InRout - A QoS Aware Route Selection Algorithm for Industrial Wireless Sensor Networks

Berta Carballido Villaverde, Susan Rea, Dirk Pesch

Nimbus Centre for Embedded Systems Research
Cork Institute of Technology
Rossa Avenue, Cork, Ireland.

Abstract

Wireless sensor networks are a key enabling technology for industrial monitoring applications where the use of wireless infrastructure allows high adaptability and low cost in terms of installation and retrofitting. To facilitate the move from the current wired designs to wireless designs, concerns regarding reliability must be satisfied. Current standardization efforts for industrial wireless systems lack specification on efficient routing protocols that mitigate reliability concerns. Consequently, this work presents the InRout route selection algorithm, where local information is shared among neighbouring nodes to enable efficient, distributed route selection while satisfying industrial application requirements and considering sensor node resource limitations. Route selection is described as a multi-armed bandit task and uses Q-learning techniques to obtain the best available solution with low overhead. A performance comparison with existing approaches demonstrates the benefits of the InRout algorithm, which satisfies typical Quality of Service requirements for industrial monitoring applications while considering sensor node resources. Simulation results show that InRout can provide gains ranging from 4% to 60% in the number of successfully delivered packets when compared to current approaches with much lower control overhead.

Keywords: Wireless Sensor Networks (WSNs), IEEE 802.15.4, Routing, Industrial, Quality of Service, Reinforcement Learning.

1. Introduction

To allow industry to expand the lifetime of expensive manufacturing equipment and to keep up with today’s dynamic and competitive market, low-cost industrial monitoring systems are required [1]. Traditionally, industrial monitoring systems have been based on sensors with wired communication links. However, wired systems require communication cables to be installed and regularly maintained with a cost of as much as €87 per meter of cable in a chemical plant to an extreme of €4337 per meter in a nuclear power plant. Moreover, hardwired network connectivity may not be practical or feasible where the equipment is located at remote locations or on moving platforms or where network infrastructure is not available. This has led to a sparse deployment of wired monitoring systems in industrial plants [1, 2].

Wireless sensor networks (WSNs) have been gaining in popularity for industrial monitoring applications, as an alternative to wired systems, due to their relatively low costs and ease of retrofitting into existing infrastructure. In an industrial wireless sensor network, small battery operated wireless sensor nodes are attached to industrial equipment to monitor different parameters relevant to the performance of the machinery, such as vibration, pressure, temperature, etc. The sensed data is then wirelessly transmitted to sink nodes where further analysis of the information can be performed. This allows plant personnel to replace or repair equipment before their efficiency falls or they fail completely. This can be of capital importance as, for instance, in a steel manufacturing plant, the typical cost of production downtime is €117k per day or the lost production in a paper mill equates to around €17k per hour [3].
However, even though WSNs present several advantages over traditional wired and wireless systems, there is significant concern as to the reliability of wireless communications due to the unpredictability of the wireless channel [4, 5]. Furthermore, due to the diverse range of applications that can be found in industrial scenarios and their differing requirements, there is a need for quality of service (QoS) provisioning so that the technology can be successfully adopted, especially in terms of reliability, energy and message transmission delays. However, achieving QoS provision in wireless sensor networks is not a straightforward task due to the strict energy, computational and memory limitations within the sensor nodes. These conflicting requirements and limitations are inhibiting the widespread adoption of the technology.

Routing is a key process to consider in WSNs when dealing with QoS requirements as, for instance, any routing decision can have an impact on the network lifetime, packet delivery ratios or end-to-end delays. Recently, several standardisation efforts have been made, from a wireless sensor network communication perspective, with the ultimate goal of designing reliable wireless communication systems on industrial environments [6–8]. Nevertheless, these groups only provide basic recommendations on how to perform routing tasks leaving the design of more comprehensive algorithms open. The IETF ROLL working group [8] identifies multipath routing as the most suitable routing technique for WSN in industrial environments due to the lossy characteristics of the networks. Inline with this, current research efforts in the multipath routing domain for WSN are limited in the sense that they do not consider the limitations imposed by the sensor nodes such as duty cycling schedules, restricted buffer capacities, etc. to perform routing decisions [6–15]. This makes them unsuitable for realistic scenarios.

Motivated by this, the aim of the work presented here is to promote the use of WSNs in industrial monitoring applications by designing an efficient route selection algorithm, InRout, that satisfies network and application QoS requirements while considering the sensor nodes limitations. InRout combines the information available locally at each node and the multiple routes, provided by the underlying multipath routing protocol, to achieve QoS-based route differentiation with low memory consumption and associated overhead. Specifically, the contributions of InRout design are:

- InRout is designed to satisfy the applications QoS requirements while simultaneously considering realistic constrained WSN scenarios (battery, duty cycle, limited buffers, etc). Thus, InRout is designed for realistic wireless sensor network environments.
- InRout design is developed specifically to need minimal memory and computational requirements as well as to create very low control overhead. This makes it suitable for resource constrained nodes.
- Finally, InRout is easily implementable over different underlying multipath routing schemes. This makes InRout a very flexible solution to adopt and enhance the performance of existing basic multipath routing protocols for WSNs.

In designing InRout we utilize Q-learning techniques for their low computational and memory requirements, which makes them suitable for WSN protocol designs. Q-learning is a type of reinforcement learning technique whose operation is based on learning the expected utility of taking a specific action in a given state and following a predefined policy thereafter. Experimental analysis based on computer simulations is used to demonstrate the improved performance over existing approaches and the capacity of InRout to satisfy application QoS requirements.

The rest of this paper is organised as follows. Section 2 analyses application and routing characteristics and requirements for wireless sensor networks in industrial scenarios. This information is used later to analyse and test the proposed algorithm. Section 3 details related work with regard to multipath routing solutions and it gives an introduction to Q-learning techniques. Section 4 describes the operation of InRout. Section 5 describes the simulation scenario and results that accentuate the distinct advantages of the proposed approach when compared with other schemes. Finally, conclusions are drawn and future work is described in Section 6.

2. Industrial WSN Application & Routing Requirements

In this section, we analyse the application and routing requirements for wireless sensor networks in industrial scenarios from the viewpoint of the ISA100 standardisation committee and the ROLL working group. This will assist in identifying the characteristics that should be considered by the proposed InRout route selection algorithm.
The ISA-SP100.11a [16], a standard created by the ISA-SP100 [6] standards committee, was formed with the mission of recommending a wireless communication standard for industrial applications. The ISA-SP100 classifies industrial applications as described below and outlined in [16]. The ISA-SP100.11a’s main focus is on the non-critical application space, which are classes 1 to 5 (see Table 1).

<table>
<thead>
<tr>
<th>Category</th>
<th>Class</th>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>0</td>
<td>Emergency Action</td>
<td>Always Critical</td>
</tr>
<tr>
<td>Control</td>
<td>1</td>
<td>Closed Loop Regulatory Control</td>
<td>Often Critical</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Closed Loop Supervisory Control</td>
<td>Usually Non-Critical</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Open Loop Control</td>
<td>Human in the Loop</td>
</tr>
<tr>
<td>Monitoring</td>
<td>4</td>
<td>Alerting</td>
<td>Short-term Operational Consequences (seconds to minutes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(i.e. Event based maintenance)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Logging &amp; Downloading/Uploading</td>
<td>No Immediate Operational Consequences</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(i.e. History collection, sequence-of-events, preventive maintenance)</td>
</tr>
</tbody>
</table>

Table 1: ISA100.11a Applications Classification

Among all the possible applications our primary focus is on monitoring applications where the successful delivery of data is of paramount interest and where the applications required delays are in the order of seconds to minutes [17]. This is driven by a recent study [17] carried out by the International Society of Automation where, when considering deploying a wireless sensor network in a factory installation, 88.8% of interested companies (users and vendors) would use the network for monitoring purposes (condition and process monitoring). More critical uses, such as high-speed control applications were only considered by 13% of those companies surveyed.

Thus, with regard to monitoring applications (pressure, vibration, switches, signal level, flow, fluid levels, volts, amps, phase, etc.), the ISA-SP100 standards family describes the following considerations [18]: Monitoring typically requires the periodic transmission of a limited amount of data, such as pressure, temperature or signal level. For many applications, reports can take place on a periodic schedule from one to several minutes. However, some applications necessitate an update every few seconds. With regard to reporting periodicity and latency requirements, monitoring messages may be periodic and/or event-driven. To capture varying traffic rates, as part of the experimental analysis in Section 5, a range of packet interarrival times are used to reflect this application traffic diversity.

According to the Networking Working Group (NWG) of the Internet Engineering Task Force (IETF) [8], typical industrial scenarios may have multiple sinks with the number of sinks being far smaller than the total number of nodes. Networks may be composed of between 10 to 200 field devices and usually the maximum number of hops is 20. An example of a typical industrial topology is presented by ISA SP100.11 in [19] (see Figure 1). It is assumed that the field devices themselves will provide routing capability for the network. In addition, targeted field devices should be small and easily deployed with reduced battery and memory capacity. The wireless devices should be able to operate in a wide range of environmental conditions found in industrial scenarios. Also, it is generally expected that nodes with routing capabilities will be stationary as well as the sinks that will be connected to the backbone (see Figure 1).

