UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GUSTAVO KÜNZEL

APPLICATION OF REINFORCEMENT LEARNING WITH Q-LEARNING FOR THE ROUTING IN INDUSTRIAL WIRELESS SENSORS NETWORKS

Porto Alegre
2021

GUSTAVO KÜNZEL

APPLICATION OF REINFORCEMENT LEARNING WITH Q-LEARNING FOR THE ROUTING IN INDUSTRIAL WIRELESS SENSORS NETWORKS

Thesis presented to Programa de Pós-Graduação em Engenharia Elétrica of Universidade Federal do Rio Grande do Sul as part of the requirements for obtaining the degree of Doctor in Electrical Engineering.
Concentration field: Automation Systems

Supervisor: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre

2021

GUSTAVO KÜNZEL

APPLICATION OF REINFORCEMENT LEARNING WITH Q-LEARNING FOR THE ROUTING IN INDUSTRIAL WIRELESS SENSORS NETWORKS

This thesis was considered adequate to obtain the degree of Doctor in Electrical Engineering and approved in its final form final by the Supervisor and the Examination Committee.

Supervisor: _____
Prof. Dr. Carlos Eduardo Pereira
Doctor by Universität Stuttgart –Stuttgart, Germany

Examination Committee:

Prof. Dr. Leandro Buss Becker, UFSC
Doctor by Universidade Federal do Rio Grande do Sul –Porto Alegre, Brazil

Prof. Dr. Dennis Brandão, USP
Doctor by Universidade de São Paulo –São Paulo, Brazil

Prof. Dr. Edison Pignaton de Freitas, UFRGS
Doctor by Halmstad University –Halmstad, Sweden

Coordinator of PPGEE: _____
Prof. Dr. Sérgio Luís Haffner

Porto Alegre
February 2021

ACKNOWLEDGEMENTS

To Professor Carlos Eduardo Pereira, for providing the opportunity to work in this research field, for the advice and trust in my work.

To Professor Leandro Soares Indrusiak for the advice, guidance, and opportunity to work as a researcher at the University of York.

To the post-graduate colleagues Jean Michel Winter, Gustavo Pedroso Cainelli, Max Feldman, Dylan Timm, Felipe Tondo, Luis Felipe Zeni, Christian Alan Krötz, Alexandre dos Santos Roque, Yuri das Neves Valadão, Márcio José da Silva, Juliano de Lima, Fernando Sacilotto Crivellaro, and others, for the support, exchanges, and researches developed together.

To the Professors Ivan Müller, João César Netto, Alexandre Balbinot, Valner João Brusamarello, Edison Pignaton de Freitas, Leandro Buss Becker, Dênis Brandão and others, for the advice, exchange of ideas and researches conducted together.

To the administrative technicians and members of the Post-Graduate Program in Electrical Engineering of UFRGS, especially to Miriam Rosek, for the assistance and support.

To the Federal Institute of Science, Technology and Education of Rio Grande do Sul (IFRS) and my fellow colleagues from Campus Farroupilha for providing the opportunity, the time, and the resources needed to accomplish this work.

To the University of York, YARCC and the IT Services crowd, for providing the computational resources required for the simulations conducted in this research.

Finally, to my friends and parents, for the unconditional support and help throughout this journey.

