

# Publish/Subscribe Protocol in Wireless Sensor Networks: Improved Reliability and Timeliness

Ernesto Garcia Davis<sup>1,2</sup>, and Anna Calveras Augé<sup>2</sup>

<sup>1</sup> Faculty of Computer System Engineering, Technological University of Panama  
Campus Victor Levi Sasso, Panama - Panama  
[e-mail: ernesto.garcia@entel.upc.edu]

<sup>2</sup> Wireless Network Group, Universitat Politècnica de Catalunya  
C/Jordi Girona, 1-3 Mòdul C-3 08043 Barcelona - Spain  
[e-mail: anna.calveras@entel.upc.edu]

\*Corresponding author: Ernesto Garcia Davis

---

## Abstract

The rapidly-evolving demand of applications using wireless sensor networks in several areas such as building and industrial automation or smart cities, among other, makes it necessary to determine and provide QoS support mechanisms which can satisfy the requirements of applications. In this paper we propose a mechanism that establishes different QoS levels, based on Publish/Subscribe model for wireless networks to meet application requirements, to provide reliable delivery of packet and timeliness. The first level delivers packets in a best effort way. The second one intends to provide reliable packet delivery with a novel approach for Retransmission Timeout (RTO) calculation, which adjusts the RTO depending on the subscriber Packet Delivery Ratio (PDR). The third one provides the same reliable packet delivery as the second one, but in addition, it provides data aggregation trying to be efficient in terms of energy consumption and the use of network bandwidth. The last one provides timeliness in the packet delivery. We evaluate each QoS Level with several performance metrics such as PDR, Message Delivery Ratio, Duplicated and Retransmitted Packet Ratio and Packet Timeliness Ratio to demonstrate that our proposal provides significant improvements based on the increase of the PDR obtained.

---

**Keywords:** wireless sensor networks, publish/subscribe protocols, packet delivery, timeliness, data aggregation.

## 1. Introduction

---

This work is partially supported by the Professional Excellence Program SENACYT-IFARHU of the Government of Panama and by the ERDF and the Spanish Government through project TEC2016-79988-P, AEI/FEDER, UE.

Wireless Sensor Networks (WSNs) [1] are a collection of different types of nodes with the capability of sensing and communicating with other sensors and platforms. These elements are able to detect events and notify them through wireless links. The widespread deployment of WSN based applications that can be used in environmental monitoring, traffic and building management, battlefield surveillance, and home automation, have demonstrated that WSNs are one of the most promising technologies to obtain information from the environment and provide event service. In this sense, Publish/Subscribe communication model [2, 3] has been adopted, in most cases, for the design of event services in WSNs to deploy many of the applications in the aforementioned areas. Fig.1 shows the basic elements of a general Publish/Subscribe architecture. It consists on the subscribers who have the ability to express their interest in events produced by publishers [4]. A subscription is the process of registering interest in an event. Publishers are the elements that generate certain information. In addition, a node called broker, manages subscriptions, stores them, and sends publications to the subscriber nodes. Also, this broker node could be used as an interconnection component among other networks; meaning that it could serve as a Gateway for external networks such as Internet. However, in this article we focus on the events caused by the Publish/Subscribe communication model inside the WSN.

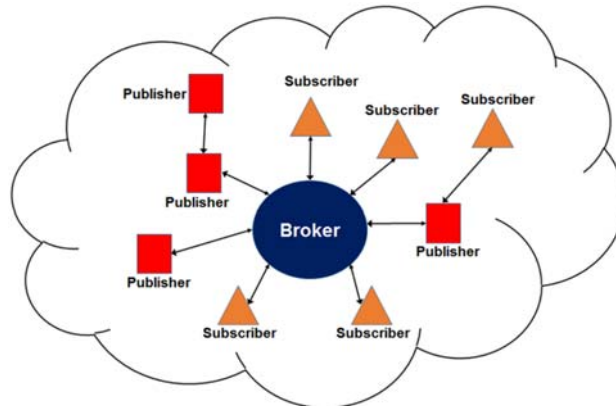


Fig. 1. General Publish/Subscribe Architecture

Decoupling in time is the strength of this model so that publishers and subscribers do not need to be actively participating simultaneously in the event interaction. Furthermore, as publishers and subscribers do not need to know about each other, we can assume that the model is also decoupled in space. Another relevant event interaction is its asynchronous manner of working. In other words, an event can be asynchronously notified to a subscriber while it is simultaneously performing some activity; meanwhile, publishers are not blocked while producing events. This allows greater scalability and flexibility and provides a more dynamic network topology, making it appropriate for applications based on WSN. However, WSNs face several barriers or challenges to be widely adopted for the industry. One of the most important challenges is the lack of QoS support to meet applications requirements, mainly in the reliable data delivery, and the timeliness of the data delivery.

Today, the paradigm of Smart City [5, 6] is an application scenario for the Publish/Subscribe model. The proper operation of critical infrastructures such as gas, water, and electric systems play an important role in ensuring the welfare of the population. Providing a smart way to monitor and maintain the functioning and operation of the components of these critical infrastructures could save on maintenance and repair costs. Moreover, it would provide a continuous service for citizens. We

consider that WSN technologies and the use of Publish/Subscribe model can be a solution for this purpose.

For example, consider a scenario where wireless sensors are deployed to monitor electrical substation environmental parameters as temperature, humidity or physical security. The sensors are attached to different places such as power transformer, power lines and so on.

For this scenario, there are different types of messages such as normal operation message, alarm message and critical status message. The application needs to meet different QoS requirements for each of these messages. For example, a normal operation message should be delivered without reliability since it does not require performing an action, this could be the case of normal temperature values from the power transformer. However, if the temperature of substation is too high, the power equipment might not work normally. In this situation, alarm message should be transmitted to control the air-conditioning, to adjust automatically according to the temperature. Thus, it is crucial that sensor networks provide a reliable delivery of every alarm message back to the application to register abnormal conditions and to perform a proper action. Several works have proposed reliability mechanisms to satisfy this requirement. Most of them propose a fixed retransmission scheme [7-15]; however, they do not consider the network conditions that are crucial to calculate an appropriated RTO value.

Another use case is considering the example, of electrical substations with several movement sensors detecting unauthorized intrusions. The sensed data has a deadline, that is, it needs to be delivered within a time-limited bound to react properly to risky situations in real-time. This would be a use case for critical messages. In this sense, there are few proposals that focus on this issue where they satisfy this requirement only through priority mechanisms [8,10,11,16,17] in which a packet with a defined deadline is sent first than the rest of the others. These proposals do not consider whether the deadline of packets has expired before reaching destination.

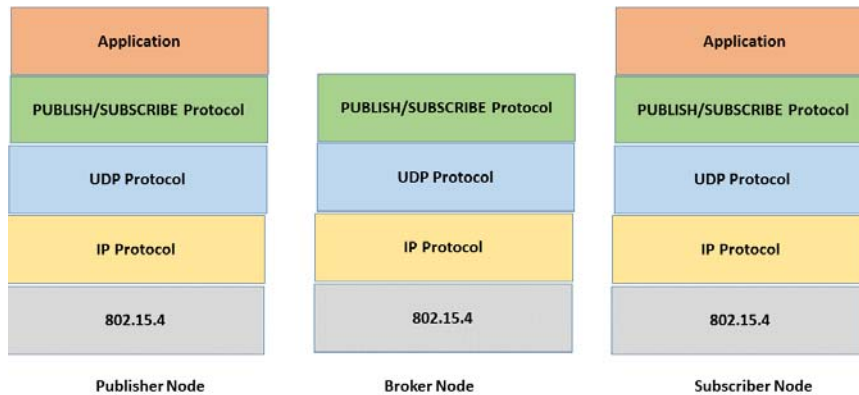
In the above realistic scenario, the Publish/Subscribe model is a feasible communication model since there are a large number of nodes that publish data from its different environmental sensors (publisher nodes) and a small number of nodes that are interested to receive and consume this data (subscriber nodes). Moreover, one of the advanced features of this model is to support QoS in terms of reliability and timeliness. QoS in WSNs can be characterized by reliability, timeliness, robustness, availability, and security, among others depending on the type of target application [18].

Considering the above mentioned application scenario and the presented QoS requirements application as a motivation, we propose in this article a mechanism that establishes different QoS levels to provide reliable packet delivery and timeliness based on Publish/Subscribe model for wireless sensor networks to meet application requirements. The first level (QoS0) is oriented to deliver packets in a best effort way. The second one (QoS1) intends to provide reliable packet delivery with a novel approach for Retransmission Timeout (RTO) calculation. The third one (QoS2) provides the same reliable packet delivery as the second one, but in addition provides data aggregation [19-21] with the aim to be efficient in terms of energy consumption and the use of network bandwidth. The last one (QoS3) provides the same reliable packet delivery packet as the second one but providing mainly a timeliness in the packet delivery.

The structure of the rest of the paper is as follows. Section 2 defines the proposed mechanism to provide QoS levels in an adaptive manner and presents a comparison with other proposals. In Section 4, we discuss and analyse the performance and evaluation tests. Finally, Section 5 presents the conclusions.

## 2. Publish Subscribe and Quality of Service

Before detailing the different levels of quality of service proposed in this work, it is important to present the general operation of the Publish/Subscribe protocol. Fig. 2, shows that the Publish/Subscribe service mechanism is located between the Application layer and the rest of the other layers in the protocol stack for all nodes: publisher, broker and subscriber. On the Publisher node, the Application layer is responsible of generating event packets that are transmitted to the Broker node, where Publish/Subscribe protocol is in charge of verifying subscriber lists and performing QoS mechanism to satisfy subscriber nodes requirements. Finally, Application layer on Subscriber nodes receives event packets.