Because of the lossy nature of wireless networks, a basic requirement for the routing protocol in place is the provision of multiple paths towards the destination with potentially different costs. The routing protocol should be capable of establishing metrics that can be used to weight links when computing a route with regard to some objective function (i.e. minimise packet loss). It must also route over paths that are capable of supporting application requirements. Furthermore, the routing protocol is also required to support the ability to recompute paths based on underlying link attributes/metrics that may change dynamically. This is the basic motivation and design principle of the proposed InRout algorithm. Section 4 outlines the design of the InRout algorithm and the mechanisms used to capture the requirements referred to above and summarised next.

1. Low to high reliability ≈ 80%-100% packet success, i.e. packet error rate (PER) in the range of 0-20% [17].
2. Medium to high reporting frequency ≈ seconds to minutes.
3. Soft Latency requirements ≈ seconds to minutes.
4. Event-based or periodic reporting.
5. Scalability - networks can be composed of up to hundreds of sensors and multiple sinks.
6. Redundancy of paths from the publisher to the subscriber - because of the lossy nature and changing conditions of the network.
7. Capacity to balance the resources among different applications/flows.
8. Capacity to adapt to application requirements and network changes.
9. Support node-constrained routing - energy, memory, etc.

With these requirements to the fore in the next section we analyse current approaches towards routing in wireless sensor networks suitable for industrial scenarios.

3. Related Work

The basic requirement for industrial communication systems is the provision of multipath routing as a consequence of the instability of the wireless medium. As a single link failure could compromise the communication flow if only one route is available, path redundancy is absolutely necessary. As multipath routing is a fundamental requirement, we focus here on analysing existing multipath routing protocols and their route selection algorithms in particular. In addition, for reliable routing in industrial WSNs, node energy, buffer limitations and duty cycles must also be considered. Duty cycling is driven by the data transmission load, offered by applications running over the WSN, and the need for reducing energy consumption to prolong network lifetime. A low duty cycle may be more energy efficient but it restricts the offered load as a sensor node’s sleep period is longer. Furthermore, lower duty cycles can translate into more contention as the available time for transmission is reduced. Therefore, when estimating the reliability of links and delays, the duty cycle schedule must be considered. Energy must be considered as well, as using the same battery operated nodes for forwarding repeatedly can lead to rapid energy drain in these nodes potentially causing network partitioning and orphan nodes. Managing energy and buffer capacities distributes the load on the network and eases localised congestion. This reduces the likelihood of buffer overflows and the need for retransmissions, all of which contributes to more reliable routing.

3.0.1. Standards Routing

Looking at industrial standardisation groups, both WirelessHART [7] and ISA100.11a [16] employ Graph Routing and Source Routing algorithms, with Graph Routing being used in most cases. Both standards use a central manager that, after obtaining information relating to the nodes connectivity and status, computes multiple routes for every node and downloads them to the individual nodes. However, details on specific mechanisms to compute these routes or select among them are left open.

Work in progress within the ROLL group defines in [20] a routing protocol for low power lossy networks referred to as HYDRO. With this protocol, nodes build multipath routes on demand to the sinks. In order to select the best
available route to the sinks, the nodes use a simple link quality metric. The design of more comprehensive routing metrics is again left open.

Finally, the Zigbee specification [21] also provides recommendations for routing. However, these are based on a simplified version of AODV and Tree Routing that do not comply with the industrial requirement for multipath routing. Having only one available route means that, in industrial scenarios with varying radio channel characteristics, the connectivity is likely to be lost often - this makes this class of protocols unsuitable for such scenarios.

### 3.0.2. Single Metric Route Selection

Proposed in [22] is a hierarchy based multipath routing protocol. This protocol first builds clusters among the different nodes in the network and then nodes select the different possible routes on a round robin basis for energy equalization purposes.

Presented in [23] is a multipath routing protocol for WSNs with the goal of reducing collisions. In order to find collision-free paths, the protocol uses a central base station that gathers information from all nodes in the network.

The authors propose in [15] a variation of the Directed Diffusion algorithm [24] to create multiple node-disjoint or braided routes in WSNs. This data centric protocol builds the routes from sources to different sinks depending on the interests shown by the sinks. This protocol switches from one route to another when the route fails.

The work in [14] presents an extension of the AODV protocol [25] to provide multipath capabilities and is referred to as Ad-Hoc On Demand Multipath Distance Vector Routing - AOMDV. This protocol, originally designed for Ad-Hoc networks, can be also applied to WSNs due to the similarities among both types of network, that is, both consist of distributed autonomous devices with routing capabilities. In AOMDV, each node selects as its main route the route with the lowest number of hops and keeps the other routes as back-up routes. The advantage of AOMDV is that it can build routes on demand and requires low memory consumption and overhead. With regard to AOMDV variations with single metric route selection, the algorithms presented in [26] and [27] propose a simple scheme to improve the network lifetime when the AOMDV routing protocol is used. The algorithms gather the minimal residual energy of each node along every link disjoint route and then, once the information is obtained, they choose as a primary route the route that has the node with the maximum minimum residual energy.

### 3.0.3. Multi Metric Route Selection

Another protocol for multipath routing in WSNs is the MCMP presented in [9] which has a goal of providing soft-QoS to WSN applications in terms of delay and reliability. Although the goal is achieved for the proposed scenario, the protocol does not consider inherent properties of WSNs such as duty cycling or energy and buffer limitations, which is an unrealistic assumption. Moreover, it employs RTS/CTS primitives for gathering channel information which are commonly avoided in energy constrained WSN [28]. Reliability here is modelled as a minimization problem which has to be solved at each node with consequent requirements in processing power.

In [29], another variation of Directed Diffusion is presented, where nodes build multiple paths depending on several metrics and, after the paths are established, routes are selected depending on the type of traffic to be sent (real-time or best effort) to satisfy their delay requirements and again duty cycling is not considered. Segmentation and encoding techniques are combined with the routing protocol to improve success delivery rate.

The authors in [13] propose a routing algorithm, referred to as AdaR, which uses reinforcement learning techniques to calculate routes depending on several metrics. To gather the decision information, at each hop, the full hop information is added to the data packets and rewards are generated at a base station. When the base station receives a required amount of packets, it calculates some weights for the different metrics offline. The number of required packets is however undefined which impedes replicating the experiment. This work has poor scalability and is difficult to apply in WSN since the algorithm makes centralized decisions, which incur into too much overhead and energy costs as every time a packet crosses a node, the routing information is appended. Moreover, buffer limitations are not considered for reliability calculations.

With regard to AOMDV variations with multi metric route selection, the work in [30] proposes a modification of AOMDV, named AM-AOMDV. The protocol considers the RSSI (received signal strength indicator), latency and buffer occupancy as metrics, however no information on how to use these metrics to decide which path is optimal is provided and therefore it is not possible to replicate the experiments in our analysis of the proposed InRout algorithm. Presented in [31] is an ant colony optimization modification of AOMDV to select the main and back-up paths. This protocol uses several metrics to calculate pheromone levels per node per path that later are translated into a probability
to choose the different paths. Since this protocol has not been designed specifically for WSN, it does not take into account the energy consumption which is a fundamental concern for the battery operated sensor nodes. Moreover, the route selection process focuses specifically on mobile node metrics which is not compatible with the scenario targeted here.

3.0.4. Global Positioning Based Route Selection

Proposed in [10] is an extension of MCMP named EMCMP that uses global positioning information to improve the performance of MCMP in terms of energy. Nevertheless, this protocol does not consider buffer limitations or duty cycling and again like MCMP it relies on costly (in terms of resource usage energy, bandwidth) RTS/CTS primitives to obtain network information.

Proposed in [11] is another routing protocol that again uses global positioning information to perform multipath routing in WSN. This protocol uses geographic progress towards the destination, residual energy and expected sojourn time of a packet at the receiving node to perform route selection. This protocol does not consider the nodes duty cycle in the delay calculations. Moreover, the packets lost at the buffer are not taken into account for the reliability estimations.

In [32] the authors propose MMSPEED. This multipath protocol is specifically designed for short-living sensor network applications and requires information on node position and distances to construct routes. Due to the nature of the targeted applications (i.e. short-living applications) it does not consider the energy expenditure of the nodes.

Work in [12] proposes a proactive routing protocol for industrial wireless sensor networks where nodes route packets towards a unique destination sink. In order to select a destination, the protocol uses multiple metrics such as delay, energy and reliability. This protocol requires exact positioning information to perform the routing tasks and to calculate some of the route selection metrics. This protocol does not consider inherent properties of WSNs such as the need for duty cycling or the buffer limitations of the sensor nodes.

Note that the assumption of having precise positioning information in industrial indoor scenarios is unrealistic [33]. Although the previous protocols comply with the multipath requirement, they have deficits. For instance in terms of requiring global accurate positioning information, which cannot be reliably achieved in indoor scenarios [33]. Some existing routing methodologies either lack a route selection algorithm or have only a very basic one, such as only considering energy or hop count. Moreover, there are other protocols that do not consider sensor nodes limitations or application requirements, which makes them unsuitable for realistic scenarios. We consider that having an intelligent route selection algorithm is as important as providing multipath availability since having multiple paths alone does not guarantee having the best possible performance. A good route selection algorithm should consider not only a singular optimization goal, such as minimising energy use, but it should consider all the relevant parameters that affect the network and applications performance. Moreover, it should not require high processing power or create high control overhead or memory consumption due to the limitations of the sensor nodes. In addition, dependency on additional support such as positioning/localisation systems should be avoided as this creates further overhead and significant infrastructure costs and, as mentioned before, it is not possible to provide error free positioning information in indoor scenarios [33]. Considering this, we propose the InRout algorithm, a Q-learning based route selection algorithm that uses the information available at each node to select the different routes in order to satisfy various industrial application requirements and to adapt to different industrial network and channel conditions while respecting node energy and memory restrictions. The InRout route selection algorithm can run over any underlying route discovery protocol be it centralised or distributed. InRout uses Q-learning techniques to select the best possible route, based on current application requirements and network conditions, with low overhead and minimal memory requirements. This means that unlike basic multipath routing algorithms such as Graph Routing or AOMDV, InRout can make intelligent route selection decisions online to provide efficient performance.