ABSTRACT

Industrial Wireless Sensor Networks (IWSN) usually have a centralized management approach, where a device known as Network Manager is responsible for the overall configuration, definition of routes, and allocation of communication resources. The routing algorithms need to ensure path redundancy while reducing latency, power consumption, and resource usage. Graph routing algorithms are used to address these requirements. The dynamicity of wireless networks has been a challenge for tuning and developing routing algorithms, and Machine Learning models such as Reinforcement Learning have been applied in a promising way in Wireless Sensor Networks to select, adapt and optimize routes. The basic concept of Reinforcement Learning is the existence of a learning agent that acts and changes the state of the environment, and receives rewards. However, the existing approaches do not meet some of the requirements of the IWSN standards. In this context, this thesis proposes the Q-Learning Reliable Routing approach, where the Q-Learning model is used to build graph routes. Two approaches are presented: QLRR-WA and QLRR-MA. QLRR-WA uses a learning agent that adjusts the weights of the cost equation of a state-of-the-art routing algorithm to reduce the latency and increase the network lifetime. QLRR-MA uses several learning agents so nodes can choose connections in the graph trying to reduce the latency. Other contributions of this thesis are the performance comparison of the state-of-the-art graph-routing algorithms and the evaluation methodology proposed. The QLRR algorithms were evaluated in a WirelessHART simulator, considering industrial monitoring applications with random topologies. The performance was analyzed considering the average network latency, network lifetime, packet delivery ratio and the reliability of the graphs. The results showed that, when compared to the state of the art, QLRR-WA reduced the average network latency and improved the lifetime while keeping high reliability, while QLRR-MA reduced latency and increased packet delivery ratio with a reduction in the network lifetime. These results indicate that Reinforcement Learning may be helpful to optimize and improve network performance.

Keywords: Industrial Wireless Sensor Networks. Routing. Reinforcement Learning. Q-Learning.

RESUMO

As Redes Industriais de Sensores Sem Fio (IWSN) geralmente têm uma abordagem de gerenciamento centralizado, onde um dispositivo conhecido como Gerenciador de Rede é responsável pela configuração geral, definição de rotas e alocação de recursos de comunicação. Os algoritmos de roteamento precisam garantir a redundância de caminhos para as mensagens, e também reduzir a latência, o consumo de energia e o uso de recursos. O roteamento por grafos é usado para alcançar estes requisitos. A dinamicidade das redes sem fio tem sido um desafio para o ajuste e o desenvolvimento de algoritmos de roteamento, e modelos de Aprendizado de Máquina como o Aprendizado por Reforço têm sido aplicados de maneira promissora nas Redes de Sensores Sem Fio para selecionar, adaptar e otimizar rotas. O conceito básico do Aprendizado por Reforço envolve a existência de um agente de aprendizado que atua em um ambiente, altera o estado do ambiente e recebe recompensas. No entanto, as abordagens existentes não atendem a alguns dos requisitos dos padrões das IWSN. Nesse contexto, esta tese propõe a abordagem *Q-Learning Reliable Routing*, onde o modelo Q-Learning é usado para construir os grafos de roteamento. Duas abordagens são propostas: QLRR-WA e QLRR-MA. A abordagem QLRR-WA utiliza um agente de aprendizado que ajusta os pesos da equação de custo de um algoritmo de roteamento de estado da arte, com o objetivo de reduzir a latência e aumentar a vida útil da rede. A abordagem QLRR-MA utiliza diversos agente de aprendizado de forma que cada dispositivo na rede pode escolher suas conexões tentando reduzir a latência. Outras contribuições desta tese são a comparação de desempenho das abordagens com os algoritmos de roteamento de estado da arte e a metodologia de avaliação proposta. As abordagens do QLRR foram avaliadas com um simulador WirelessHART, considerando aplicações de monitoramento industrial com diversas topologias. O desempenho foi analisado considerando a latência média da rede, o tempo de vida esperado da rede, a taxa de entrega de pacotes e a confiabilidade dos grafos. Os resultados mostraram que, quando comparado com o estado da arte, o QLRR-WA reduziu a latência média da rede e melhorou o tempo de vida esperado, mantendo alta confiabilidade, enquanto o QLRR-MA reduziu a latência e aumentou a taxa de entrega de pacotes, ao custo de uma redução no tempo de vida esperado da rede. Esses resultados indicam que o Aprendizado por Reforço pode ser útil para otimizar e melhorar o desempenho destas redes.