**Fig. 2.** Publish/Subscribe protocol stack

The transport protocol used is UDP since there are three fundamental problems with TCP over constrained-node networks, such as WSN. First, TCP has an increased header size compared with that of UDP (this issue is greater in 6LoWPAN networks, where header compression reduces UDP header even further, but no header compression has been defined for TCP). Secondly, TCP is not compatible with multicast, while many use cases in constrained-node networks involve multicast-type traffic (e.g. a user turning on a group of lights in a building). Finally, a TCP receiver provides feedback to a TCP sender regardless of the application on top. However, some applications in constrained-node networks may not require end-to-end reliability, but might benefit from saving energy and bandwidth if reliability is not used. It should also be noted that TCP does not allow flexibility on the type of service provided.

## 2.1 QoS Levels Proposal

In this section, we describe each one of the QoS levels we proposed in order to meet the requirements regarding to reliability and timeliness of delivery packet. QoS level 2 and QoS level 3 are based on QoS level 1, therefore we only explain the common mechanisms in QoS level 1 section. We have three different QoS levels in order to compare the effects of reliable packet delivery with and without aggregation and timeliness. As stated before, aggregation has the aim to be efficient in terms of energy consumption and the use of network bandwidth, and timeliness provides timer application requirements.

### 2.1.1 QoS Level 1 (Reliable Packet Delivery)

If we take into account reliable packet delivery, some aspects must be considered. Packets losses occur due to the specific nature of the wireless links, or network congestion caused by multiple nodes attempting to transmit its data. A common approach to provide reliable delivery of packets is to use

the acknowledgments (ACK) and the lost packet retransmission. Several protocols as we mentioned above [7, 8-15, 22,23] propose different mechanisms to determine the appropriate retransmission timeout (RTO) in order to consider that a packet is lost and transmit it again. Aspects such as number of duplicate received packets, number of retransmitted packets, and number of received packets should be considered when designing a RTO mechanism because that could result in increasing resources consumption in an unnecessary manner such as: network bandwidth, energy, processing, and the decrement of the Packet Delivery Ratio (PDR).

In our work, in order to provide a reliable delivery packet, we propose a new approach to determine the RTO considering the aforementioned important aspects about RTO design in order to use efficiently the resources of the WSN. Specifically, we will use the the RTO mechanism that uses a K parameter to calculate the RTO previously presented in [4]. In this work, we proposed to use the same algorithm to compute the RTO value in the same way as RFC 6298 [23], but not using the Round Trip Time variance (RTTVAR), instead of this, we proposed the multiplication of Smoothed RTT (SRTT) by a K factor. In previous works, we studied the effect of different fixed K values in order to improve the PDR in the subscriber nodes. In the present work, we adjust the K value in a dynamic way by using a feedback information from the subscriber nodes.

### 2.1.1.1 Proposed adaptive RTO mechanism

Generally, the calculation of RTO takes RTT as a parameter to estimate the condition of the network. Once the RTO expires, it assumes that the data packet was lost, however it does not consider other causes such as the loss of the confirmation packet or unexpected delays. This absence of further information could result in an unsuitable calculation of RTO and it does not reflect other cases or situations. With additional information from the receiver such as received data packets, duplicate confirmation and the measured PDR, we can adjust the value of RTO for subsequent transmissions.

As stated in [4], then, we propose the RTO calculation by multiplying the estimated SRTT by a K parameter:

$$RTO = SRTT \times K \quad (1)$$

The SRTT calculation is the one proposed in RFC 6298 [31] and is performed as follows: the first time an RTT measurement is obtained at time j,  $SRTT_j = RTT_j$ . The following calculations are in the following way, where SRTT [31] is:

$$SRTT_{j+1} = (1 - \alpha)SRTT_j + \alpha(RTT_{(j+1)}) \quad (2)$$

where  $\alpha = 1/8$ , we take initially this value based on the RTO computation referred on RFC 6298 where it is calculating the SRTT using a EWMA (Exponential Weighted Moving Average). It gives more relevance to averaged samples of network condition than a recent network samples.

On the other hand, the RTT parameter is calculated as the time a publication packet is generated and its confirmation packet is received at the application level. The adaptation of the K parameter is necessary because this parameter should reflect the conditions of the network at a given time to estimate the value of RTO to appropriately react to lost packets or unexpected delay of packets.

Furthermore, we propose a verification window aimed to adapt the K parameter to be suited to the causes of the obtained PDR. The verification window is established whenever the RTT measurement is obtained, that is, every time a node sends a new publication packet (broker node or publisher node) and until it receives an acknowledgment packet. Each time the verification window is established, we continuously calculate the PDR.

The PDR is dynamically calculated for each verification window based on information obtained from the destination node (broker node or subscriber node) related to the number of not duplicated packets received at the destination node. In addition, the node sending publication packets (publisher or broker node) keeps the number of packets sent (retransmissions included) for each verification window.

Once the PDR is calculated, we must infer the causes of the obtained PDR. The causes that we attempt to discover are:

CASE 1: There was no loss of publication packet.

CASE 2: Loss of publication packet (PUBLISH LOST).

CASE 3: Loss of Confirmation packet (PUBACK LOST).

CASE 4: Spurious retransmissions of publication packet (SPURIOUS RTX).

Once the cause of resulting PDR has been determined, the value of K may be adjusted depending on the situation. In this case, the new value for K parameter is:

$$K = K' \pm F \quad (3)$$

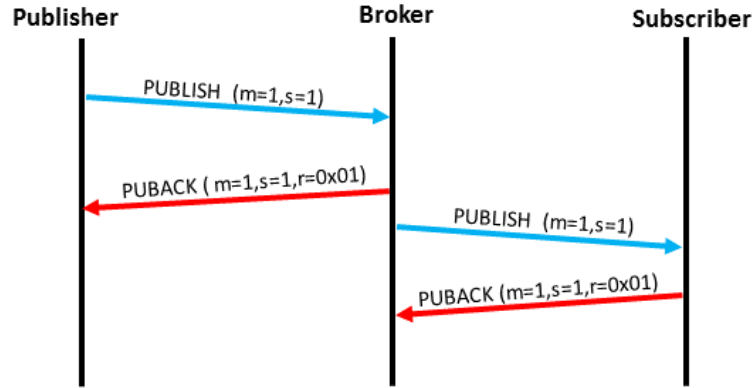
Where K' is the previous calculated value for K parameter and F, is the adjusting factor that increases or decreases the value of K according to the obtained PDR. For the adjusting factor of the K parameter, F, the used value is 0.5, for decreasing the value of K parameter (the case for lost packet) and a value of  $F = 1$  is established for increasing the value of K parameter (the case for spurious retransmissions). The reason is that when there is data loss we must respond as soon as possible to recover the packet; thus, it is important to decrease slowly the value of K parameter in order to avoid "spurious" retransmissions. However, when there is "spurious" retransmissions the reaction must be more conservative to avoid duplicated packets on the subscriber node, thus we need to increment the value of K parameter rapidly. We evaluated several alternative F values, where, 0.5 and 1 presented the best results.

To be able to ascertain the causes of the obtained PDR, the RTO mechanism uses the information from the fields of the data packet (PUBLISH) and the confirmation packet (PUBACK).

For the PUBLISH packet, it contains a field called Packet Identifier (PckId) which contains a unique identifier of that packet. In addition, we propose to add a field called "NumSeq" indicating the sequence number of the data packet being sent to the destination.

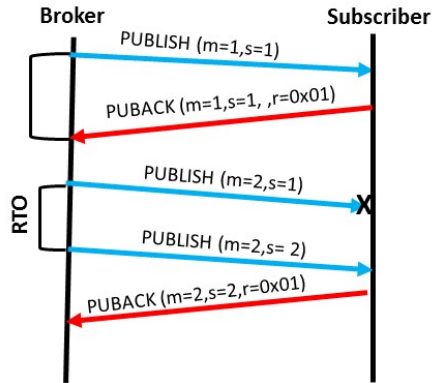
In the case of the PUBACK packet, the ReturnCode field indicates if the packet was received by the destination. We propose adding a "NumAck" field, which will contain the sequence number of the confirmation packet that is sent to the source through the packet PUBACK. Finally, we propose to create a new return code "0x02" for the "ReturnCode" field, which will indicate that it is not the first time that the confirmation packet is sent to the source. Therefore, in this situation the source could infer a PUBLISH duplicated packet caused by a confirmation packet lost.

**Fig. 3** shows the packet exchange where m indicates the Packet Identifier (PckId), s indicates the Sequence Number of PUBLISH packet and r represents the ReturnCode for PUBACK Packet. The value of "NumSeq" field of PUBLISH packet will be equal to one (1) for the first time this packet is sent to the destination. Besides the "ReturnCode" field will contain the value 0x01, which indicates that it has received and processed the corresponding PUBLISH packet by the destination.



**Fig. 3.** Packet exchange between nodes

In case there is a packet retransmission, the “PckId” field maintains the same identifier value, but the value of “NumSeq” increases sequentially, which is illustrated in Fig. 4.



**Fig. 4.** Packets retransmission

For PUBACK packet, the “PckId” field corresponds to the same value of the “PckId” field from the received PUBLISH packet. The “NumAck” field relates to the value of the “NumSeq” field from the received PUBLISH packet. In case the “ReturnCode” field value is 0x02, it indicates a PUBLISH duplicated packet as we explained above.