4. InRout: A QoS Aware Route Selection Algorithm for Industrial Wireless Sensor Networks

The main operation of the InRout algorithm is based on Q-learning (QL) techniques. QL is well suited for distributed problems, like routing. It has medium requirements for memory and imposes low computational requirements
on individual nodes and low overhead [34]. It needs some time to converge, but it is easy to implement, highly adaptable to topology changes and achieves optimal results [34].

4.1. Q-Learning Based Route Assessment

In many common decision situations, choices are made with the goal of obtaining the best possible reward. In most cases, the choices not only provide some reward but they can also help in acquiring new knowledge that can be used to make future decisions. The main dilemma for all these cases is how to balance the reward maximization, based on the knowledge previously acquired, and how can trying new actions increase the knowledge further. This is known as the exploitation vs. exploration tradeoff in Reinforcement Learning (RL) [35].

Reinforcement Learning (RL) is a machine learning (ML) approach that finds the optimum value through trial-error iterations [36]. A key advantage of RL compared to other ML approaches is that it does not require information about the environment except for a reinforcement signal [37]. Thus, RL is within the ML techniques the most popular method for solving WSN problems. This is because it requires minimal communication overhead and achieves optimal results. This is especially appropriate to embedded systems such as sensor nodes were resources are scarce.

The standard RL model is depicted in Figure 2. The learning agent, for instance a sensor node, has a finite set of possible states $S$ where $s$ represents the current state of the agent. The agent remains in one state at a time which defines its internal state or location in the environment. A set of actions $A$ is linked to the set of states $S$ and there is an associated immediate reward $r(s,a)$ for each of the actions taken within the set. The goal of the agent is to learn the sequence of actions that provide the maximum expected accumulated reward. The RL agent learns the best set of actions, called policy $\pi$, through trial and error interactions with the environment. At each step, the agent selects a possible action $a$ among the set and receives an immediate reward $r$ from the environment for the current state $s$. The agent learns from the environment through the rewards received. The process is then repeated with the objective of increasing the long-run sum of values of the reinforcement signal or reward $r$.

Formally, the model is usually described as a Markov Decision Process (MDP) consisting on the experience tuple $(S,A,T,R)$. $S$ is a discrete set of environment states $s_1,s_2,\ldots,s_n$. $A$ is the set of possible actions from each state $a_1,a_2,\ldots,a_m$. $T(s,a,s')$ is the transition probability from state $s$ to a successor state $s'$ when taking action $a$. Finally, $R(s,a)$ is the reward function obtained when taking an action $a$ [36].

The multi-armed bandit problem, originally described by Robins [38], is an example of a decision problem that can be solved with RL techniques. A multi-armed bandit, also called K-armed bandit, is similar to a traditional slot machine (one-armed bandit) but in general has more than one lever. When pulled, each lever provides a reward drawn from a distribution associated to that specific lever. Initially, the gambler has no knowledge about the levers, but through repeated trials, he can focus on the most rewarding levers. In this work, we consider a multi-armed bandit problem, where each slot machine lever represents a link-disjoint route to the destination, and the agent has to choose the route that maximizes the reward with regard to the application QoS requirements and network and node status at each time. In order to obtain the values of the actions, i.e. selecting the different routes, we use the RL technique known as Q-learning.

Q-learning [39] is a popular form of a Reinforcement Learning algorithm that does not need a model of its environment (i.e. learns without being given any prior information). In the Q-learning process, when an action is selected and executed, in our case the action represents selecting a route, a reward is received representing the quality of that action. This reward is then used to update the Q-value. Over time the node can learn the real action values and therefore select the most suitable. After being initialized to arbitrary numbers, Q-values are estimated as follows:
1. From the current state $s$, select an action $a$. This will cause the receipt of an immediate payoff $r$, and arrival at a next state $s'$.

2. Update $Q(s,a)$ as $Q(s,a) \approx Q(s,a) + x(r + y \cdot \max Q(s',a') - Q(s,a))$, where $x$ is the learning rate and $y$ is the discount factor ($0 < x, y < 1$).

3. Go to 1

The Q-Value $Q(s,a)$ represents the current expected total reward for each action and state pair. At the beginning Q-Values are usually initialized with zeros, representing the fact that the agent knows nothing. The values are updated after receiving a reward. The learning rate ‘$x$’ determines to what extent the newly acquired information will override the old information. A factor of 0 will mean that the agent will not learn anything, while a factor of 1 would make the agent consider only the most recent information. The discount factor ‘$y$’ determines the importance of future rewards. A factor of 0 will make the agent "opportunistic" by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward. If the discount factor meets or exceeds 1, the Q values will diverge.

This algorithm is guaranteed to converge to the correct Q-values with a probability of one if the environment is stationary and depends on the current state and the action taken in it. A lookup table is used to store the Q-values.

For our use case, there is a lookup table for every node composed of Q-values where each route represents a state $s$ (lever of the slot machine). Each action $a$ represents selecting a different route (pulling a lever of the slot machine). Therefore, our Q-function has to measure the value of selecting a route $a$ to reach a sink. To speed up the learning process we set $x$ to 1 and we only consider current rewards so we set $y$ to 0, therefore the Q-values are updated as follows:

$$Q_{\text{new}}(a) \approx Q_{\text{old}}(a) + (r(a) - Q_{\text{old}}(a)) \rightarrow Q_{\text{new}}(a) \approx r(a) \quad (1)$$

Where $Q(a)$ represents the goodness of selecting the route $a$ and this value is updated when the reward $r(a)$ is received after sending a packet through the route $a$. In order to calculate the reward for the routes, the reward function described in section 4.1.1 is used. The selected reward will ensure that the result from the assessment of the routes reflects the network status and considers the application QoS requirements and node constraints. After obtaining the reward, the exploitation strategy described in section 4.1.2 is then used to select the optimal routes depending on the application requirements and Q-values obtained.

The reward is sent by any sink after receiving a data packet and the process is repeated through the nodes in the route until the source node is reached. This means that whenever the action of choosing a route to send a packet has been performed by a node, its Q-value is updated with the reward, that is, the node learns the goodness of the action. To minimize the overhead, the rewards are embedded in the payload of the next beacon frame of the nodes in the selected route after they have received the packet. Note that a beacon frame is a control frame used by IEEE 802.15.4 [40] networks and it is sent at the beginning of each active period. Therefore, if more than one packet is received from a certain node of a certain route in one Beacon Interval (BI), that is, the time between consecutive active periods, only one reward will be produced for that node of that route in that BI.

### 4.1.1. InRout Reward Function

The InRout algorithm uses a reward function based on two different QoS parameters: Packet Error Rate (PER) and energy. Based on the requirements outlined in section 2 we identify the PER and energy as being the parameters that directly affect network goodput and lifetime. Delay is also considered by InRout but using a separate method described in section 5.2. The PER and energy reward is generated once a packet is received at the destination and propagated towards the source embedded in the beacon payloads of each node along the route. Therefore, receiving a reward will trigger sending a reward. The reward sent by a node after receiving another reward is the best Q-value that can be found to reach a destination sink. The reward sent by any destination sink after receiving a packet has a constant set of values $R = [1, 1]$, which are the maximum possible values of reward. The Q-value is composed by two terms, reliability $rl$, which is composed by the link PER and the buffer PER and it is calculated as 1-PER, and energy $e$, which reflects the residual energy. Let $Q$ be the Q-table containing the Q-values for each route. Each value in $Q$ is composed by a set of two elements $e$ and $rl$ as per Equation (2).

$$Q = \left(e_1, rl_1 \cdots e_x, rl_x\right) \quad (2)$$
Where \( x \) represents the route ID, for instance the address of the next hop. Let \( R' \) be a set of two values \([e, rl]\). Every time a reward has to be sent, \( R' \) is calculated as per equation (3). \( R' \) is the Q-value set of the healthiest route in terms of energy \( e \) (max) among the set of routes \((\arg\max)\) with the healthiest reliability \( rl \) that can be found in \( Q \).

\[
R' = \max\arg_{rl} Q
\]  

(3)

Once \( R' \) is known, the reward value \( R \) to be sent, also composed by a set of two values \([e, rl]\), is calculated as per equation (4). This step is only performed to advertise the minimum residual energy of the nodes in the route. If the residual energy \( RE \) of the node \( n \) sending the reward is lower than the energy component of \( R' \), the energy part of the reward is updated to reflect this.

\[
R(e) = \min[RE_n, R'(e)], R(rl) = R'(rl)
\]  

(4)

Where \( RE_n \) is obtained according to equation (5). \( RE_n \) represents the residual energy of the node \( n \). \textit{Cons\_Energy} is the already consumed energy and \textit{Init\_Energy} is the initial battery energy.

\[
RE = 1 - \frac{\text{Cons\_Energy}}{\text{Init\_Energy}}
\]  