Palavras-chave: Redes de Sensores Industriais sem Fio. Roteamento. Aprendizado por Reforço. Q-Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACK	Acknowledge
AI	Artificial Intelligence
ANL	Average Network Latency
ANOVA	Analysis of Variance
AP	Access Point
ASN	Absolute Slot Number
BFS	Breadth-First Tree
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CCA	Clear Channel Assessment
DLPDU	Data-Link Protocol Data Unit
EBGR	Energy-Balancing Graph Routing
ELHFR	Exhenced Least-Hop First Routing
ENL	Expected Network Lifetime
FLO-RG	Frame Level Optimized Reliable Graph
ID	Identifier
I4.0	Industry 4.0
IoT	Internet of Things
IIoT	Industrial Internet of Things
IWSN	Industrial Wireless Sensor Network
ISM	Industrial, Scientific, Medical
JRMLR	Joint Routing Algorithm for Maximizing Network Lifetime
KA	Keep-Alive
LLC	Logical Link Control
MAC	Medium Access Control
MANET	Mobile Ad-Hoc Network
MARL	Multi-agent Reinforcement Learning
MDP	Markov Decision Process

ML	Machine Learning
MPAR	Multipath Routing Protocol
NM	Network Manager
NS-2	Network Simulator 2
NS-3	Network Simulator 3
OTCL	Object-Oriented Tool Command Language
PDR	Packet Delivery Rate
POMDP	Partial Observable Markov Decision Process
PRN	Percentage of Reliable Nodes
PPGEE	Post-Graduate Program in Electrical Engineering
QoS	Quality of Service
QDAR	Q-Learning Based Delay-Aware Routing
QLRR	Q-Learning Reliable Routing
QLRR-WA	Q-Learning Reliable Routing with a Weighting Agent
QLRR-MA	Q-Learning Reliable Routing with Multriple Agents
RL	Reinforcement Learning
RSL	Received Signal Level
SM	Security Manager
TDMA	Time Division Multiple Access
UFRGS	Federal University of Rio Grande do Sul
VoQL	Voting Q-Learning
WH	WirelessHART
WMN	Wireless Mesh Network
WSAN	Wireless Sensor and Actuador Network
WSN	Wireless Sensor Network
XML	Extensible Markup Language

LIST OF SYMBOLS

	Symbols related to Q-Learning
α	learning rate
γ	discount factor
a	action
A	set of actions
s	state
S	set of states
Q	Q-table
r	reward
R	reward function, reward value
P	probability matrix
π	policy
ε	exploration probability
t	time instance, instant
i	agent, agent of a node
	Symbols related to graphs
G	graph
g	gateway
V	set of vertexes, or set of nodes
v	node
u	parent node, successor node
E	set of edges
$e_{v,u}$	edge with origin v and destination u
	Symbols related to the QLRR approaches
V_{AP}	set of Access Points
G_U	uplink graph
V_U	uplink graph vertexes
E_U	uplink graph edges

t_s	time interval
d_{t+1}	current average network latency
d^v_{t+1}	current average latency of node v
l_{t+1}	current expected network lifetime
D	array used to store the last measurements of the average network latency
L	array used to store the last measurements of the expected network lifetime
k	number of measurements stored in array D or L
U_v	set of successors of node v
E_v	set of edges with the successors of node v
S'	set of candidate nodes
S''	set of candidate nodes
h_v	average number of hops from node v to g
h_{max}	maximum number of hops in a set of candidate nodes
n_v	number of outgoing edges from v
n_{max}	maximum number of outgoing edges in a set of candidate nodes
c	cost of a candidate node
s	received signal level of and edge
s_d	desirable signal level for and edge
w_x	weight value of parameter x
p	energy restriction value
N_w	set of weights values
M	number of transitions
W_f	weight factor
	Symbols related to the scheduler
F	superframe F
l_F	length of superframe F in slots
t_{ADV}	period for advertisement
n_{ADV}	number of advertisement slots
t_{DP}	period for permanent downlink management slots
n_{DP}	number of permanent downlink management slots
t_{DN}	period for normal downlink slots

n_{DN}	number of normal downlink slots
t_{UP}	period for uplink management slots
n_{UP}	number of uplink management slots
t_{UN}	period of normal uplink slots
t_{UN}	number of normal uplink slots
	Math symbols
$\exists$	exists
$\in$	belongs
$\forall$	for all
:	such that
$\|X\|$	cardinality, size of set X
$\subseteq$	subset
$\cup$	union