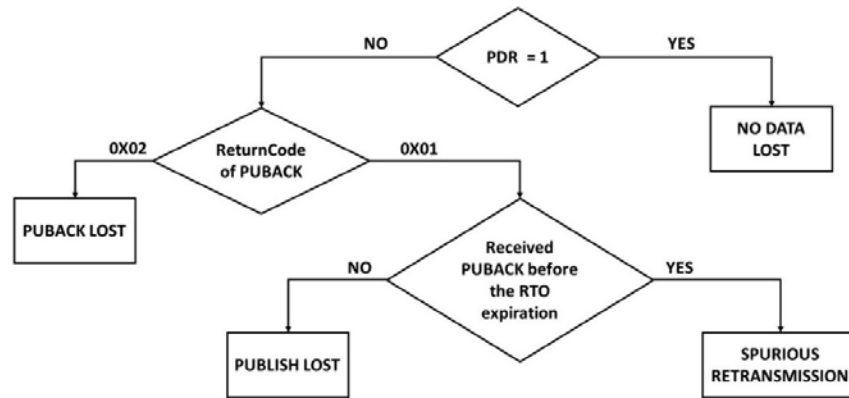
Considering the information obtained from PUBACK and PUBLISH packets, we can infer the cases such as showed in algorithm 1:

```

IF PDR == 1 THEN
    No DATA LOST (CASE 1)
ELSE
    IF (PUBACK Returncode == 0x02) then
        PUBACK LOST (CASE 3)
    ELSE
        IF (PUBACK ReturnCode == 0x01) then
            IF (Received PUBACK before the RTO expiration) then
                SPURIOUS RETRANSMISSIONS (CASE 4)
            ELSE
                PUBLISH LOST (CASE 2)

```

**Algorithm 1.** Algorithm to infer the causes of the obtained PDR.



**Fig. 5.** Flow Chart to decide the causes of the Obtained PDR

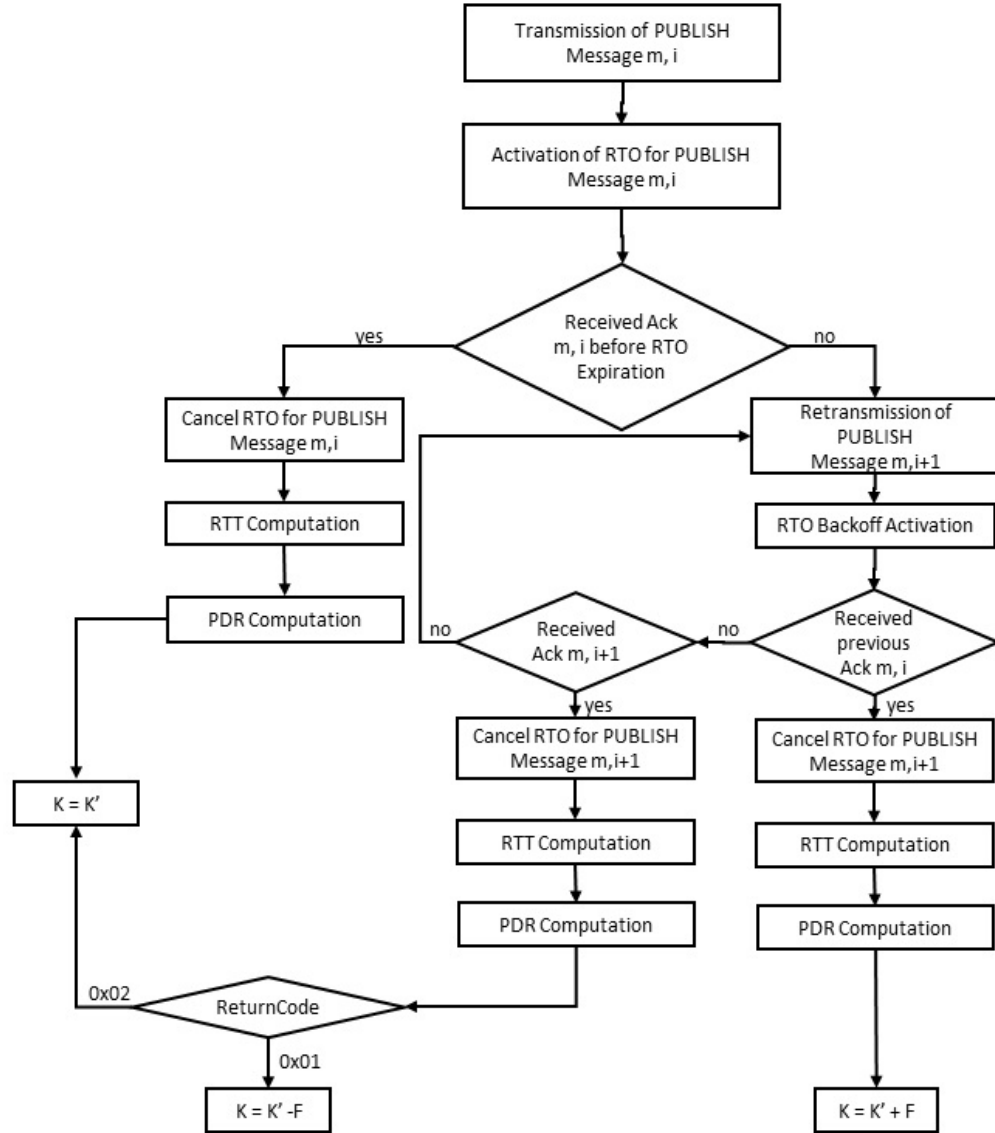
Once defined the motivation of the study, in the next section we will discuss the implementation of the adaptive algorithm.

### 2.1.1.2 Computation of K Parameter

It is important to note as we mentioned before that publisher nodes generate Publication packets and the broker node receives them. The latter is responsible for sending publication packets to the appropriate subscriber nodes. Because of this way of working, the algorithm presented for calculating the RTO runs on both nodes publishers and the broker node, including the calculation of the PDR. The proposed algorithm uses an initially conservative value for K parameter equal to 4 and then it is subsequently adjusted according to network conditions.

Based on the following criteria we choose the initial value of K: it is necessary an initial K value to avoid spurious retransmissions but not too large to avoid long time reaction to lost. Several alternative K values for small and large RTO values were evaluated, where, 4 presented the best results.

**Fig. 6** shows how the K parameter is adapted for case 1, case 2, case 3 and case 4 above mentioned. In this figure, m indicates the PckId and i represents the Sequence Number of PUBLISH packet. It is important to note in this chart that PDR computation value is equal to 1 when it receives an ACK before RTO expiration. Moreover, PDR is less than 1 when it receives an ACK after at least the first retransmission. We have represented this process in the chart with the puporse of clarifying the flow of the algorithm for the computation of the K parameter.



**Fig. 6.** Algorithm for adjusting the K parameter

In addition, the technique of deriving the value of K parameter could be summarized in equation (4).

$$K = \begin{cases} K' + F, & \text{if Received Previous Ack } m, i \text{ after a RTO} \\ K' - F, & \text{if (Received Previous Ack } m, i + 1 \text{ before a RTO) and (ReturnCode} = 0x01) \\ K', & \text{otherwise} \end{cases} \quad (4)$$

**Fig. 7** shows an example for each case to calculate the Verification Window previously detailed. In this figure, m indicates the PckId and i indicates the Sequence Number of PUBLISH packet also r represents the ReturnCode for PUBACK Packet.

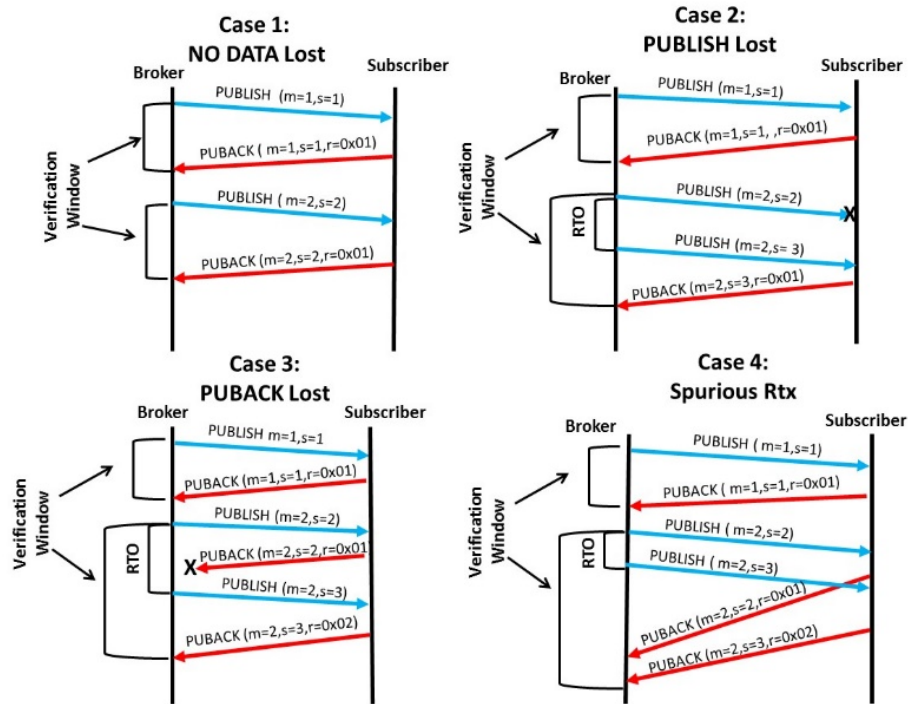


Fig. 7. Verification Window calculation for each case.