(5)

When the reward \( R \) is received at the next node, it is then transformed into the modified reward \( r \), as per equation (6), at the receiving node and, after that, equation 1 is applied. The reward is modified to take into account the PER cost of the receiving node and it is composed of the link and buffer PER.

\[
r = (R(e), c(rl) \cdot R(rl))
\]  

(6)

Where \( c(rl) \) represents the cost introduced by the node \( n \) receiving the reward in terms of PER. As can be seen in (6), the PER reward is updated in a multiplicative fashion. The cost \( c(rl) \) is calculated as per equation (7).

\[
c(rl) = (1 - PER_L) \cdot (1 - PER_B)
\]  

(7)

Where \( PER_L \), calculated in (8), is the packet error rate of the link between the node that sends the reward and the node that receives it. \( STx_L \) represents the successful packet transmissions over the link and \( Tx_L \) is the total number of transmissions performed over the link.

\[
PER_L = \frac{STx_L}{Tx_L}
\]  

(8)

In addition, \( PER_B \), calculated in (9), is the packet error rate at the buffer, and considers the packets lost at the buffer due to overflow. \( S\text{Enq\_Pkt} \) is the successfully enqueued packets at the buffer and \( \text{Enq\_Attempts} \) is the enqueuing attempts (dropped and enqueued packets). Both the link and buffer PER variables have a window length of 16 values (16bits need to store this). This value is chosen in order to minimize memory usage and to still maintain sufficient measurement granularity.

\[
PER_B = \frac{S\text{Enq\_Pkt}}{\text{Enq\_Attempts}}
\]  

(9)

Finally, if a beacon is lost from any node that acts as parent for any route, the Q-value is updated to consider this, i.e. a negative reward for that route is applied. The Q-value is updated as follows:

\[
Q(rl) = Q(rl) \cdot \frac{c(rl)_{\text{new}}}{c(rl)_{\text{old}}}
\]  

(10)

Where \( c(rl)_{\text{old}} \) represents the stored value for the \( c(rl) \) of that route. The \( c(rl)_{\text{new}} \) value is the old \( c(rl) \) cost updated where the link packet error rate (PER\_L) has been increased to take into account the loss of a beacon.

We consider that all of the factors selected for the cost and reward functions are necessary to support efficient routing in industrial scenarios with QoS and node constraints, as identified in Section 2. Regarding energy, evenly balanced energy consumption throughout the network increases its lifespan. Considering buffer capacity is important.
as full buffers translate into dropped packets and buffer sizes in sensor nodes are restricted. Link status is essential, because bad link quality corresponds to lost packets and retransmissions which increase energy consumption and buffer occupancy. It is worth noting that the link PER not only takes into account the link quality but also the collisions that are likely to happen in contention based IEEE 802.15.4 networks.

Note that in the case of a route or node failure it is the role of the underlying routing protocol to look for more routes, erase the routes, etc. If no action is taken by the underlying protocol, InRout can ignore any route if several consecutive beacons are lost for the next node on a route. If after some time, beacons are received again for the ignored route, InRout can take up the route again.

4.2. The Exploitation vs. Exploration Strategy

Having a dynamic network means that there will be no optimal solution for a long period of time. Rather than that, there will be changing optimums and the job of our route selection strategy involves assessing how the network changes and how the routes should be explored and exploited so the route information reflects the current network status and the optimal route is selected at all times.

At network start-up nodes do not necessarily have any information relating to the quality of the routes (these routes can be generated by a centralised network manager or on demand locally depending on the underlying route discovery protocol). Therefore, at network start-up the nodes (referred to as source nodes) explore each of their available routes using a deterministic strategy (round robin) until convergence is achieved.

After this, the source has two functions, exploiting the best route so packets are sent through the best possible path and exploring other routes so their route information is kept up to date. However, assessing how much the source should explore some routes and exploit some others needs to be carefully considered as, for instance, the source should not explore too many routes that are providing very low Q-values as this could compromise delivering the exploration packets to the destination.

$\epsilon$-greedy is possibly the simplest and the most extensively used strategy to solve the bandit problem [35]. The $\epsilon$-greedy strategy consists of choosing a random lever with $\epsilon$-frequency, and otherwise choosing the lever with the highest estimated mean, the estimation being based on the rewards observed thus far. $\epsilon$ must lie in the open interval $(0, 1)$.

In its basic form the $\epsilon$-greedy strategy is sub-optimal because asymptotically, the constant factor $\epsilon$ prevents the strategy from getting randomly close to the optimal lever. A variant of the $\epsilon$-greedy strategy is the $\epsilon$-decreasing strategy [35]. The $\epsilon$-decreasing strategy consists of using a decreasing $\epsilon$ for getting arbitrarily close to the optimal strategy asymptotically. The lever with the highest estimated mean is always pulled except when a random lever is pulled instead with an $\epsilon_t$ frequency where $t$ is the index of the current round.

In this work, since the best route (lever) might not remain optimum for a long period of time, we propose a strategy that combines the $\epsilon$-decreasing with an $\epsilon$-increasing policy. The idea behind this approach is that as long as the network remains stable, we can use the $\epsilon$-decreasing approach to reach the optimum. But once conditions start to change, we need to have more exploration and reduce the exploitation to discover if, with the new conditions, the optimums have changed. The optimal route will be that route whose PER Q-value satisfies the PER requirements of the application and whose energy Q-value is the highest, that is, the healthiest route in terms of energy among those that satisfy the PER. More specifically, the strategy works as follows (Algorithm 1 shows the InRout pseudo code for the exploration strategy of the routes):

1. Explore all possible routes to a destination on a round robin basis for a limited number of rounds.
2. Use an $\epsilon$-decreasing strategy.

Select optimum route with probability $1-\epsilon$ and any of the rest of routes with frequency $\epsilon$

When a suboptimal route should be selected, use relative probability $P_i$ among those routes to decide on which one to use:

$$P_i = 1 - \frac{1 - Q(rl)_i}{\sum_{j=1}^{N}(1 - Q(rl)_j)}$$

$N$ is the number of suboptimal routes, $Q(rl)$ is the PER component of the Q-value, and $i$ the index of each route. Then, the policy to select a route $a$ follows equation (12), where $rl_{req}$ is the reliability requirement of
the packet:

\[
\Pi(a) = \begin{cases} 
P_a(Q(rl)) & \text{if } \text{rand} < \epsilon, \\
\max_{\epsilon}[Q(e)|Q(rl) \geq rl_{req}] & \text{otherwise}.
\end{cases}
\]  

(12)

Where \( \text{rand} \) is a random number uniformly distributed in the \((0, 1)\) open interval.

3. When the status of the optimal routes deteriorates due to changes in the network, use an \( \epsilon \)-increasing policy until new convergence is reached. At this point use again the \( \epsilon \)-decreasing policy.

Algorithm 1 InRout Algorithm

1: function InRout ★ Initialise last_opt = 0, \( \epsilon = \epsilon_{\text{max}} \);
2: if New Packet Ready then
3:   if Packet_type == Delay_constrained then
4:     next_route = min hop count route
5:   else
6:     if Initial Phase then
7:       next_route = Round Robin
8:     else
9:       next_route = \( \Pi(a) \)
10:      if rand \( \geq \epsilon \) then
11:         if next_route \( \neq \) last_opt and \( \epsilon < \epsilon_{\text{max}} \) then
12:            \( \epsilon = \epsilon + \text{step} \)
13:         else if next_route == last_opt and \( \epsilon > \epsilon_{\text{min}} \) then
14:            \( \epsilon = \epsilon - \text{step} \)
15:      end if
16:   end if
17: end if
18: end if
19: if Reward Received then
20:   \( Q(a) = r(a) \)
21: end if
22: end function

The convergence time for InRout, considering static conditions, is shown in Equation 13 where \( N \) is the network size and \( H \) is the number of one-hop neighbours (or routes) for each node. For our working case of 60 nodes, this implies 240 iterations if every node has 4 possible routes. This is the worst case scenario as this would imply that every node would have to explore every single route available which is highly unlikely. In a normal scenario, once a node explores a route, the other nodes along that route would not need to explore it again as they would obtain rewards for forwarding the packets. Thus, the real convergence could be several tens lower depending on how the nodes start exploring the network. Note that during this time a round robin strategy is used, therefore the energy consumption would be even among all nodes during this period.