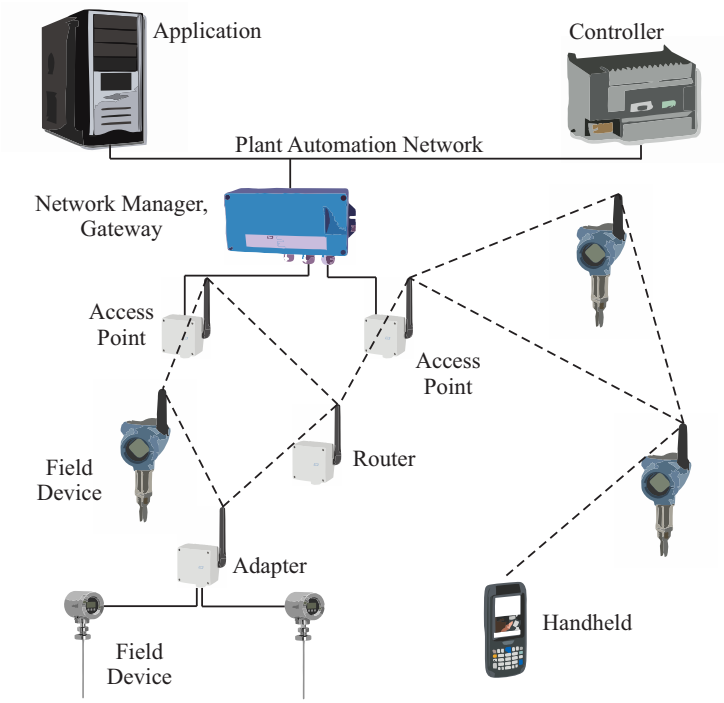
CONTENTS

1 INTRODUCTION

Industrial Wireless Sensor Networks (IWSN) are an attractive technology for communications in process automation and allow the incorporation of Internet of Things (IoT), Industrial Internet of Things (IIoT) and Industry 4.0 (I4.0) concepts (SHA et al., 2017; XU; HE; LI, 2014). Flexibility, mobility, expansion, ease of maintenance and reduced wired infrastructure are the main advantages of IWSN (WINTER et al., 2014). The global IWSN market size is anticipated to reach USD 8.67 billion by 2025, and it is expected a reduction of infrastructure costs between 50 % to 90 % when compared to wired solutions (WANG; CHAI; WONG, 2016; GVR, 2018).

IWSN consist of a set of wireless field devices (nodes) connected to a gateway through Access Points (AP). The gateway provides a connection with the automation network. A device known as Network Manager (NM) is connected to the gateway and is responsible for the management of the network, admission control, configuration, routing, and scheduling. Auxiliary devices such as routers, adapters and handhelds are used to increase the network range, connect wired devices to the wireless network, and configure field devices, respectively. Figure 1 depicts an IWSN topology.

Figure 1 – An IWSN example.



Source – Adapted from Künzel (2012).

IWSN applications often require reliable, low-latency, and real-time communications. Low energy consumption is another requirement as batteries are often used to power devices

(SHA et al., 2017). Meeting these requirements while optimizing the network performance is often complex because of the characteristics of the devices, topologies, and the wireless network properties (shared medium, interference, signal reflections, and signal strength) (SHEN et al., 2014; IKRAM et al., 2014; NIU et al., 2014).

Standards such as WirelessHART (WH), ISA SP100.11a and WIA-PA were introduced in the last decade for process monitoring and control, and have been used in IWSN applications (WINTER et al., 2015). They are based on the IEEE 802.15.4 standard, suitable for applications with battery-powered devices (SHA et al., 2017). These standards usually form a mesh network, where nodes may act as routers to increase path availability for communications (NOBRE; SILVA; GUEDES, 2015b). Centralized management is used to better control the network operation and to simplify the hardware and software of the nodes (CHEN; NIXON; MOK, 2010).