Finally, in this work, we propose a new approach to deliver a reliable packet in WSN using Publish/Subscribe model that is in accordance with some aspects of the CoAP standard protocol and CoCoA mechanism. Thus, we consider our proposal is also suitable for WSN as we demonstrate by means of the presented results and conclusions. The used publication discipline is of great importance related to the PDR performance metric evaluation. We adopt the CoAP publication discipline. It consists on giving priority to sending the new publications rather than attempting to retransmit the old ones. In this case, the PDR could be decreased because a packet lost will not be recovered when the publisher or broker node generate a new publication packet.

### 2.1.2 QoS Level 2 (Data Aggregation)

We will include to this new RTO mechanism the data aggregation capability. In most of WSN applications, sensor nodes deployed in common area, could transmit similar or same data that could result in redundant data in the network. That results on consumption of valuable resources and in network congestion. Depending on application requirements, aggregated sensor data is enough. By using data aggregation, several data values (messages) transported through the same path can be aggregated to reduce the amount of traffic and thus reducing the network congestion and helping to preserve scarce resources such as bandwidth and energy consumption. In our approach, the node playing the broker role will perform data aggregation of the next two messages being received after a specific case has occurred (publication lost, a publication acknowledgment lost or a spurious retransmission), and then transmits this in a single packet to the subscriber node. For data aggregation, we have considered two or more messages. In this paper we only show the results in which we consider data aggregation for two messages since the MDR decreases as more messages we aggregate. This is due to the CoAP publication discipline we use, as we have mentioned in section 2.1.1.2. Besides, when we have more than two aggregated messages there is a higher probability that the data get lost.

Our proposal does not consider the aggregation of messages until a successful delivery from a publisher node to broker node. Instead of that, it realizes this work only between broker node and subscriber node that requests QoS level 2. The reason for that is that other subscriber nodes have different QoS requirements so data aggregations could be not suitable for them.

This single packet contains each data value corresponding to the two previous packets. For example, the first packet contains the temperature data value 33. A second packet is received containing the data value 40. In this case, the broker node puts these values in a new packet, which is transmitted to the subscriber node. Additionally, this method provides a way to send data values that are considered important for the application. The described process is called Lossless Data Aggregation [20].

This QoS level is oriented for messages that are not critical in the “arrival on time” target to the destination node, thus there is no conflict with real time sensor readings.

It is important to note that in the presented paper we do not propose a new data aggregation method as a novelty. We use a data aggregation mechanism as a way to avoid an increase of network congestion if a node continues transmitting in a congested medium, by reducing energy consumption and the use of network bandwidth, which are other important performance metrics on WSN. The presented results will show that the simple aggregation mechanism could help to reduce congested situations and to improve performance.

### 2.1.3 QoS Level 3 (Timeliness)

As we previously explained, the full time and space decoupling and scalability of this model, makes the broker node the key point of decision to deliver the packets to subscriber nodes with different QoS levels, for example based on a deadline target previously specified by subscriber node. In contrast with other proposals [8], where each node takes the deadline decision, they do not provide the decoupling we previously mentioned. Moreover, it is very complex to manage different QoS levels. That is, each subscriber node would need to know the offered QoS level for each publisher node and publisher nodes would need to maintain a list of subscription and requested QoS level for each subscriber node. This way of working is not scalable and the resources on memory and processing resources could be high for the node.

We propose a deadline mechanism (DM) that is explained as follows. A Smooth Delay calculation for each received packet from publisher nodes to broker node (SDP2B) for instant  $j+1$  is calculated by:

$$SDP2B_{j+1} = (1 - \alpha)SDP2B_j + \alpha(DP2B_{j+1}) \quad (5)$$

where,  $DP2B_{j+1}$  is the measured delay for each packet from publisher node to broker node,  $\alpha = 1/8$ , and  $SDP2B_j$  is the SDP2B for instant  $j$ . As we mentioned above, the  $\alpha$  value is based on the RTO computation referred on RFC 6298 where it is calculating the SRTT using a EWMA (Exponential Weighted Moving Average).

In the same way showed in equation (6), we calculate a Smooth Delay for each received packet from broker to subscriber node (SDB2S) using:

$$SDB2S_{j+1} = (1 - \alpha)SDB2S_j + \alpha(DB2S_{j+1}) \quad (6)$$

where,  $DB2S_{j+1}$  is the measured delay for each packet from broker node to subscriber node,  $\alpha = 1/8$ , and  $SDB2S_j$  is the SDB2S for instant  $j$ .

Then we use the SDP2B and SDB2S values to calculate the Deadline Factor (DF), jointly with the Certainty level (CL). The CL is the probability of delivering the packet to the subscriber node based on the previous measured delay from broker to subscriber node. Then, the DF that controls the deadline mechanism is expressed as showed in equation (7).

$$DF = (SDP2B + SDB2S) \times CL \quad (7)$$

Finally, the DF value is compared with the Deadline Target (DT) that was previously

established by the subscriber node. A packet with  $DF < DT$  is transmitted to the subscriber node. Otherwise, it is discarded.

We consider that a deadline mechanism at application level that considers network conditions is vital to ensure a timeliness for delivering packet.

In conclusion, it is important to point out that this QoS level does not aim for the subscriber node to obtain a better PDR, but the packets arrive to the subscriber node as determined by the target deadline. To cope with that, the broker node selects the packets that have a higher probability to achieve the subscriber node in accordance with equation (7). In this equation, the delay is calculated as a measurement of the network condition. Therefore, those packets with a high requirement of timeliness would have higher probability to be discarded depending on the network conditions.

## 2.2 Comparison with other proposals (related work)

Several publish/subscribe approaches in WSN have been proposed in the literature. Most of them satisfy QoS reliability requirement, and others satisfy besides the timeliness requirements. In this section, we discuss and compare relevant approaches.

In [24,25], authors propose QoS in two domains: reliability and timeliness focusing on network protocol and MAC layer. For reliability domain, they consider the sensor node density in the network to route packets for multiple paths using a geographic location protocol or GPS in the node. Therefore, they can guarantee with certain probability the required end-to-end reliability level. For timeliness domain, they allow to choose the proper speed options for packets to satisfy different end-to-end deadlines. As each node maintains a delay estimation to each neighbour, it can make a decision to guarantee different deadline targets.

Nevertheless, an important disadvantage is the power overhead due to the increment of the amount of data transmission by transmitting duplicated packets. This approach could severely affect applications requiring a long network lifetime such as those based on Publish/Subscribe Model. In [26], authors propose a framework combining network and MAC layer called SchedEx-GA. In this case, the proposal guarantees the identification of a configuration to accomplish the QoS requirements. One of the drawbacks of this solution is the placement of more sink nodes to achieve the QoS required, when a configuration cannot be achieved. This solution is not compatible with publish/subscribe model for WSN. In [27] authors present a survey of several routing protocols providing QoS requirements. Most of them are considered for Real-Time Multimedia network so none of them considers the publish/subscribe architecture we are studying in our proposal. Our proposal focusses on Publish/Subscribe protocols that work on application layer to satisfy application QoS requirements. These research works mainly present mechanisms based on network and medium access layers, as well as cross-layer approaches. We consider that an approach at the application layer protocol is necessary because it provides to the application a way to express directly QoS requirements. This feature could be provided by proposals based on Publish/Subscribe in conjunction with application layer, as the work presented in this paper. In this way, the application layer could know directly the network conditions and could adapt its requirements according to the specific situation. In addition, this could lead to a network efficient application, for example by reducing a network resource consumption and achieving a better application performance. From that point, we focus and compare the research that focuses on the application layer.

The authors in [16, 17] define QoS support and a publish/subscribe middleware for real-time WSNs. These proposals evaluate a mechanism so that the dispatcher gives priority to send packets with real-time constraints, and their desired operation conditions such as delay, data rate, and energy consumption. In our proposal, we improve the timeliness by establishing a deadline mechanism in which the broker node is in charge of taking the decision on sending the packet to the subscriber node comparing the measured delay of the received packet with the specified target delay. However, it does

not provide any mechanism for reliable packet delivery. Authors in [28], propose a publish/subscribe middleware called Mires. The proposed architecture makes that each publisher advertises the data he can publish. Then, user applications, looking at the advertised services can select what it is interested to them. However, Mires does not provide any QoS support. A policy-based publish/subscribe middleware for sense and react applications is proposed in [7]. This middleware defines the delivery protocols of the packets generated by the publishers and subscribers with certain delivery guarantees: that is unreliable or reliable delivery. There are other research works [29,30] where authors only propose the design and implementation of applications in wireless sensor networks using the Publish/Subscribe model. However, none of them provides any QoS support. In PS-QUASAR [8], there is a proposal of a Publish/Subscribe middleware that implements timeliness, priority, and reliability as parameters for QoS support. The reliability is achieved through a retransmission mechanism, and the timeliness is accomplished by using a mechanism that implements deadline concept to allow discarding packets before reaching destination if their deadline has expired. The main differences between PS-QUASAR and our approach are mainly two. Firstly, our proposed RTO mechanism is adaptive based on feedback information of network conditions. Secondly, our approach provides data aggregation capability to reduce the number of packets in the network to save network bandwidth and energy resources, which is a key aspect in this type of networks. Other works, such as [9], propose a mechanism to provide reliable delivery packet through a Gateway that computes a dynamic RTO for all nodes. One of limitations of this work is that each node connects to the gateway directly, thus it is not feasible for multi-hop environments. In addition, it does not offer other QoS mechanisms to provide data aggregation and timeliness in the packet delivery. Another publish/subscribe middleware in TinyDDS [10], and sDDS [11] proposal adopts the OMG (Data Distribution Service) DDS standard [12]. Both middleware provides QoS mechanism for reliability and timeliness of the packet delivery. A QoS policy indicates the level of data transmission reliability that supports best effort and reliable packet delivery mode. In reliable mode, these middleware attempt to deliver all packets. The missed packets are retransmitted with a maximum number of retries. For best effort mode, the middleware does not retransmit the packet and relies on MAC layer until the transmission is successful or not. On the other hand, to provide timeliness, both middleware rearranges the order of packets in the packet queues. The packet-desired latency will prioritize its publication. However, both middleware is still not lightweight enough to fit the WSN requirements.