After this number of steps the protocol converges and the exploration phase ends if the network remains stable. If the network conditions change, the protocol uses probabilistic exploration/exploitation strategies to find the new optimal routes. For such cases with unstable conditions, global convergence is not achieved but rather than that the protocol adapts to the temporary optimal solutions.

\[
I < N \cdot H = O(N \cdot H)
\]  

(13)

4.3. Memory Requirements and Scalability

One of the requirements for designing a routing algorithm for wireless sensor networks is the consideration of resource constraints such as energy and memory. Energy use is managed providing energy balancing mechanisms. Memory requirements are kept low by limiting the resources in terms of bit size needed by the QL-based algorithm.
In order to keep the memory consumption to a minimum, the Q-value of every route is limited to 8 bits. Four bits are used for the PER information and 4 bits are used to store the energy information. Using 4 bits means that there are 16 intervals with an interval size of 6.25% in the range 0-100% over which PER and energy values are mapped to, i.e. a binary value of 1111 is used to quantify PER or energy values in the range 93.75%-100%, as shown in Table 2. Mapping the values into regions also ensures that the selection of optima does not fluctuate frequently.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>InRout</th>
<th>AOMDV</th>
<th>EARQ</th>
<th>HYDRO</th>
<th>MRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Sinks per Node</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max. Routes per Sink</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffer PER (per node)</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Link PER (per link)</td>
<td>16</td>
<td>0</td>
<td>32 -2xint</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q-Value (per route)</td>
<td>8</td>
<td>0</td>
<td>64 -2xfloat</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cost to advertise (per node)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next, Last, Destination Addr (per route)</td>
<td>48 -3xint</td>
<td>48 -3xint</td>
<td>48 -3xint</td>
<td>48 -3xint</td>
<td>48 -3xint</td>
</tr>
<tr>
<td>Hop Count / Delay Cost (per route)</td>
<td>3</td>
<td>16 -1xint</td>
<td>16 -1xint</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Positioning Info (Latitude, Longitude)</td>
<td>0</td>
<td>0</td>
<td>64 -2xfloat</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Energy status (per node)</td>
<td>32 -2xint</td>
<td>0</td>
<td>32 -2xint</td>
<td>32 -2xint</td>
<td>32 -2xint</td>
</tr>
</tbody>
</table>

Table 2: PER and Energy Q-value Mappings

In addition, three bits are used to store the number of hops to every sink. This means that the number of hops from a node towards any sink is limited to 8. This implies that any two sinks may be separated by a maximum of 16 hops. For instance, if we consider a maximum indoor transmission range of 30 meters [41] per node, this translates into a maximum separation of 480 meters between any two sinks. This is a reasonable assumption even for large manufacturing plants (however additional sinks may be necessary for harsher environments).

Considering the 8 bits needed for the PER and Energy Q-values, together with a limitation on the number of destination sinks per node to two and routes per sink to four, the maximum memory consumption per node is shown in Table 3 for the InRout algorithm and the test case algorithms AOMDV, EARQ, HYDRO and MRE (section 5.2 explains the reasoning behind the selection of these algorithms) with AOMDV being used as the baseline algorithm for the analysis.

As can be seen, InRout consumes 136 bits more than the baseline reference algorithm AOMDV and 104 bits more than MRE. This represents 1.5 % of the SRAM memory available on a MicaZ node [41] (4kB of memory available in total) which is modelled in the simulation analysis presented in section 5. Nevertheless, simulation results show that while InRout has slightly higher memory consumption than AOMDV or MRE, the performance of InRout is significantly better than those algorithms. InRout has significantly lower memory consumption than the HYDRO and EARQ algorithms. The control overhead introduced by InRout, shown in Table 4, and later as part of the experimental analysis in Table 8, is lower than all test algorithms except for AOMDV. This is to be expected for AOMDV as it uses hop count as the only metric for route selection and this is stored in the route tables and not advertised.

Finally, as the InRout algorithm ensures that the memory consumption remains constant independent of the size of the network, the scalability of the algorithm is guaranteed.
5. Experimental Analysis

This section presents a detailed computer simulation based performance analysis of the proposed InRout route selection algorithm.

5.1. Experiment Design

In designing the experimental environment we rely on the multihop IEEE 802.15.4 OPNET simulation model [42] developed with OPNET Modeler [43] as a basis for implementing and testing the proposed InRout algorithm.

Typical industrial scenarios may have multiple sinks with the number of sinks being far smaller than the total number of nodes. Networks may be composed of between 10 to 200 field devices and usually the maximum number of hops is 20 [8]. For experimental purposes we define a random network topology as shown in Figure 2 to analyze the performance of the InRout route selection algorithm. To satisfy the InRout algorithm memory restrictions and maximum hop count limit, as defined in Table 2 above, and assuming the use of MicaZ nodes [41] for monitoring purposes, with a maximum indoor transmission range of 25m on maximum power (MicaZ datasheet specifies 20m-30m for indoors) and a hop count limit of 8, this gives a maximum diagonal distance of 184m. This corresponds to a planar area of approximately 130m x 130m. In defining the number of nodes for the simulation model we consider a smaller average transmission distance of approximately 18m based on experiences with empirical measurement campaigns which gives a total of 60 nodes for the simulation environment.

MicaZ nodes are typical examples of low power wireless sensor nodes used for monitoring applications. They have a SRAM memory size of 4kB and are powered by two AA batteries. We fix the number of source nodes and vary the offered load instead of varying the node density in an effort to keep the MAC layer beacon scheduling period reduced and to limit excessive simulation run times. Varying the number of source nodes without increasing the node density has a similar effect as varying the load. The maximum distance from any node to any sink is 8 hops (based on the maximum number of bits available, i.e. 3 bits to store the number of hops) and there are between 1 to 9 neighbors per node. Nodes select a destination sink randomly respecting the 8 hops limitation.

As described previously in section 2, typical industrial applications may generate periodic and event-based data with low to high bandwidth requirements. These applications are part of classes 4-5 of the ISA classification for industrial applications described before, which are monitoring applications with short-term or no immediate operational consequences. To include these types of applications in our simulation scenario, nodes will generate single packets following a Poisson distribution and we simulate different mean packet interarrival times (offered loads) to analyze the performance of the InRout algorithm under different levels of stress.

We use data frames sizes of 127 bytes which is the maximum possible size in IEEE 802.15.4 networks. Since the sensor nodes have strict memory limitations, the buffer size at the MAC layer for all nodes is restricted to 10 packets. A buffer size of 10 packets is chosen based on the default buffer size used in the IEEE 802.15.4 MAC standard implementation for TinyOS-2.x developed by the Telecommunication Networks Group (TKN) group [44]. A buffer size of 10 packets using the maximum frame size corresponds to 31% of the SRAM available.

To analyse the benefits of the InRout algorithm we set different application requirements in terms of PER. The PER is a measure of the packets lost at the physical layer during transmissions and the packets dropped at the buffer due to the size limitations. The simulation analysis is performed under different channel conditions:

- **Good Channel Conditions** where the mean density of obstacles between two nodes ranges from 0.0 to 0.1 m-1 as per the physical layer model described in 5.1.1, this could be a factory where nodes are located with good line of sight and there are few obstacles among them. Note that the density of obstacles refers to how many obstacles such as walls, objects, people, etc., exist per square meter between two communicating nodes.

- **Variable Channel Conditions** with densities of obstacles ranging from 0.0 to 0.5 m-1, this could be a factory where some nodes are surrounded by obstacles whereas others have good line of sight.

- **Bad Channel Conditions** high obstacle densities in the range 0.4 to 0.5 m-1, this could be a factory with a high number of obstacles and poor line of sight visibility among nodes.

Furthermore, we analyse the consumed energy by the nodes for different load conditions to investigate the ability of InRout to balance the offered load and hence evenly distribute energy consumption. Finally, an analysis of the control
overhead introduced by InRout and the other algorithms under test is presented to examine the relative difference between these algorithms in terms of control overhead.

Table 5 shows the simulation parameters for our experimental analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sinks</td>
<td>2</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>50</td>
</tr>
<tr>
<td>Topology</td>
<td>Mesh</td>
</tr>
<tr>
<td>Nodes’ Position</td>
<td>Random/Static</td>
</tr>
<tr>
<td>Nodes’ Transmission Power</td>
<td>0 dBm</td>
</tr>
<tr>
<td>MAC layer protocol</td>
<td>IEEE 802.15.4 - DBOP [42]</td>
</tr>
<tr>
<td>MAC Layer Buffer Size</td>
<td>10 [packets]</td>
</tr>
<tr>
<td>Epsilon_max, Epsilon_min, step</td>
<td>0.9, 0.1, 0.1</td>
</tr>
</tbody>
</table>

Table 5: Simulation Parameters

5.1.1. Physical Layer Model

The default wireless physical layer model available in OPNET uses a free space propagation model. To create a more realistic indoor environment, we modified the physical layer model in [42] so that the channel of a wireless node would consider the pathloss power law and the fading due to the existence of obstacles in the radio path as per the model described in [45]. This model considers a similar approach to the well known Lutz model [46] in which transitions between line of sight (LOS) (good channel conditions) and non line of sight (NLOS) (bad channel conditions) occur in typical node locations due to obstacles in the radio path such as people, objects, etc. In good channel conditions fading is characterised by Ricean fading whereas a bad channel is represented by a combination of Rayleigh and log-normal distributions. The transitions between these two states are mainly controlled by the densities of obstacles.

The received power is used together with the noise power to obtain the signal to noise ratio (SNR). The obtained SNR is mapped onto the bit error rate (BER) using the look-up table. The model uses the BER curves for different fading channels. Thus, the BER curves in our model consider the fading effect of real indoor environments. In addition, the model obtains the number of errors by mapping a uniform random number in [0, 1] via the inverse of the cumulative mass function (CMF) for the bit error count distribution. The error correction threshold is set to zero since the IEEE 802.15.4 standard does not define any frame error correction techniques and WPAN receivers do not usually have bit error correction capabilities. Thus, a single error in a packet implies discarding in our simulated physical layer.