Routing is an essential task of the NM. The routes built by the NM are used by the devices to send data through the network. The paths used for sending data must be carefully chosen to ensure the desired network performance and meet the requirements of IWSN applications (NOBRE; SILVA; GUEDES, 2015b). To increase the reliability of the communications, path redundancy is used to build routes and is implemented through graph routing. A graph is a route that connects nodes on the network, and each intermediate node on a route to the destination may have multiple neighbors to forward a message to. If the communication with a neighbor fails, a node can try to send the message through another neighbor (SHA et al., 2017; HAN et al., 2011).

Graph routing algorithms for centralized management protocols were described over the last decade in Jindong, Zhenjun and Yaopei (2009), Han et al. (2011), Künzel (2012), Zhang, Yan and Ma (2013), Memon and Hong (2013), Zhang et al. (2014), Wu et al. (2015), Wu et al. (2016), Sepulcre, Gozalvez and Coll-Perales (2016), Künzel, Cainelli and Pereira (2017), Künzel et al. (2018), Han, Ma and Chen (2019), Madduma-Bandarage (2020). These algorithms try to increase reliability through path redundancy while reducing latency, energy consumption, transmission errors and resource usage. Parameters, heuristics, and weighted cost equations are used to choose the connections in the graphs. Usually, the parameters and weights are statically defined and suitable only for certain network operation conditions (NOBRE; SILVA; GUEDES, 2015b).

1.1 MOTIVATION

It is inconvenient to manually adjust the parameters of the routing algorithms, aiming to improve the network performance. This task requires several tests, the periodic monitoring of the network status, the user must know about the properties of the algorithms, and a network system representation is often unavailable for tests (KÜNZEL et al., 2018). These adjustments could be made in a way that achieves an adaptation according to the current network operational conditions while balancing or optimizing some performance metrics. Centralized routing algorithms that

can optimize the performance of the IWSN are a relevant research topic and have not been widely explored in the literature (NOBRE; SILVA; GUEDES, 2015b).

The use of Machine Learning (ML) for creating and adjusting routes may be useful for IWSN and future IoT, IIoT, and I4.0 protocols (XU; HE; LI, 2014; SAVAGLIO et al., 2019). Machine Learning (ML) provides a system with the ability to learn and improve from experience, and Reinforcement Learning (RL) relies on the existence of an agent that acts in an environment and receives rewards based on the results of its actions. By exploring the environment, it learns which behavior it must take to maximize its rewards (SUTTON; BARTO, 2018). RL demands low computational resources and implementation efforts, thus providing high flexibility to topological changes and near-optimal results, without requiring any *apriori* network model (SAVAGLIO et al., 2019).

RL algorithms like Q-Learning have been used in centralized and decentralized routing approaches in general-use network technologies and also in Wireless Sensor Networks (WSN), as presented in the works and surveys of Al-Rawi, Ng and Yau (2015), Habib, Arafat and Moh (2019), and Mammeri (2019). In decentralized approaches, each node is modeled as a learning agent that selects routes to forward its packets. The decentralized approaches are not suitable for the current IWSN protocols since they require nodes to exchange information independently, reconfigure, and decide its routing strategies (KÜNZEL et al., 2018). It would be necessary to change the current IWSN protocol stacks to use these approaches. Similarly, the available centralized approaches are not suitable for IWSN since they are intended to be used with other protocols and do not build graphs or routes with path redundancy.

The use of Q-Learning for creating graphs and routes in a centralized fashion may be useful for IWSN protocols and future wireless IoT, IIoT and I4.0 protocols (KÜNZEL et al., 2018). Routing algorithms that can optimize the performance of IWSN using RL techniques, adapting to changes in the operational conditions, are a relevant research topic not widely explored in the literature. The growth of the IWSN market and the use of IoT, IIoT, and I4.0 motivates the research in this field because these technologies still have aspects to improve to become attractive. Besides, the state-of-the-art graph routing algorithms are evaluated using different scenarios and protocols. The parameters and operational conditions vary (the quantity and characteristics of the devices used, spatial distribution, topologies, signal propagation and error models, metrics, and methodologies used for comparing results, among others). In general, the simulation tools used for comparison do not have a complete stack implementation of a IWSN protocol. In this sense, it is also relevant to identify the available simulators for a given protocol and to compare and analyze the performance of the graph-routing algorithms considering the same operational conditions.