In [13] authors propose PSWSN-MM middleware. It provides reliability for packet delivery based on retransmission as the only QoS mechanism; it does not implement data aggregation and deadline. In this case, the packet is sent when previous one is acknowledged. Therefore, this proposal does not support real-time requirements.

In MQTT-S [14], subscribers can define three reliability QoS levels. First, a best effort delivery service; second, with retransmission of a notification until receiver acknowledges it (duplicated packets can be received); third, a service that ensures that a packet is not duplicated.

CoAP [15] is a standard application protocol suitable for WSN published on RFC 7252. This protocol considers most of the issues related to packet delivery in a reliable way in WSN. This protocol provides reliable packet delivery through two levels. In the first level, the packets are called NON packets, these ones are sent in best-effort mode. The second level uses CON packets, which require confirmation packets from the destination. The extension of this protocol on RFC 7641 [31] proposes that the publisher (called server) is the only one to decide the level of reliability to send the packets. There is no support for the subscriber (called observer) to negotiate or express the required level of reliability.

Both MQTT-S and CoAP protocols present different approaches to establish the RTO that have been analysed by the authors in a previous work [4]. However, we consider MQTT-S lack of additional information of the network to establish an appropriate RTO for the network conditions. On the other

hand, CoAP has proposed a new mechanism called CoCoA (CoAP Simple Congestion Control/Advanced) [30] that potentially improves congestion control mechanism, but it could result in a more complex mechanism for constrained devices in WSN. In addition, neither of these protocols provides mechanism for timeliness and data aggregation.

Our proposal and CoCoA are similar since both take the RTT as a measure of network condition to adjust the RTO value in a dynamic manner. Both techniques are called adaptive for that reason. Although both proposals use the RTT as a measure of network condition, they differ in how they use this information to calculate the subscriber's RTO. CoCoA uses two estimators in parallel to calculate the RTO called RTO weak and RTO strong. The first one is calculated when an ACK is received after the first transmission of a packet. RTO weak is computed when an ACK is received after the first retransmission of a packet. In contrast, our proposal calculates RTO value based on RTT measures and the adjusting of K parameter depending on the causes of the obtained PDR by subscriber node such as data losses, acknowledgement losses, spurious retransmissions, or successful data delivery. One of the benefits of CoCoA is the use of two estimators in parallel that provide more network information to calculate a RTO value nearer to network conditions leading to obtain a more PDR by subscriber node. However, the use of these estimators in parallel could result in more wasting of process and memory resources in the node. In contrast, our proposal, besides using the RTT also uses the PDR obtained by the subscriber, in this manner the subscriber node gets more PDR. Moreover, our proposal provides reliability with data aggregation, which improves the PDR obtained by subscriber node.

Table 1 compares the main QoS features implemented in the revised proposals as well as our proposed mechanism.

**Table 1.** Publish/Subscribe protocols QoS features summary

	<b>Reliable Delivery</b>	<b>Data Aggregation</b>	<b>Timeliness</b>
Sharifi, M. et al [16,17]	X	X	Priority Queuing
Mires [28]	X	X	X
Russello, G., et al [7]	Fixed Retransmission Scheme	X	X
TinyMQ [29]	X	X	X
Cam, H., et al [30]	X	X	X
PS-QUASAR [8]	Fixed Retransmission Scheme	X	Deadline Mechanism
Hui-Ling, C. et al [9]	Fixed Retransmission Scheme	X	X
TinyDDS [10]	Fixed Retransmission Scheme	X	Priority Queuing
sDDS [11]	Fixed Retransmission Scheme	X	Priority Queuing
PSWSN-MM [13]	Fixed Retransmission Scheme	X	X
MQTT-S [14]	Fixed Retransmission Scheme	X	X
CoAP [15]	Fixed Retransmission Scheme	X	X
CoAP+CoCoA [22]	Dynamic Retransmission Scheme	X	X
Our Proposal	Dynamic Retransmission Scheme	Lossless	Deadline Mechanism

As it can be noted, in Table 1, most of the presented protocols providing reliable delivery packet use a fixed retransmission scheme and a priority queuing to provide timeliness, but none of them provides data aggregation mechanism as a mechanism to reduce collisions and energy consumption. In contrast, our proposal provides a complete functionality of considering all these features by taking advantage of data aggregation to be efficient with the WSN traffic.

### 3. Evaluation and Results

In this section, first, we present the description of the simulation environment and the used performance metrics, to evaluate each proposed QoS Level. Finally, we analyze and discuss the obtained results from the evaluation of each QoS level by means of simulations. In each case, we use a confidence interval with a probability of 95%. We show the confidence interval in the main performance metrics presented when they are relevant.

In order to evaluate that our proposal is better than other protocols using fixed and adaptive RTO, we compare the PDR values for subscriber with QoS level 1. In this last case, we show the impact of data aggregation in order to reduce the network congestion and increase the probability of message delivery. Finally, we present the evaluation of timeliness for publication packets with the objective of delivering packets to the subscriber nodes on time.

#### 3.1 Simulation Environment

Simulation experiments were carried out using OMNet++ [32]. The channel model we used was “Free Space” model and the propagation model was “PathLossReception,” which were the default and compatible models in the OMNet++ version we used for simulations. We use a multi-hop network topology. As in [4], we consider a local WSN where the goal of the application is the monitoring and control of critical environmental parameters such as temperature and humidity in electrical substation through a WSN deployment. In this context, two types of devices are deployed in different parts of the application area: publisher and subscriber nodes. The publisher nodes are responsible for measuring the critical parameters in the electrical substation. Additionally, there are four subscriber nodes (each for a different QoS level). In addition, each publisher node sends a packet in average every 5 seconds and in total, each publisher node sends about 100 packets to the broker node and this one to the corresponding subscriber node.

We consider a multi-hop scenario consisting of nodes that are located up to 3 hops from the broker node and at equal distance among them, as shown in Fig. 8.



Fig. 8. Simulated network topology

Packets originated at the publisher node are routed to the broker node through multiple relay nodes. When these packets arrive at the broker node, the application layer of this node generates a new publication packet with the desired QoS level for the corresponding subscriber node. This new publication packet is routed in a similar way to the subscriber nodes as shown in Fig. 9 and 10. Relay nodes could be also publisher or subscriber nodes. Also in these figures, it can be noted the coverage area for each node which allows the network communication.

The simulation scenario consists of a number of publisher nodes (10, 20, 30, 40 and 50). We

fixed 4 subscriber nodes. For this work, we have considered four subscriber nodes, each one with a different QoS level. This is because we are focusing on an application scenario where the same publication message sent by the publisher nodes is required by each subscriber node with a different QoS level. However, a QoS level 1 is always considered between publisher and broker nodes to guarantee a reliable communication in this way.

For example, the first subscriber does not require reliability on packet delivery (best effort), the second one requires reliability on packet delivery, the third one requests reliability and could accept data aggregation if it is necessary, and finally, the fourth one is interested in timeliness of packets. We place Subscriber nodes so that we can study the most pessimistic scenario of each QoS level.

In this work, we use static routes among nodes to eliminate the delay caused by the implementation of routing protocol for routing decision. The implementation of a routing protocol and its effects are out of the scope of this study and would be considered for future work.

Each simulation experiment lasts 500 seconds. Each point in the graph presented in this section is the average of ten simulation runs.

We use the 2.4 GHz range with a bandwidth of 250 kbps based on IEEE 802.15.4 [33] for the PHY layer, with a maximum number of MAC-layer retransmissions set to 3, which is the default value of IEEE 802.15.4 [33]. About changing the MAC layer to a different standard it would change several aspects such as: wireless coverage, battery and node life, and throughput. No other MAC layers are considered in this study. We use the current consumption parameters based on Zolertia Z1 [34] datasheet to obtain the results for energy consumption of the nodes.

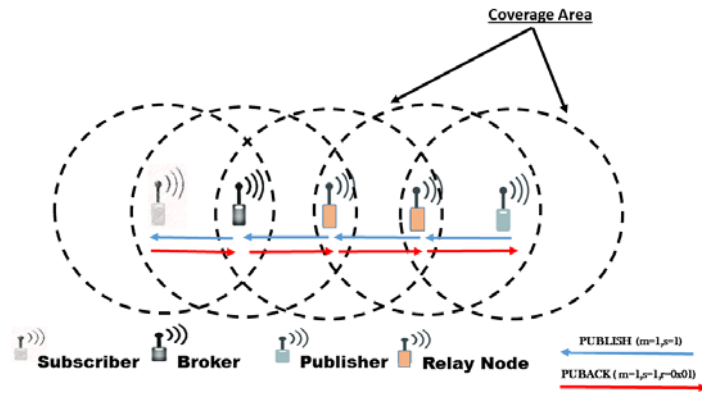


Fig. 9. Flow of packet with Subscriber node at one hop from Broker node.