5.1.2. Battery Model

To evaluate the energy efficiency of the communication protocols, the simulation model used for our simulations [42] implements an estimation of the consumed and remaining energy levels in the sensor nodes. To be able to calculate the average power consumption of each node, the model estimates the instantaneous power consumption of the transceiver when operating in and switching between states. The power consumed by the sensors, microcontroller and other electronics is not considered as the radio protocols do not have any effect on them. The battery model is based on the well known TI CC2420 radio chip. The transceiver supports five states: off, power down, idle, receive and transmit. The energy utilized for switching from one mode to another are also considered by the model as they can impact the total energy consumption when operating at low duty cycles [47]. The total energy $E_c$ consumed by the radio chip is calculated with the following linear state model:

$$E_c = V_t \cdot \sum (I_m \cdot (t_m + t_t)) \quad \text{where} \quad m = tx, rx, id, sp$$

(14)

Where $V_t$ is the supply voltage, $I_m$ the current draw for the possible energy modes (transmit, receive, idle, sleep) and $t_t$ the transition time for switching to that energy mode.

In this work, to isolate the influence of the routing protocol on the energy consumption, we do not take into account the energy spent on sleeping or idle listening as that energy expenditure is directly related to the MAC protocol. We consider that duty cycling adaptation mechanisms are included at the MAC level [37] to adapt the superframe duration to the load to minimize idle listening. Overhearing is considered.
5.2. Test Cases

Based on the discussion presented in Section 3, routing protocols suitable for WSNs targeting industrial monitoring applications must at a minimum provide multiple paths and, for meaningful route selection, the protocol must have some mechanism for prioritising routes. To evaluate the proposed InRout algorithm we choose a selection of the most relevant topology and position based routing protocols that support multipath routing and route selection to be representative of state-of-the-art approaches towards WSN routing targeting industrial scenarios as outlined in Table 6. These protocols are broadly categorised based on their route discovery approach, the network topology supported, functionality (centralised or distributed), and their route selection mechanism.

In order to gauge the performance of the route selection algorithms under test it is necessary to benchmark their performance against a baseline reference algorithm. AOMDV has been selected as an appropriate reference algorithm as it is suitable as an approach for routing in industrial environments because it supports multipath route discovery with low overhead and has a simple route selection mechanism based on the shortest number of hops.

While InRout is proposed as a route selection algorithm it can be used in conjunction with any underlying route discovery protocol be it proactive or reactive, with InRout being distributed. To demonstrate this within the analysis, InRout is used in conjunction with AOMDV protocol (referred to as InRout_A) and with a Directed Acyclic Graph based routing protocol (labelled as InRout_DAG). Both graph routing and AOMDV are loop-free multipath protocols and can support multiple sinks. InRout_A is a distributed reactive routing protocol whereas InRout_DAG is a centralised proactive routing protocol and this allows the comparison of InRout over opposing route discovery algorithms. In addition, industrial based standards like WirelessHART and ISA100.11a promote the use of graph based routing with the IETF through ROLL [8] suggesting the use of directed acyclic graph routing for industrial scenarios. Consequently InRout_DAG is suitable as a network layer protocol for WirelessHART and ISA100.11a standards and as such it is used as a representative of the state of the art for standards based route selection. The directed acyclic graph for the network topology used in the simulation environment is shown in Figure 3.

To compare our work with existing approaches we choose route selection algorithms from routing protocols specifically developed for industrial scenarios and those that target similar types of performance metrics to InRout, that is, metrics of relevance to industrial wireless sensor networks. Among the position based protocols previously discussed in Section 3 [10–12, 32] the proactive positioned based EARQ protocol [12] is selected as being the most relevant for comparison. EARQ is chosen for comparison as it, like InRout has been developed for industrial environments. Next we consider the topology based HYDRO routing protocol being proposed by the IETF [20] for industrial scenarios where route selection is based on the expected number of transmissions (ETX) to perform a successful data packet delivery. Finally, we compare the route selection presented together with a modified version of AOMDV in [26] that relies on residual energy as a selection mechanism, where energy is one the most important factors that must be considered by battery operated wireless sensor nodes.

For the remainder of this paper we refer to the route selection mechanisms of the protocols under test as EARQ_RSM, HYDRO_RSM and MRE_RSM respectively.

The route selection mechanism proposed in EARQ, first considers the delay and then the energy together with the reliability, where reliability refers to successful delivery of packets. However, this work does not consider a node’s duty cycle in their delay calculations and this is a critical parameter that affects the end-to-end delay. We consider that due to the energy constraints of the sensor nodes duty cycling is a much needed requirement. Therefore, when evaluating EARQ route selection mechanism we omit the delay part as this does not consider duty cycling, leaving

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Route Discovery</th>
<th>Topology</th>
<th>Functionality</th>
<th>Route Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOMDV</td>
<td>Reactive</td>
<td>Mesh</td>
<td>Distributed</td>
<td>Shortest Path</td>
</tr>
<tr>
<td>InRout_A</td>
<td>Reactive</td>
<td>Mesh</td>
<td>Distributed</td>
<td>InRout</td>
</tr>
<tr>
<td>InRout_DAG</td>
<td>Proactive</td>
<td>DAG</td>
<td>Centralised + Distributed Route Selection</td>
<td>InRout</td>
</tr>
<tr>
<td>(WirelessHART and ISA100.11a)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MRE</td>
<td>Reactive</td>
<td>Mesh</td>
<td>Distributed</td>
<td>Energy</td>
</tr>
<tr>
<td>EARQ</td>
<td>Proactive</td>
<td>Tree</td>
<td>Distributed + Position based</td>
<td>Energy, Reliability</td>
</tr>
<tr>
<td>HYDRO</td>
<td>Reactive</td>
<td>Tree</td>
<td>Distributed</td>
<td>Expected number of transmissions</td>
</tr>
</tbody>
</table>

Table 6: Protocol Comparison
EARQ_RSM to consider energy and reliability requirements. With EARQ_RSM, when a node selects a route to send a packet, it first takes into account a probability that depends on the residual energy and distance between nodes. If after using that probability, the selected path does not match the reliability requirements, the node sends a second redundant packet through another neighbour. In addition the reliability metric calculation depends on the successful reception of packets over each link as well as on the probability related to the energy levels. The reliability does not consider packets lost at buffers.

5.3. MAC Layer Parameter Selection and Delay Analysis

We analyse the performance of InRout for different MAC layer parameters so appropriate settings can be established. The MAC parameters to be tuned are SO and BO and these define the active and sleep periods of the nodes respectively [40], see Equation 15. The aBaseSuperframeDuration constant represents the minimum length of the active period when BO is equal to 0. SD represents the active time and BI - SD represents the sleep time. BO and SO parameters must fulfil the following relationship: $0 \leq SO \leq BO \leq 14$. We use as MAC layer the IEEE 802.15.4 modification for mesh networks described in [42] and referred to as DBOP.

$$BI = aBaseSuperframeDuration \cdot 2^{BO}$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO}$$  \hspace{1cm} (15)

For mesh networks when using CSMA/CA transmissions and duty cycles (fraction of time that the node is awake) lower than 100%, the delay will be mainly dependent on the network load (application traffic) and the number of hops to the sink node [42]. For battery powered WSNs it is desirable to have low duty cycles in the region of 10% or less depending on expected application traffic rates. Also, due to the non-deterministic nature of CSMA/CA transmissions, an exact calculation of the delay cannot be provided nor can specific delay limitations be guaranteed.

To guarantee specific delay requirements the network parameters cannot be selected at random. To support this, we provide a delay analysis with regard to the MAC layer parameters so the best configuration of those can be selected.
Once the correct network settings have been established, InRout will choose the path with the lowest number of hops to deliver delay bounded packets.

We fix SO to 2 and we vary the BO parameter that defines the sleep time and periodicity of the beacon frame. We use a value of 2 as smaller values of SO show decreased network performance [48]. We select a low value for SO so for a specific duty cycle, beacon frames are sent more frequently. For different BO values and traffic settings, we analyse the PER and the delay obtained with InRout to decide the most appropriate BO, SO settings. We set an arbitrary application requirement to provide a PER of 5% or less for the PER tests. For the delay test, nodes preferentially use the paths with the lowest number of hops. As stated previously, tolerable delay requirements for industrial monitoring applications are in the order of seconds, i.e., they have soft delay requirements. To take into account the influence of the channel in this experiment we simulate variable link conditions across all links. Finally we analyse the performance for variable offered loads to evaluate the algorithm performances under different levels of stress. With this goal we use different packet rates with mean inter-arrival times ranging from 25s to 70s (following a Poisson distribution) with respective network offered loads (NOL) ranging from 2.438kbps to 0.871kbps (based on 60 nodes and a packet size of 127bytes) to mimic industrial monitoring application data generation rates. As it can be seen in Figure 4, the performance of the algorithm remains stable for lower values of BO (with BO values of 4, 5 duty cycles of 25% and 12.5% are achieved) in terms of PER with a fixed value of SO = 2. For BO values bigger or equal than 6 we obtain a duty cycle of 6% for BO = 6 and 3% for BO = 7 and we see that the performance decreases for higher offered loads as the network starts to become saturated. This is due to the fact that, for the same duty cycle, higher loads translate into more saturation of the superframe active time, i.e. the same amount of load has to be sent in a smaller period of time. This means more collisions, less chance to access the channel, higher buffer occupancies, etc. With regard to the delay, Figure 5 shows the delay for those nodes that are furthest from the sink, that is, the nodes that experience the highest delay in the network. Each point on this plot is the measure recorded at the 95th percentile on the cumulative distribution function, i.e. 95% of packets arrive with a delay below the value shown on the plot in Figure 5. As it can be seen, the BO values selected have a great influence on the total end-to-end delay. Delay here ranges between a couple milliseconds for lower BO values to several seconds for the higher BO values.