1.2 OBJECTIVES

In this context, this thesis has as general objective to propose the application and performance evaluation of RL techniques such as Q-Learning for routing in centralized-management IWSN standards.

The following specific objectives are necessary to achieve this general objective:

- To analyse and classify the state-of-the-art, identifying contributions in the fields related to this thesis;
- To propose centralized approaches to build routing graphs using Q-Learning;
- To discuss the use of RL for routing in IWSN protocols;
- To propose procedures, scenarios, and metrics to evaluate the performance of graph-routing algorithms;
- To compare the performance of the new approaches against the state-of-the-art routing algorithms;
- To identify future research possibilities.

1.3 CONTRIBUTIONS

The main contributions of the thesis are:

- The presentation, classification, and analysis of the state of the art;
- The routing algorithms proposed for the creation of graphs using RL;
- The performance comparison of state-of-the-art routing algorithms;
- The procedure and scenarios used for performance comparison;
- The improvements in the simulation environment;
- The discussion of the use of RL for routing in IWSN;
- The discussion of future works.

1.4 THESIS STRUCTURE

The thesis is structured as follows. Chapter 2 presents the main theoretical concepts such as Graph Theory, Wireless Mesh Networks (WMN), Reinforcement Learning, Q-Learning, WirelessHART and WirelessHART simulators. Chapter 3 analyzes the state of the art. Chapter 4 presents two new routing algorithms using Q-Learning. Chapter 5 presents the performance evaluation and discusses the results. Finally, chapter 6 presents the conclusions and future works.

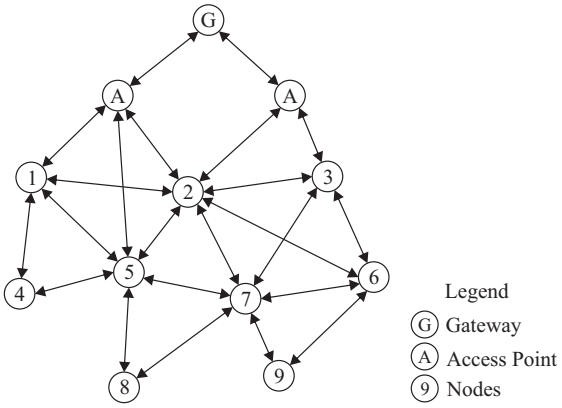
2 GRAPHS, WMN, RL, WIRELESSHART AND SIMULATORS

This chapter presents the theoretical concepts used throughout the thesis. A definition of graphs and WMN is presented, followed by the concepts of RL, Q-Learning, and the relevant details of the WirelessHART protocol and WirelessHART simulators.

2.1 GRAPHS

The graph model is used for the representation of communication networks since it allows a natural and intuitive mapping of these. Formally, a graph $G = (V, E)$ is a structure composed by vertices V and edges E. The vertices represent devices in the network (also known as nodes in wireless networks), whereas edges represent the connections between the devices. An edge between two devices exists only if they can communicate with each other. The definitions presented here are the same used in Künzel (2012) and Han et al. (2011). Figure 2 depicts the topology of a network through a graph. This topology will be further used along for exemplification purposes.

Figure 2 – Topology of a network represented by graphs.



Source – Künzel (2012)

2.1.1 Graph classification

The following graph classification is used in this work:

a) By orientation: Graphs can be classified as directed or non-directed. They are directed when edges have a direction associated with them. The direction of an edge is represented by an arrow indicating the flow of data;

b) By value: In valued graphs, vertices and edges may have values which represent costs associated with some characteristics.