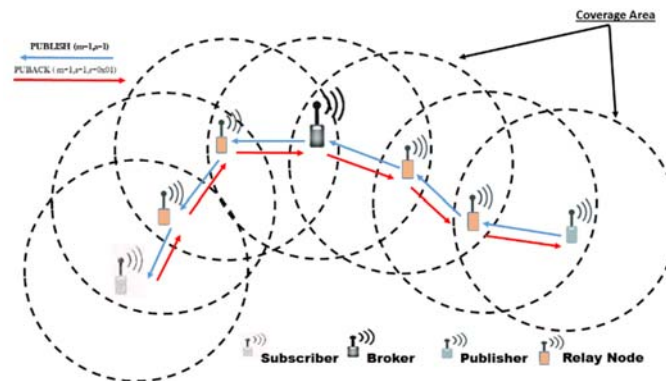


Fig. 10. Flow of packet with Subscriber node at three hop from Broker node.

### 3.2 Performance Metrics

In this section, we define the used performance metrics [4].

- **Packet Delivery Ratio (PDR):** It expresses the total number of publication packets received by each subscriber node divided by the total number of publication packets generated by all publisher nodes of the events to which the subscriber node has been subscribed. It does not consider duplicated publication packets received by subscriber nodes.
- **Message Delivery Ratio (MDR):** It expresses the total number of publication messages received by each subscriber node divided by the total number of publication messages generated by all publisher nodes of the events to which the subscriber node has been subscribed. It does not consider duplicated publication messages received by subscriber nodes.
- **Packet Timeliness Ratio (PTR):** it expresses the number of delivered packets before deadline divided by the total number of publication packets generated by all publisher nodes of the events to which the subscriber node has been subscribed. It does not consider duplicated publication packets received by subscriber nodes.
- **RTX Packet Ratio (RPTXR):** It is the ratio of the total number of publication packets retransmitted, divided by the total number of sent publications packets. This metric is evaluated for the total number of publisher nodes.
- **RTX Message Ratio (RMTXR):** It is the ratio of the total number of publication messages retransmitted divided by the total number of sent publications messages. This metric is evaluated for the total number of publisher nodes.
- **Duplication Packet Ratio (DPR):** This indicates the ratio of the number of duplicated publication packets received to the total number of publication packets received. We evaluate this metric in the subscriber node.
- **Duplication Message Ratio (DMR):** This indicates the ratio of the number of duplicated publication messages received to the total number of publication messages received. We evaluate this metric in the broker node.
- **Energy consumption:** this metrics refers to the total amount of consumed energy (J) by the broker node in order to transmit the total number of publication packet for each subscriber node with the corresponding QoS level. We evaluate this metric in the broker node.

### 3.3 Packet Delivery Ratio (PDR) QoS Level 1

As we mentioned above in Table 1, the most of proposal providing reliability are based on Fixed Retransmission Scheme and in general, they are very similar with MQTT-S and CoAP. Thus, we have chosen these protocols to compare it with our proposed dynamic method for RTO calculation. On the other hand, CoCoA proposes a mechanism to adapt RTO using network conditions in a similar manner as our proposal; therefore, we also compare it with our proposal.

We observe that, the highest PDR for the subscriber node with QoS level 1 is obtained when using our adaptive RTO method as we have proposed in this paper. As illustrated in Fig. 11 subscriber node increases the PDR up to 43% compared with the MQTT-S protocol and up to 26% compared with the CoAP protocol. On the other hand, in most cases, the highest PDR for the same subscriber is obtained when using K Parameter compared with CoCoA. In this situation, the subscriber node increases the PDR up to 3% compared with CoCoA.

One of the reasons for that is that our proposal adjusts a K parameter depending on the calculated PDR on each verification window for the subscriber node. That is, our proposal uses sequence numbers and acknowledgment numbers that allow infer a data loss, an acknowledgment loss, or a spurious retransmission and act in consequence. In contrast, MQTT-S [27] defines a fixed RTO

value of 10 to 15 seconds, and CoAP protocol chooses a RTO value from an RTO interval. Both of them do not consider any network conditions to adjust the RTO. This situation results in a low capacity to reaction to data, acknowledgment loss or a spurious retransmission and thus the subscriber node receives a lower PDR. Furthermore, CoCoA considers only RTT as a network conditions parameters. Although this could be a good criteria, we consider it is not enough to adjust RTO.

On the other hand, an important aspect to consider in the PDR is the effect caused by the publication discipline. We adopt the CoAP publication discipline. It consists of giving priority to sending the new publications rather than attempting to retransmit the old ones. Considering this situation, it can be noticed that using a fixed value for parameter K could result in greater RTO value which results in retransmission activated too late to recover publication packets in the case of loss.

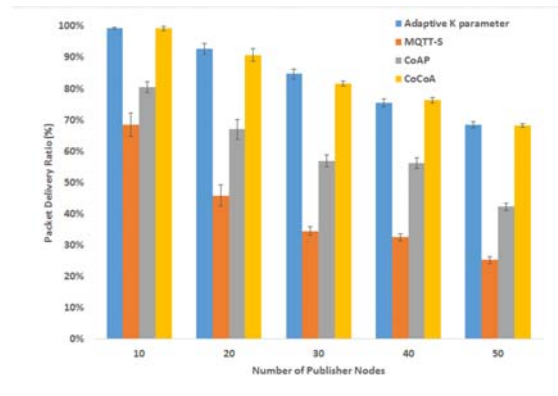


Fig. 11. PDR Comparison for QoS Level 1 using our proposal with other protocols

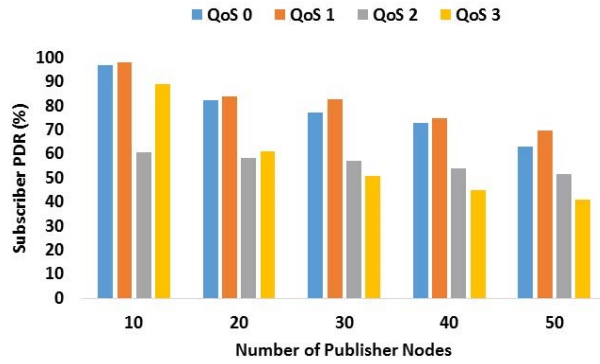
As we have observed in this section, the subscriber node using our adaptive RTO method gets a higher PDR than the other shown protocols. From this point of the article, we focus on the other QoS levels of our proposal in order to provide data aggregation and timeliness.

### 3.4 Data Aggregation QoS Level 2

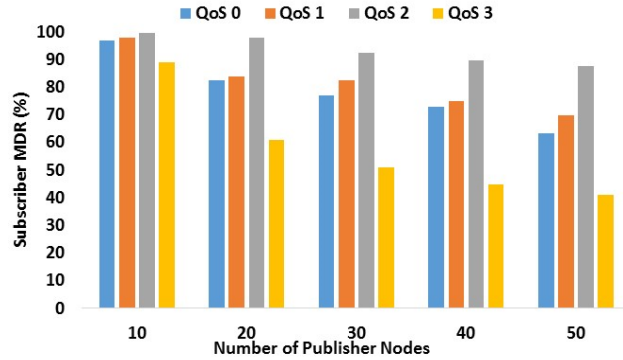
Data aggregation plays an important effect in the network traffic, since it allows transport a collection of values (messages) from several packets into a single packet, thus helping to reduce the network congestion. In the presented results, the broker node aggregates into a single packet the collected values from the next two packets after a publication packet lost, publication acknowledgment lost or spurious retransmission has occurred previously for the subscriber node. We evaluate the effect of the Data Aggregation by evaluating the PDR at the Subscriber node taking into account the same effect in the Subscriber Message Delivery Ratio (MDR). Results showed that PDR for subscriber node with QoS Level 2 was up to 37% (10 nodes) and 18% (50 nodes) lower than subscriber node with QoS Level 1, as can be seen in Fig. 12. One of the reasons for this is that the broker node stops the packet transmission to subscriber nodes with QoS Level 2 each time the data aggregation is performed. At this moment, channel occupation is reduced thus the other subscriber nodes take advantage to receive publication packets and to access the channel with less contention to send its confirmation packets.

In contrast, one may observe in Fig.13, that the MDR for subscriber node with QoS Level 2 was up to 2% (10 nodes) and 18% (50 nodes) greater than the one obtained by subscriber node with QoS Level 1. The reason for this is that each received publication packet by subscriber node with QoS level 2 could contain up to 2 messages, thus it increases the amount of received messages in comparison with the others subscriber nodes that only could receive one message for each received

packet. It can be noted that the network congestion increases as the number of publisher nodes increases, which in turn results in an increase of packet losses. In this case, the subscriber node with QoS Level 1 will attempt to recover each lost packet by retransmitting it (1 message). However, this situation could lead to increase the network congestion and thus the subscriber node with QoS Level 1 gets a lower MDR. On the other hand, the same situation could happen for the subscriber node with QoS Level 2, but in this case, data aggregation helps to recover a higher number of publication messages, because of each retransmitted publication packet could contain up to 2 publication messages.



**Fig. 12.** Subscriber PDR Comparison between different QoS Levels



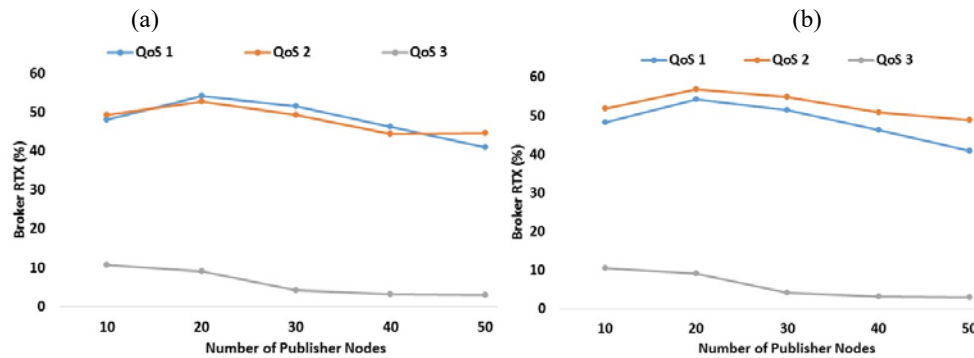
**Fig. 13.** Subscriber MDR comparison between different QoS Levels

Although data aggregation has been added, the subscriber node does not decrease its number of received messages regarding messages generated; it demonstrates the feasibility of our proposal for these kind of networks.

