communications and Mobile Computing, CMC, Qingdao, China, 18–20 April 2011. [\[CrossRef\]](#)
14. Li, Z.; Chen, Q.; Zhu, G.; Choi, Y. A Delay-Aware MAC Protocol with Implicit RTS. In Proceedings of the 2014 International Conference on Information and Communication Technology Convergence (ICTC), Busan, Korea, 22–24 October 2014; pp. 541–544.
15. Ndoye, E.H.; Jacquet, C.; Misson, M.; Niang, I. Using a Token Approach for the MAC Layer of Linear Sensor Networks: Impact of the Node Position on the Packet Delivery. In Proceedings of the 2014 IFIP Wireless Days (WD), Rio de Janeiro, Brazil, 12–14 November 2014. [\[CrossRef\]](#)
16. Wang, J.; Ren, X.; Chen, F.J.; Chen, Y.; Xu, G. On MAC optimization for large-scale wireless sensor network. *Wirel. Netw.* **2016**, *22*, 1877–1889. [\[CrossRef\]](#)
17. Sokullu, R.; Demir, E. A comparative study of MAC protocols for linear WSNs. *Procedia Comput. Sci.* **2015**, *52*, 492–499. [\[CrossRef\]](#)
18. Vuletic, P. Application of WSN in Railway Intelligent Transportation System (RITS). In Proceedings of the 2015 23rd Telecommunications Forum, Belgrade, Serbia, 24–25 November 2015. [\[CrossRef\]](#)
19. Shokri, R.; Yazdani, N.; Yazdani, N.; Khonsari, A. Chain-Based Anonymous Routing for Wireless Ad Hoc Networks. In Proceedings of the Consumer Communications and Networking Conference, Las Vegas, NV, USA, 11–13 January 2007; pp. 297–302.
20. Sun, Z.Z.X.; He, J.; Chen, Y.; Ma, S. A New Routing Algorithm for Linear Wireless Sensor Networks. In Proceedings of the 6th International Conference on Pervasive Computing and Applications, Oulu, Finland, 11–13 May 2011; pp. 497–501.
21. Chen, X.; Xu, L.; Xu, M.; Wang, L. Research on the AODV routing algorithm in linear topological wireless sensor networks. *Int. J. Simul. Syst. Sci. Technol.* **2016**, *17*, 29.1–29.5. [\[CrossRef\]](#)
22. Malarselvi, G.; Sivajayaprakash, A. A Comprehensive Survey on Applications and Transport Layer Protocols in Wireless Sensor Networks. *Int. J. Eng. Technol.* **2018**, *5*, 146–150.
23. Paek, J.; Govindan, R. RCRT: Rate-Controlled Reliable Transport for Wireless Sensor Networks. In Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, Sydney, NSW, Australia, 6–9 November 2007; pp. 305–319. [\[CrossRef\]](#)
24. Sadouni, S.; Benslama, M.; Mebarki, A.; Beylot, A. SCTP-WSN New Extension for More Reliable Sparse Wireless Sensor Networks. In Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 2109–2114. [\[CrossRef\]](#)
25. Park, S.; Vedantham, R.; Sivakumar, R.; Akyildiz, I. GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2008**, *7*, 214–230. [\[CrossRef\]](#)

26. Sohraby, K.; Minoli, D.; Znati, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*; Wiley-Interscience: Hoboken, NJ, USA, 2007; pp. 229–244.