2.1.2 Graph concepts

The following concepts are used in this work:

a) Neighbors: two vertices are neighbors if they have an edge that connects them in the graph;

b) Incident edge or vertex: an edge or vertex is incident to another vertex when the latter is the destination or origin of the edge. Since the source node is represented as v and the destination node is u, the edge is represented as $e_{v,u}$;

c) Successor: is the vertex u that is the destination of the edge $e_{v,u}$ outgoing from v;

d) Chain: is a sequence of edges of a graph (directed or not), such that each edge has a vertex in common with the preceding edge (except for the first) and another vertex in common with the subsequent edge (except for the least);

e) Path or route: corresponds to a chain in which all the edges have the same direction or destination;

f) Cycle: it is a chain in which some nodes are connected in a way that they form loops. In networks, cycles may cause a message to propagate indefinitely and never reach the destination;

g) Hop: each movement of a message (or packet) from one device to another within a route is called a hop (HCF, 2009). The distance between the source node and the destination node is usually determined by the number of hops that a message needs to travel;

h) Symmetry: An edge is symmetric if the transmitter node can send a message through the edge and the receiver node can respond with an acknowledgment. An edge will only exist in a graph in this work if it is symmetric.

2.1.3 Structural features

The following structural features of graphs are used in this work:

a) A graph is connected when a chain can connect every pair of vertices of the graph;

b) A tree is a directed graph that has no cycles.

2.2 WIRELESS MESH NETWORKS

In WMN, all nodes in the network may act as routers. They must be able to receive messages originated from other nodes in the network and forward them towards the destination. WMN have characteristics as self-organization and self-configuration, where nodes automatically establish their connections with neighbors. In some WMN, a central manager is responsible for configuring nodes. Such features bring advantages such as easy maintenance of the network, easy insertion of new nodes, robustness and reliability (AKYILDIZ; WANG; WANG, 2005).

Two message delivery mechanisms are typically used in WMN:

a) Unicast: when a source node sends a message to a specific destination node;

b) Broadcast: when a node sends a message that all nodes in the network should receive.

In WMN, the data traffic generally flows from the nodes towards a central device and vice versa (YE; ZHANG; YANG, 2015). Typically, the central device is known as a gateway, base station, or sink. In WMN, many of the devices are connected to a continuous source of power (to the electrical grid, for example) and form a fixed infrastructure of communication, while other devices have limited resources and may have mobility (LI et al., 2011). These limitations often define several aspects related to the configuration and organization of these networks (KÜNZEL, 2012).

Three types of routing graphs are commonly used in mesh networks that have a gateway, and are exemplified in Figure 3:

a) Broadcast: connects the gateway towards all devices and is used to disseminate common configurations and control messages;

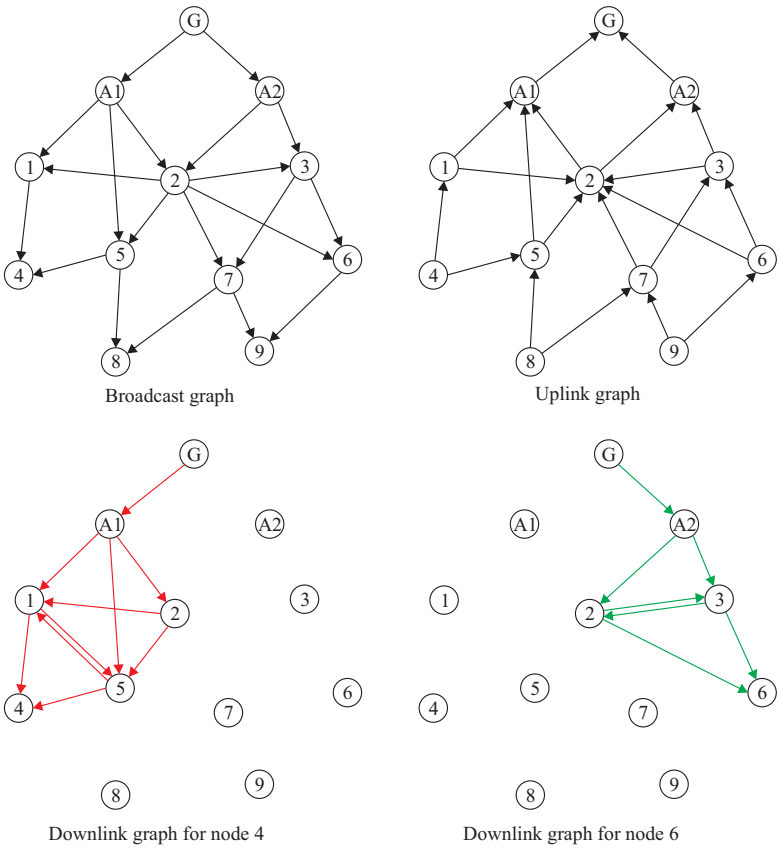
b) Uplink: connects all devices towards the gateway, and is used to send responses to configuration commands, requests, and sensor readings;