For the case where BO is equal to 7, Figure 5 shows two curves for the delay. One of the curves uses the hop count as metric and the other one uses the mean end-to-end delay as metric (sent as reward from node to node). As can be observed, using the end-to-end delay as metric does not show improvement when compared to using the hop count as metric. Therefore, for simplicity we use the hop count.

For the simulation analysis in the next section we use a value of 7 for the BO parameter based on a fixed value of SO = 2, i.e. a duty cycle of 3%. This allows us to see how the algorithm performs for challenging scenarios where the nodes sleep most of the time, which is indeed the most desired situation for energy saving purposes. The delays

![Figure 4: PER Analysis for Different BO Settings](image-url)
for the duty cycle selected remain in the order of seconds. This is sufficient for industrial monitoring applications, i.e., ISA classes 4 and 5, which have soft delay requirements in the order of seconds. Nevertheless, the important conclusion here, as shown in Figure 15, is that in order to assure certain delays, MAC layer parameters have to be tuned accordingly. Note that these settings are reliant on the network size and offered loads.

We use a static duty cycle here as the focus is on routing, however complementary duty cycle adaptation techniques could be used to adapt the active time to the offered load at each node.

5.4. Simulation Analysis

To analyse the performance of the proposed InRout algorithm with respect to the comparison algorithms in terms of satisfying the application PER requirements, we perform 3 rounds of simulations (with 2 seeds per round) for the 3 possible channel conditions, Good, Variable and Bad. Figures 5-7 show the successfully delivered packets to the two available sinks under the different channel conditions for several offered loads (mean packet inter-arrival times from 25s-70s). Industrial monitoring applications have packet success rate requirements ranging from 80% to 100% [17].

For each channel condition type we show the performance for two different PER requirements 5%, and 15% with InRout over AOMDV (InRout_A) and Graph Routing (InRout_DAG) and we compare the results to the AOMDV, HYDRO_RSM, EARQ_RSM and MRE_RSM.

Figure 6 shows the successfully delivered packets to both sinks under variable link conditions. In variable link conditions, the obstacle density experienced by each link randomly varies between $[0.0, 0.5]$ $m^{-1}$. As shown in Figure 6, InRout is able to find routes that satisfy the maximum PER condition of 15% at all times. For the stricter case of 5% PER the requirement can be satisfied until the offered load approaches high levels (1.742-2.438kbps). When the PER cannot be satisfied, the algorithm selects the best available route. The lower success rate is due to the fact that buffers cannot hold the entire offered load and also because a higher number of packet transmissions translates into more collisions due to the hidden terminal problem. This means that as long as the offered load does not surpass the network capacity, InRout is able to find routes that satisfy all the required packet error rate demands. What is more, the performance of InRout_A and InRout_DAG is very similar. InRout_DAG performance is marginally better due to the fact that all the possible routes are fed to the nodes by a central controller, whereas for InRout_A routes are found on demand and not all routes are stored. With regard to the test algorithms, AOMDV always selects the route with the lowest number of hops irrespective of link and buffer conditions. This translates into a poorer success rate when compared to the adaptive behaviour of the InRout algorithm. Paths with a low number of hops can often have long distances between the links and can have poor quality in terms of signal strength making it prone to errors. In addition repeatedly selecting the same shortest path will have the traffic load concentrated on the same set of links which can overload the buffers of those nodes. MRE_RSM and EARQ_RSM have quite a similar performance this
is mainly due to the fact that both consider the energy as the main selection mechanism. Since MRE\textunderscore RSM does not consider reliability, the algorithm performance, which is based only on residual energy levels, is poor with regard to the number of delivered packets. On the other hand, the EARQ\textunderscore RSM algorithm uses a reliability metric based on the successful transmissions performed at each link in addition to the energy metric. This reliability metric is used to analyse whether redundant packets have to be sent when the required reliability is not satisfied (set to 100\% for our simulations). However, since EARQ\textunderscore RSM does not define exploration mechanisms to keep this metric updated, the obtained values only reflect the initial status of the links. Because of the redundant packets sent by EARQ\textunderscore RSM when reliability requirements are not satisfied, the EARQ\textunderscore RSM performance worsens as sending redundant packets creates more load and the buffers are already full. Moreover, since the EARQ\textunderscore RSM reliability metric does not consider the packets lost at buffers due to overflow, EARQ\textunderscore RSM can keep choosing overloaded nodes which translates into more lost packets. This situation repeats as well for the rest of algorithms analysed here and all of the algorithms listed in the state-of-the-art review in section 3. These results highlight the need for buffer capacity consideration and, as InRout considers this, we see that this gives InRout a great advantage over the rest of algorithms as shown in Figures 5-7. In the figures we can see that where the performance of the other algorithms decreases with the load dramatically, InRout is able to maintain a good steady performance until the offered load surpasses the network capacity. Finally, HYDRO\textunderscore RSM, which is the other only algorithm (apart from InRout) that considers reliability as main performance metric, is the one that shows the best performance among all the other algorithms used for as comparison cases in terms of successfully delivered packets. However, because it does not consider packets lost at the buffer, the performance decreases severely with the offered load. Moreover, HYDRO\textunderscore RSM does not define how the routes should be explored in order to keep reliability metrics updated, therefore the metric is based on the initial values gathered using a round robin strategy. This shows a poorer performance than InRout as it explores the routes using the exploration/exploitation strategy defined previously to continuously update route metrics.

Figure 7 shows the successfully delivered packets under good link conditions. In good link conditions, the obstacle density experienced by each link randomly varies between \([0.0, 0.1] m^{-1}\). As it can be observed, when the conditions are good, the InRout algorithm satisfies all PER requirements. This is due to the fact that since the link quality of more routes is good, the packets can be sent through more of those routes hence distributing the load and making the buffers less congested overall which allows the PER requirements to be satisfied. The comparison algorithms performance is also better in this case but as before, the performance decreases with increasing load as none of the algorithms consider buffer limitations. For lower offered loads, the performance of the comparison cases improves as the collisions are reduced and the buffers are less congested. When the required PER for InRout is 15\%, the number of delivered packets is lower compared to the comparison algorithms in some cases. This is because for that requirement, the nodes trade-off delivered packets (while still maintaining the required PER) for energy balancing.
that is, packets are sent through poorer quality routes with the goal of improving the global network lifespan with the PER requirement still being satisfied.

Finally, Figure 8 shows the successfully delivered packets under bad link conditions. In bad link conditions, the obstacle density experienced by each link randomly varies between \([0.4, 0.5]\) \(m^{-1}\). As it can be seen, under bad conditions and high load, InRout is unable to satisfy stricter PER requirements, because none of the available routes can guarantee these PER limits. This however is expected as the routing algorithm cannot counteract the bad channel conditions but only react to them by choosing the best available routes as a consequence of those channel conditions. As the load on the networks decreases the PER requirements are satisfied. This is due to the fact that buffers are not overloaded even though there are retransmissions caused by the bad channel conditions. In this case, InRout_A shows poorer performance than InRout_DAG. This is due to the fact that AOMDV, the underlying protocol upon which
InRout_A runs, finds fewer routes in the harsh network conditions as more route request and route reply packets are lost. For this scenario, the reference cases exhibit the poorest performance as well.

We can conclude from the PER analysis that InRout is able to satisfy application PER requirements, given fixed buffer limitations and the unpredictability of the wireless channel, in the majority of the scenarios tested. As long as the offered load does not surpass the network capacity, InRout is able to satisfy the application PER requirements. As shown by previous analysis, InRout_DAG and InRout_A have very similar performances, and it is only when the channel conditions are bad that InRout_A performs significantly worse than InRout_DAG due to the fact that InRout_A relies on AOMDV to learn multiple routes on demand, and for that harsh case, less route request and route reply packets are able to reach their destinations. Nevertheless, the analysis shows that InRout can be used successfully over different underlying protocols. But independently of the underlying protocol, InRout performance surpasses the comparison algorithms performance in all cases. The main reason for this improvement is that InRout considers the buffer limitations and explores the routes with the exploration/exploitation of the Q-learning strategy to keep the algorithm metrics updated. This also implies that InRout is more energy efficient as it can deliver good performance even if the duty cycle is low (the other algorithms would need longer duty cycles to avoid overloaded buffers in order to achieve a better performances).

When compared to the reference algorithms in terms of gain, InRout demonstrates how the Q-learning process combined with the selected metrics and the exploration/exploitation strategy can make a difference in terms of successfully delivered packets with gains ranging from 4% (Good channel, HYDRO_RSM, 0.87kbps) to 60% (Good channel, AOMDV, 2.43kbps).

With regard to the energy analysis, one of the goals of the InRout algorithm is to balance the energy consumption when the PER requirements of the application allow it. This is done to avoid network partitioning. A network may become partitioned once a node dies and some nodes or groups of nodes can not reach the sinks, for example because a single node is chosen all the time as router by all its neighbours. Therefore by balancing the energy consumption, the time to network partitioning can be prolonged. Table 7 shows the mean energy consumed by the nodes in the network, the standard deviation and the relative standard deviation for one hour of simulation for the highest and lowest network offered load for all the algorithms under test.

To calculate the influence of the routing algorithm on the energy expenditure, we just consider the energy spent by the nodes on transmitting and receiving frames. We do not take into account the energy spent on sleeping or idle listening as that energy expenditure is directly related to the MAC protocol. For instance, duty cycling adaptation mechanisms can be included at the MAC level [37] to adapt the superframe duration to the load to minimize idle listening.