27. Gitman, I. Comparison of Hop-By-Hop and End-To-End Acknowledgment Schemes in Computer Communication Networks. *IEEE Trans. Commun.* **1976**, *24*, 1258–1262. [[CrossRef](#)]
28. Rosberg, Z.; Liu, R.P.; Dong, Y.A.; Tuan, L.D.; Jha, S. ARQ with Implicit and Explicit ACKs in Wireless Sensor Networks. In Proceedings of the GLOBECOM—IEEE Global Telecommunications Conference, Taipei, Taiwan, 7–11 December 2008. [[CrossRef](#)]
29. Rusli, M.E.; Harris, R.; Punchihewa, A. Performance Analysis of Implicit Acknowledgement Coordination Scheme for Opportunistic Routing in Wireless Sensor Networks. In Proceedings of the 2012 International Symposium on Telecommunication Technologies, Kuala Lumpur, Malaysia, 26–28 November 2012. [[CrossRef](#)]
30. Ruiz, M.T.; Lytras, M.D.; Mathkour, H. Innovative services and applications of wireless sensor networks: Research challenges and opportunities. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 5. [[CrossRef](#)]
31. Rahbari, D.; Nickray, M. Low-latency and energy-efficient scheduling in fog-based IoT applications. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1406–1427. [[CrossRef](#)]
32. Dwivedi, R.K.; Kumar, R. Sensor Cloud: Integrating Wireless Sensor Networks with Cloud Computing. In Proceedings of the 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, Gorakhpur, India, 2–4 November 2018. [[CrossRef](#)]
33. Kim, B.S.; Il Kim, K.; Shah, B.; Chow, F.; Kim, K.H. Wireless sensor networks for big data systems. *Sensors* **2019**, *19*, 1565. [[CrossRef](#)] [[PubMed](#)]
34. Ali, S. SimpliMote: A Wireless Sensor Network Monitoring Platform for Oil and Gas Pipelines. *IEEE Syst. J.* **2018**, *12*, 778–789. [[CrossRef](#)]
35. Zimmerling, M.; Dargie, W.; Reason, J.M. Energy-Efficient Routing in Linear Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, Washington, DC, USA, 7 November 2007. [[CrossRef](#)]
36. Hammoudeh, M. A wireless sensor network border monitoring system: Deployment issues and routing protocols. *IEEE Sens. J.* **2017**, *17*, 2572–2582. [[CrossRef](#)]
37. Elnaggar, O.E.; Ramadan, R.A.; Fayek, M.B. WSN in monitoring oil pipelines using ACO and GA. *Procedia Comput. Sci.* **2015**, *52*, 1198–1205. [[CrossRef](#)]
38. Tong, F.; Ni, M.; Shu, L.; Pan, J. A Pipelined-Forwarding, Routing Integrated and Effectively-Identifying MAC for Large-Scale WSN. In Proceedings of the GLOBECOM, Atlanta, GA, USA, 9–13 November 2013; pp. 225–230. [[CrossRef](#)]
39. Villordo, I.; Torres, N.; Carvalho, M.; Menchaca, R.; Rivero, M.; Menchaca, R. A Selective-Awakening MAC Protocol for Energy-Efficient Data Forwarding in Linear Sensor Networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6351623. [[CrossRef](#)]
40. Khanh, H.; Ock, C.; Kim, M. RP-MAC A Cross-Layer Duty Cycle MAC Protocol with a Reduced Pipelined-Forwarding Feature for Wireless Sensor Networks. In Proceedings of the 11th International Wireless Communications and Mobile Computing Conference, IWCMC 2015, Dubrovnik, Croatia, 24–28 August 2015; pp. 1469–1474. [[CrossRef](#)]
41. Khaliq, S.; Henna, S. HE-PRMAC Hop Extended Pipelined Routing Enhanced MAC Protocol for Wireless Sensor Networks. In Proceedings of the Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, Ireland, 24–26 August 2016; pp. 392–397. [[CrossRef](#)]
42. Malick, E.; Diallo, O.; Hakem, N.; Jacquet, F.; Misson, M. Interference-Aware clustering approach improving QoS for linear WSNs using token based MAC protocol. *Int. J. Commun. Syst.* **2020**, *33*, 2581–5782. [[CrossRef](#)]
43. Lee, S.; Su, Y.W.; Shen, C.C. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In Proceedings of the Industrial Electronics Conference, Vigo, Spain, 4–7 June 2007. [[CrossRef](#)]
44. Egas Acosta, C.; Gil-Castañeira, F.; Costa-Montenegro, E.; Silva, J.S. Automatic Allocation of Identifiers in Linear Wireless Sensor Networks Using Link-Level Processes. In Proceedings of the 2016 8th IEEE Latin-American Conference on Communications, Medellin, Columbia, 15–17 November 2016. [[CrossRef](#)]
45. Wijetunge, S. Performance Analysis of IEEE 802.15.4 Based Wireless Sensor Networks. Ph.D. Thesis, University of Western Sydney, Sydney, Australia, 2013.