The purpose of both figures is to demonstrate the differences between QoS level 1 and 2. QoS level 2 carries out Data aggregation, which results in an increase of the number of messages received by the subscriber node. This is because two data messages are put into a packet each time data aggregation is activated. **Fig. 12** shows that subscriber node with QoS level 2 gets a lower PDR than QoS level 1. The reason is that PDR metric considers the received packets by the subscriber node instead of the received data messages that considers the MDR metric. In contrast, if we consider the received messages by subscriber node as shown in **Fig. 13**, it is clear that subscriber node gets a number of received messages greater than subscriber node with QoS level 1. Finally, the choice of the QoS level 1 or 2 depends on application requirements. The use of QoS level 2 could increase the delay

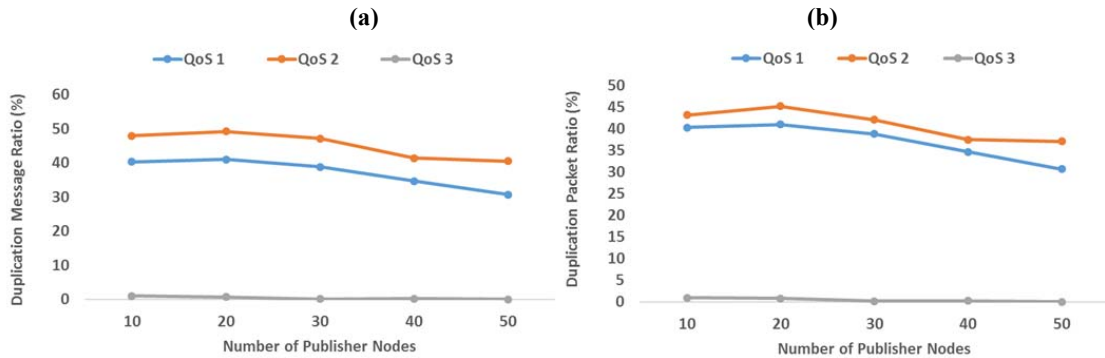
due to the time required to perform data aggregation. We recommend the use of QoS level 1 in the cases that application requires the packet of delivery with reliability once the publisher nodes have generated them. On the other hand, if the application requires reliability but it is able to tolerate delay in the delivery of packet we recommend QoS level two. Finally, if a requirement of the application is to receive data packets with a deadline target, the QoS level 3 is suitable because it provides timeliness.

In **Fig.14(a)** one may observe that retransmitted packet ratio is up to 4% (50 nodes) lower for subscriber node with QoS level 2 than subscriber node with QoS Level 1, the reason of that is because of the data aggregation process which allows retransmit up to 2 publication messages into a single publication packet. In contrast, **Fig.14(b)** shows that the subscriber node with QoS level 2 requires up to 8% (50 nodes) higher publication message retransmitted than subscriber node with QoS Level 1. Although the number of retransmitted messages for subscriber node with QoS Level 2 is higher than subscriber node with QoS Level 1, as illustrated in **Fig.14(b)**, this number of retransmissions is necessary to recover the publication message in order to obtain the highest PDR.



**Fig. 14.** Effect of the number of Publisher nodes on the Broker RPTXR (a) and Broker RMTXR (b) using different QoS Levels

In the case of Publication Duplicated Packet Ratio (DPR), in **Fig. 15(a)** and **Fig. 15(b)** the subscriber node with QoS Level 2 gets in both cases up to 9% (50 nodes) greater number of duplicated publication messages than subscriber node with QoS Level 1. That is due to, among other reasons, publication acknowledgment lost which in turn results in retransmission of publication message. In this situation, each received duplicated publication packet by subscriber node with QoS Level 2 could contain up to 2 publication message thus increasing the number of duplicated publication messages.



**Fig. 15.** Effect of the number of Publisher nodes on the Duplicated Publication Message Ratio (a) and Duplicated Publication Packet Ratio (b) from Subscriber Node using different QoS Levels

### 3.5 Timeliness QoS Level 3

We consider a useless packet if a packet has not been delivered before the deadline expires. In this section, we evaluate the effect of timeliness on subscriber node using our proposed QoS level 3. As we can observe in Fig.16, QoS level 3 ensures the timeliness delivery of packets up to 81%. It is also important to note that although the network load increases, the decrease of the packet timeliness ratio is reduced. It is important to emphasize that this QoS Level is focused on timeliness of delivered packets to the subscriber node. Broker node is in charge of transmitting only those packets that the deadline mechanism considers that they are able to accomplish with the deadline target. We are not interested on subscriber PDR obtained, because it could be lower as more packets are discarded.

In contrast, we could deduce that the energy consumption is lower using QoS level 3 because broker node would discard every packet not accomplish with the deadline target.

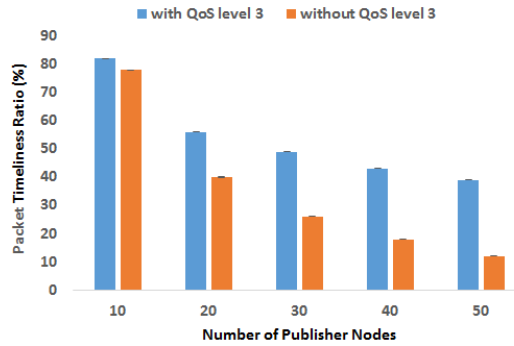
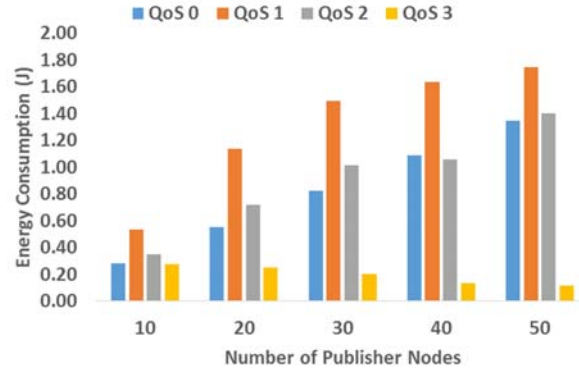


Fig.16. Effect of using Timeliness (QoS Level 3) for packet delivery

### 3.6 Energy Consumption

We focus on the energy drawn by broker node which is responsible to transmit publication packets to each subscriber node with corresponding QoS level. As we can observe on Figure 17 the results show that publication packets transmitted with data aggregation (QoS level 2) consume up to 0.59 J (40 nodes) lower energy than without this feature (QoS level 1). One reason for that is because broker node reduces the number of transmitted packets to Subscriber node.

In contrast, we get the lowest energy consumption with QoS level 3 and QoS level 0. That is because this QoS level is not focused on reliable packet delivery. QoS Level 3 as we above mentioned is focused on timeliness. Therefore, the amount of energy drawn is proportional to the number of publication packets transmitted before deadline expire. On the other hand, publication packets are transmitted on best effort for QoS level 0. Therefore, there is not addition of energy consumption for retransmission compared with the others QoS levels.



**Fig.17.** Energy Consumption of Publication packets for each QoS Level

#### 4. Conclusions

As mentioned above, for a smart way monitoring of critical infrastructures (such as electrical substation) it is important consider the reliable delivery packet and timeliness, which requires that the occurrence of an event is near real-time notified.

To meet these stringent requirements, in this work, we have proposed and discussed a mechanism that establishes different QoS levels to provide delivery of packet and timeliness on WSN based on Publish/Subscribe model to meet application requirements. The QoS Level 1 provides a reliability in the delivery of packets through an adaptive RTO mechanism that adjusts a K parameter value depending on the subscriber node PDR. The evaluation results showed that subscriber obtains an increase in PDR.

The QoS Level 2 is similar to previous QoS Level 1 but in addition, it provides data aggregation support in order to reduce network congestion situation. We showed with the results that QoS Level 2 provides a higher MDR on the subscriber node comparing with QoS Level 1. Finally, QoS Level 3 provides timeliness on the delivery packet as we showed in the obtained results. Our proposal provides several QoS Levels in order to adapt to different application requirements. Application on subscriber node could choose a QoS level 2 that provides data aggregation instead of QoS level 1. However, it is important to note that this QoS level could increase the delay in the packet delivery to the subscriber node due to the time required to perform the data aggregation. We recommend QoS level 2 for applications that require a reliability on packet delivery and delay tolerance. Finally, the decision of QoS level will depend on application requirements. In case a timeliness mechanism is required for an application on subscriber node, we recommend the use of QoS level 3 mechanism presented in this paper since this technique discards a packet that will not accomplish with the defined deadline. In addition, the network benefits by saving energy resources and reducing congestion.

Finally, based on the insight gained, for a future work, we propose the analysis and evaluation of the effect of the routing protocol in the performance of our proposal. Besides, it could propose a cross layer architecture to get information from the others layers in order to enhance the adjusting of the RTO to obtain a better PDR and packet timeliness and also get energy and bandwidth efficiency.