As shown in Table 7, as the energy consumption is estimated based on the number of transmissions and receptions, those algorithms that have a higher success rate also have higher energy consumption. In order to analyse the energy balancing properties of the algorithms, our interest focuses on the Relative Standard Deviation (RSD) of the energy consumed by the nodes. A higher RSD means a higher difference in the energies consumed by the nodes. As it can be observed, AOMDV and HYDRO_RSM have the highest RSD. This is because AOMDV always chooses the route with the lowest number of hops and HYDRO_RSM always selects the best link. Therefore, the nodes along this route consume more energy and so they will die sooner, leading to network partitioning. EARQ_RSM has the next highest RSD. This is due to the fact that EARQ_RSM sends redundant packets (i.e. it sends the same packet through more than one route to increase the chances of successful delivery) if the selected route using the energy metric provides bad link quality. Therefore the nodes along these bad quality links are more likely to die sooner. InRout_DAG and InRout_A with the 5% requirement on PER have the next best RSDs values. As the 5% PER requirement is quite strict, fewer routes are available to satisfy that requirement, however, due to the fact that the buffer load is considered in the PER estimation, there is some load balancing unlike the previous algorithms. If the PER requirement is less strict, i.e. 15%, InRout has more room for balancing the energy consumption and hence the RSD is lower. Finally, MRE_RSM, which only considers the residual energy as metric, is the algorithm that provides the lowest RSD values but these are close to that of InRout with a 15% PER requirement. While MRE_RSM performs better in terms of energy it does so at the expense of packet throughput as energy is the only metric it considers and as we have demonstrated this is not sufficient when attempting to satisfy QoS demands (as shown in figures 5-7).

We can conclude therefore that with InRout more energy balancing can be achieved when the PER requirement is relaxed. This is because as long as the PER requirement is satisfied, the route selection is done with regard to the energy. Therefore, when using InRout, there has to be a trade-off between PER and energy balancing requirements.
Table 7: Energy Expenditure Analysis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>InRoutDAG PER 5%</td>
<td>8.23</td>
<td>3.37</td>
<td>0.66</td>
</tr>
<tr>
<td>InRoutDAG PER 15%</td>
<td>7.97</td>
<td>3.10</td>
<td>0.33</td>
</tr>
<tr>
<td>InRoutA PER 5%</td>
<td>8.10</td>
<td>3.14</td>
<td>0.62</td>
</tr>
<tr>
<td>InRoutA PER 15%</td>
<td>7.85</td>
<td>3.03</td>
<td>0.34</td>
</tr>
<tr>
<td>HYDRO RSM</td>
<td>5.69</td>
<td>3.11</td>
<td>0.34</td>
</tr>
<tr>
<td>MRE RSM</td>
<td>5.22</td>
<td>2.83</td>
<td>0.21</td>
</tr>
<tr>
<td>EARQ RSM</td>
<td>5.56</td>
<td>2.85</td>
<td>0.55</td>
</tr>
<tr>
<td>AOMDV</td>
<td>3.14</td>
<td>2.65</td>
<td>0.77</td>
</tr>
</tbody>
</table>

Table 8: Control Bits and Control Energy

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Control Bits</th>
<th>Energy [J]</th>
</tr>
</thead>
<tbody>
<tr>
<td>InRoutDAG PER 5%</td>
<td>975</td>
<td>0.26</td>
</tr>
<tr>
<td>InRoutDAG PER 15%</td>
<td>944</td>
<td>0.26</td>
</tr>
<tr>
<td>InRoutA PER 5%</td>
<td>955</td>
<td>0.26</td>
</tr>
<tr>
<td>InRoutA PER 15%</td>
<td>930</td>
<td>0.25</td>
</tr>
<tr>
<td>HYDRO RSM</td>
<td>1349</td>
<td>0.86</td>
</tr>
<tr>
<td>MRE RSM</td>
<td>1317</td>
<td>0.33</td>
</tr>
<tr>
<td>EARQ RSM</td>
<td>10102</td>
<td>2.85</td>
</tr>
<tr>
<td>AOMDV</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

InRout performs better than AOMDV, HYDRO_RSM and EARQ_RSM in terms of energy balancing, as it has lower RSD values. Because AOMDV_RSM and HYDRO_RSM do not consider the energy of the nodes, network partitioning situations will happen sooner. On the other hand, although EARQ_RSM considers the energy as metric, because it sends redundant packets, the energy balancing is poorer than for example with MRE_RSM which is the algorithm with the best performance in terms of energy balancing, i.e. the lowest RSD.

By equalising the energy consumption of the network InRout provides a better network performance for a longer period of time.

To conclude, Table 8 shows the mean control overhead per node and the mean energy consumed in sending these control bits for each test algorithm. These measures are captured for the highest offered load case with variable link conditions. For a fair comparison, we embedded the control information in the beacon payload for all cases. The cost is updated from the sink to the end nodes (source node that generated the data frame) whenever the sink receives a data frame.

As it can be observed, except for the AOMDV case, which only uses the hop count as cost, InRout is the most efficient algorithm in terms of control overhead. In addition, InRout saves more energy as it requires a smaller number of control bits. The control overhead is low for InRout as a consequence of minimising the number of bits needed for advertising the Q-value, 8 bits in the case of InRout (see Table 4). This makes the InRout design more suitable than the other test algorithms as a solution for energy constrained wireless sensor networks.

6. Conclusion

While standards bodies are actively pursuing standardization in the area of low power wireless communications networks, there has been only broad based references to routing protocols. WirelessHART and ISA100.11a propose the use of multipath graph routing in a centralized manner with the specifics of the implementation being left open. Also, no recommendation is provided on how to make routing adaptable to the different network conditions and industrial application needs, which is a basic requirement for industrial scenarios. HYDRO, a multipath routing protocol proposed by the IETF ROLL group provides a basic protocol design but again leaves the development of more comprehensive routing solutions open. While there are a number of multipath routing protocols available, they do not adequately address the needs of WSNs targeting monitoring applications in industrial environments. Rather than propose yet another topology based routing protocol we view routing as being two fold and look on it as being composed of route discovery and route selection. While many routing protocols address efficient route discovery, be it on-demand or centralized, route selection has not been adequately addressed for WSNs with QoS based application demands as required in industrial monitoring applications.. Consequently we proposed the InRout Route Selection algorithm, an adaptive multi-metric based route selection algorithm that uses Q-learning to choose the best routes based on current network conditions and application settings, which can sit on top of any underlying multipath routing
protocol. InRout considers the inherent restrictions and challenges imposed by WSNs with route differentiation being driven by the goal of satisfying industrial application needs like required PER, delay or energy with a low cost in memory, while maintaining low control overhead. In our analysis we have shown how performance degradations manifest when buffer capacities are not considered with InRout being the only algorithm among our test cases to consider this and consequently it exhibited better performance. InRout has been shown to exhibit better performance over a range of test cases. In terms of energy balancing InRout has achieved comparable energy expenditure with the MRE protocol which considers only energy. The fine tuning of the MAC layer parameters allows InRout to satisfy the soft delay requirements of industrial monitoring applications. In addition we have shown that InRout is a possible solution for a network layer protocol for Wireless HART and ISA100.11a and it can be adapted for use with the HYDRO protocol being proposed by the IETF ROLL Group. As a future extension to this work we will focus on applications with tighter latency and higher bandwidth requirements and will investigate how to combine InRout with bandwidth allocation. This integration will necessitate a cross-layer approach where the network and MAC layer will work together in building routes with end-to-end bandwidth reservation allowing the estimate of end-to-end delays.

7. Acknowledgements

This work is supported by the Irish Government under the "Network Embedded Systems” project (NEMBES) in the Higher Education Authority’s Programme for Research in Third Level Institutions (PRTLI) cycle 4.

References


[40] IEEE 802.15.4 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2006.


Berta Carballido Villaverde is a postgraduate researcher in the Nimbus Centre for Embedded Systems Research at Cork Institute of Technology, Cork, Ireland. In 2006 she received a scholarship from Nokia Siemens Networks in order carry out her Master Thesis research in the NSN research centre at Aalborg University, Denmark. In September 2007 she was awarded a Masters degree in Telecommunications by Miguel Hernandez University, Elche, Spain. Her current research interests focus on the development of algorithms and protocols for QoS provision in wireless sensor networks.

Dr. Susan Rea is a researcher in the Nimbus Centre at Cork Institute of Technology and was awarded her PhD in Electronic Engineering in October 2006. Her current research interests focus on the development of applications, algorithms and protocols for wireless ad hoc networks, and middleware management for self configuration in embedded wireless sensor networks and their application to monitoring and control for smart environments. Her work has concentrated on the development of efficient wireless embedded communication protocols and middleware for wireless sensor network reconfiguration and management.

Dr. Dirk Pesch is Director of the Nimbus Centre for Embedded Systems Research at Cork Institute of Technology. Prior to joining CIT in 1999, he was a research fellow at Strathclyde University, Glasgow, Scotland and with Nokia in Germany and the UK. Dirk received degrees from RWTH Aachen University, Germany and Strathclyde University, both in Electrical & Electronic Engineering. Dirk’s research focuses on analysis/design of algorithms and protocols for heterogeneous wireless/mobile networks, methods/tools for WLAN design, and development of wireless network and service management. Dirk has over 100 publications in his area of expertise and is active in international conference organisation.