46. Zhu, J.; Tao, Z.; Lv, C. Performance improvement for IEEE 802.15.4 CSMA/CA scheme in large-scale wireless multi-hop sensor networks. *IET Wirel. Sens. Syst.* **2013**, *3*, 93–103. [[CrossRef](#)]
47. Zhou, H.-Y.; Luo, D.-Y.; Gao, Y.; Zuo, D.C. Modeling of Node Energy Consumption for Wireless Sensor Networks. *Wirel. Sens. Netw.* **2011**, *3*, 18–23. [[CrossRef](#)]
48. Rasheed, M.B.; Javaid, N.; Imran, M.; Khan, Z.A.; Qasim, U.; Vasilakos, A. Delay and energy consumption analysis of priority guaranteed MAC protocol for wireless body area networks. *Wirel. Netw.* **2017**, *23*, 1249–1266. [[CrossRef](#)]
49. Bougard, B.; Catthoor, F.; Daly, D.C.; Chandrakasan, A.; Dehaene, W. Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives. In *Design, Automation, and Test in Europe: The Most Influential Papers of 10 Years Date*; Lauwereins, R., Madsen, J., Eds.; Springer Science Business Media B.V.: Dordrecht, The Netherlands, 2008; pp. 221–234, ISBN 978-1-4020-6487-6. [[CrossRef](#)]
50. Yan, J.; Zhou, M.; Ding, Z. Recent Advances in Energy-Efficient Routing Protocols for Wireless Sensor Networks: A Review. *IEEE Access* **2016**, *4*, 5673–5686. [[CrossRef](#)]

51. Fitriawan, H.; Susanto, M.; Arifin, A.S.; Mause, D.; Trisanto, A. ZigBee Based Wireless Sensor Networks and Performance Analysis in Various Environments. In Proceedings of the 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering, Bali, Indonesia, 24–27 July 2017. [[CrossRef](#)]
52. Atmel Corporation. 8-Bit AVR Microcontroller with Low Power 2.4GHz Transceiver for ZigBee and IEEE 802.15.4. 2014. Available online: [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8266-MCU\\_Wireless-ATmega128RFA1\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8266-MCU_Wireless-ATmega128RFA1_Datasheet.pdf) (accessed on 30 January 2021).
53. Atmel Corporation. Atmel AVR10004: RCB256RFR2—Hardware User Manual. 2013. Available online: [http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42081-RCB256RFR2-Hardware-User-Manual\\_Application-Note\\_AVR10004.pdf](http://ww1.microchip.com/downloads/en/AppNotes/Atmel-42081-RCB256RFR2-Hardware-User-Manual_Application-Note_AVR10004.pdf) (accessed on 30 January 2021).
54. Atmel Corporation. Atmel AVR4029: Atmel Software Framework User Guide. 2013. Available online: [http://ww1.microchip.com/downloads/en/appnotes/atmel-8431-8-and32-bit-microcontrollers-avr4029-atmel-software-framework-user-guide\\_application-note.pdf](http://ww1.microchip.com/downloads/en/appnotes/atmel-8431-8-and32-bit-microcontrollers-avr4029-atmel-software-framework-user-guide_application-note.pdf) (accessed on 30 January 2021).
55. Hao, C. *End-to-End Delay Analysis and Measurements in Wireless Sensor Networks*; Mid Sweden University: Sundsvall, Sweden, 2012.
56. Nefzi, B.; Song, Y.Q. Performance analysis and improvement of ZigBee routing protocol. *IFAC Proc. Vol.* **2007**, *40*, 199–206. [[CrossRef](#)]
57. Tall, H.; Chalhoub, G. ABORT: Acknowledgement-based opportunistic routing protocol for high data rate multichannel WSNs. *J. Sens. Actuator Netw.* **2017**, *6*, 23. [[CrossRef](#)]
58. Calveras, A.; Ludovici, A. Implementation and Evaluation of Multi-Hop Routing in 6LoWPAN. In Proceedings of the IX Jornadas de Ingeniería Telemática (JITEL 2010), Valladolid, Spain, 29 September–1 October 2010.