## References

- [1] Ian. Akyildiz, Mehmet Can Vuran, “Wireless Sensor Networks,” *John Wiley & Sons*, 2010. [Article \(CrossRef Link\)](#).
- [2] Yasin Tekin, Ozgur Koray Sahingoz, “A Publish/Subscribe messaging system for wireless sensor networks,” *IEEE Sixth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pp. 171 – 176, 2016. [Article \(CrossRef Link\)](#).
- [3] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, Anne-Marie Kermarrec, “The many faces of publish/subscribe,” *ACM Comput. Survey*. vol. 35, pp. 114–131, 2003. [Article \(CrossRef Link\)](#).
- [4] Ernesto García Davis, Anna Calveras, Ilker Demirkol, “Improving Packet Delivery Performance of Publish/Subscribe Protocols in Wireless Sensor Networks,” *Sensors*, vol. 13, no. 1, pp. 648–680, 2013. [Article \(CrossRef Link\)](#).
- [5] Anup Burange, Harshal Misalkar, “Review of Internet of Things in Development of Smart Cities with Data Management & Privacy,” *IEEE Conf. International Conference on Advances in Computer Engineering and Application (ICACEA)*, pp. 189 – 195, 2015. [Article \(CrossRef Link\)](#).
- [6] Kehua Su, Jie Li, Hongbo Fu, “Smart city and the applications,” *International Conference on Electronics Communications and Control (ICECC)*, pp. 1028–1031, 2011. [Article \(CrossRef Link\)](#).
- [7] Feng, Xia, “QoS Challenges and Opportunities in Wireless Sensor/Actuator Networks,” *Sensors* vol. 8, pp. 1099–111, 2008. [Article \(CrossRef Link\)](#).
- [8] P.N. Renjith, E. Baburaj, “An analysis on data aggregation in Wireless Sensor Networks,” *International Conference on Radar, Communication and Computing (ICRCC)*, pp 62– 71, 2012. [Article \(CrossRef Link\)](#).
- [9] Kiran Maraiya, Kamal Kant, Nitin Gupta, “Wireless Sensor Networks: A review on data aggregation,” *Journal of Scientific & Engineering Research*; vol. 2, no. 4, 2011. [Article \(CrossRef Link\)](#).
- [10] Peter Korteweg, Alberto Marchetti-Spaccamela, Leen Stougie, Andrea Vitaletti, “Data aggregation in sensor networks: Balancing communication and delay costs,” *Theoretical. Computer. Science*, vol. 410 no. 14, pp. 1346–1354, 2009. [Article \(CrossRef Link\)](#).
- [11] E. Felemban, C. G. Lee, E. Ekici, R. Boder, S. Vural, “Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks,” *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2646–2657, 2005. [Article \(CrossRef Link\)](#).
- [12] E. Felemban, C. G. Lee, E. Ekici, “MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks,” in *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006. [Article \(CrossRef Link\)](#).
- [13] Felix Dobslaw, Tingting Zhang, Mikael Gidlund, “QoS-Aware Cross-Layer Configuration for Industrial Wireless Sensor Networks”, in *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1679–1691, 2016. [Article \(CrossRef Link\)](#).
- [14] Mohamed Zaki Hasan, Hussain Al-Rizzo, Fadi Al-Turjman, “A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Wireless Multimedia Sensor Networks,” in *IEEE Communications Surveys & Tutorials* , vol.19, no.3, pp.1424–1456, 2017. [Article \(CrossRef Link\)](#).
- [15] Mohsen Sharifi, Majid Alkaee Taleghan, Amirhosein Taherkordi, “A Publish-Subscribe Middleware for Real-Time Wireless Sensor Networks,” *International Conference on Computational Science; ICCS 2006*, LNCS, 2006, vol. 3991/2006, pp. 981–984, 2006. [Article \(CrossRef Link\)](#).
- [16] Mohsen Sharifi, Majid Alkaee Taleghan, Amirhosein Taherkordi, “A Middleware Layer Mechanism for QoS Suppor in Wireless Sensor Networks”, in *Proc. of the Fifth International Conference on Networking (ICN)*, 2006. [Article \(CrossRef Link\)](#).
- [17] Eduardo Souto, Germano Guimarães, Glaucio Vasconcelos, Mardoqueu Vieira, Nelson Rosa, Carlos Ferraz, Judith Kelner, “Mires: a publish/subscribe middleware for sensor networks,” *Personal and Ubiquitous Computing*, vol. 10, no. 1, pp 37–44, 2005. [Article \(CrossRef Link\)](#).
- [18] Giovanni Russello, Leonardo Mostarda, Naranker Dulay, “A policy-based publish/subscribe middleware for sense-and-react applications,” *The Ninth International Conference on Quality Software. Journal of Systems and Software*, vol. 84 no. 4, pp.638–654, 2011. [Article \(CrossRef Link\)](#).
- [19] Ke Shi, Zhancheng Deng, “TinyMQ: A content-based publish/subscribe middleware for wireless sensor networks,” in *Proc. of the Fifth International Conference on Sensor Technologies and Applications*

- (*SENSORCOM*), pp. 12-17, 2011. [Article \(CrossRef Link\)](#).
- [20] Hakan Cam, Ozgur Sahingoz, Ahmet Sonmez, "Wireless sensor networks based on publish/subscribe messaging paradigms," in *Proceedings of the 6th International Conference on Advances in Grid and Pervasive Computing, (GPC)*, Springer-Verlag, Berlin, Heidelberg, pp. 233–242, 2011. [Article \(CrossRef Link\)](#)
- [21] Jaime Chen, Manuel Diaz, Bartolome Rubio, Jose Troya, "PS-QUASAR: A publish/subscribe QoS aware middleware for Wireless Sensor and Actor Networks," *Journal of System and Software*, vol. 86, no. 6, pp. 1650-1662, 2013. [Article \(CrossRef Link\)](#).
- [22] Hui-Ling Chang, Cheng-Gang Wang, Mong Ting Wu, Meng-Hsun Tsaiet, Chia-Ying Lin, "Gateway-Assisted Retransmission for Lightweight and Reliable IoT Communications," *Sensors*, vol. 16 no. 10, 2016. [Article \(CrossRef Link\)](#)
- [23] Pruet Boonma, Junichi Suzuki, "TinyDDS: an interoperable and configurable publish/subscribe middleware for wireless sensor networks," *Principles and Applications of Distributed Event-Based Systems*, pp. 26, 2010. [Article \(CrossRef Link\)](#).
- [24] Kai Beckman, Olga Dedi, "sDDS: a portable data distribution service implementation for WSN and IoT platforms," in *Proc of 12th Int. Workshop on Intelligent Solutions in Embedded Systems (IEEE WISES)*, pp. 115-120, 2015. [Article \(CrossRef Link\)](#).
- [25] The proven data Connectivity Standard for the Internet of Things. Data Distribution Service (DDS) for real-time systems. <http://www.omg.org/intro/DDS.pdf>
- [26] Xiaoyu Tong, Edith Ngai, "A Ubiquitous Publish/Subscribe Platform for Wireless Sensor Networks with Mobile Mules," *IEEE 8<sup>th</sup> International Conference on Distributed Computing in Sensor Systems (DCOSS)*. pp. 99-108, 2012. [Article \(CrossRef Link\)](#)
- [27] Urs Hunkeler, Hong Linh Truon, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE)*. pp 791-798, 2008. [Article \(CrossRef Link\)](#)
- [28] Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP), Request for Comments 7252 (RFC7252)", 2014. [Article \(CrossRef Link\)](#)
- [29] K. Hartke. Observing Resources in the Constrained Application Protocol (CoAP). *Request for Comments (RFC7641)* 2015. [Article \(CrossRef Link\)](#)
- [30] C. Bormann. A. Betzler, C. Gomez, I. Demirkol, "CoAP Simple Congestion Control/Advanced. *draft-ietf-core-cocoa-04*," 2017. <https://tools.ietf.org/html/draft-ietf-core-cocoa-02>
- [31] V. Paxson, M. Allman, J. Chu, M. Sargent, "Computing TCP's Retransmission Timer. *Request for Comments: 6298 (RFC6298)*," 2011. [Article \(CrossRef Link\)](#)
- [32] OMNet++ Discret Event Simulator. <https://omnetpp.org>
- [33] IEEE Standard for Information Technology. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs); IEEE Computer Society: Los Alamitos, CA, USA, 2006.
- [34] Zolertia Z1 Datasheet Version C. 2011. [http://zolertia.sourceforge.net/wiki/images/e/e8/Z1\\_RevC\\_Datasheet.pdf](http://zolertia.sourceforge.net/wiki/images/e/e8/Z1_RevC_Datasheet.pdf)



**Ernesto García Davis** is full time professor in the Department of Computer Architecture and Network at Technological University of Panama. He received M.S. degree in Telematic Engineering from Universitat Politècnica de Catalunya, Spain, in 2009. He is currently pursuing the Ph.D. degree with the Telematics Engineering Department at Universitat Politècnica de Catalunya, Spain. His research interests include wireless sensor network, transport protocols, quality of service and Internet of Things architecture.



**Dr. Anna Calveras Augé** obtained the Ph.D. degree from the Universitat Politècnica de Catalunya in 2000. She is associate professor at the mentioned University, at the Telematics Department, in the Wireless Networks Group (WNG). Her research interests and expertise areas comprise the design, evaluation and optimization of communications protocols for wireless multihop networks, ad-hoc networks, wireless sensor networks, the Internet of Things, and application domains such as smart cities and building automation. She has been involved in several National and International research or technology transfer projects, and she has published in International and National conferences and journals.