c) Downlink: Connects the gateway towards a specific device, and is used to send configurations, commands, and setpoint values to actuators.

2.3 REINFORCEMENT LEARNING

RL is a machine learning approach where a software agent acquires knowledge by exploring their environment without the need for external supervision. With RL, an agent can obtain information from the environment, learn, adapt, and make efficient decisions over time (SUTTON; BARTO, 2018). By performing different actions on the environment, in a trial-and-error concept, the agent causes changes in the state of the environment and seeks to learn

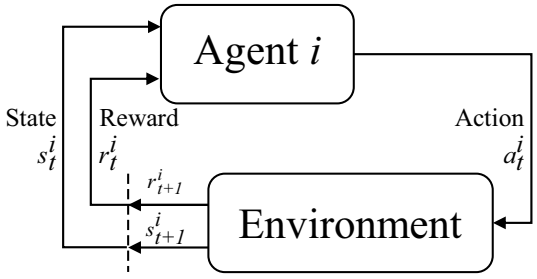
Figure 3 – Graph types.



Source – Adapted from Künzel (2012).

which actions maximize their long-term reward. (KOSUNALP et al., 2016). The reward is a numerical value that represents the goal of an RL problem. These two characteristics (trial and error and rewards) distinguish RL from the other ML approaches (SUTTON; BARTO, 2018). The interaction of an agent i with its environment is presented in Figure 4 and the basic model of RL is described in the following paragraphs.

Figure 4 – Agent interaction with the environment in RL.



Source – Adapted from Sutton and Barto (2018).

The RL problem is modeled as a Markov Decision Process (MDP). The Markovian property implies that the selection of an action by an agent at an instant t is dependent on an

action-state pair at the instant $t-1$ only (AL-RAWI; NG; YAU, 2015). In MDP, an agent is modeled by a tuple (S, A, P, R), where S is the set of all possible states of the environment. A contains all actions available to the agent. P is a probability matrix that represents the transition probability of a state in the instant t to another state in the instant $t+1$. Finally, R represents a reward function of the environment. The agent determines an optimal policy by evaluating the states and actions of the MDP and determining the actions that maximize the expected cumulative rewards (TOZER; MAZZUCHI; SARKANI, 2017).

In RL, the agent must be able to estimate the current state of the environment and to act by altering this state. The actions chosen will provide an immediate reward, but will also affect the accumulated rewards. Besides, there is a policy associated with transition probabilities, known as π, that defines the choice of the next action. The policy balances the exploration of actions not yet taken or with little knowledge about the rewards, and the exploitation of actions with knowledge about the rewards (KAELBLING; LITTMAN; MOORE, 1996). The balance between exploration and exploitation is described in section 2.3.2.

The concepts of states, actions and rewards are described below.

- States: An agent i has a set of states S that represents situations of its operating environment. At a given instant t, the agent i observes the state $s_t^i \in S$. The determination of which states will exist in the model are made at project time. In network communications, a state can represent internal factors such as buffer occupancy and transmission error rates, or external, such as delivery times and destination node;

- Actions: An agent has a set of actions A. Based on the observed state, the agent i learns to select an action $a_t^i \in A$, which will cause the transition in the environment from a state s_t^i to another state s_{t+1}^i and will maximize its immediate and future rewards. In network communications, an action may represent, for example, the choice of the next node to forward a message;

- Reward: Whenever an agent i performs an action $a_t^i \in A$, it receives a reward $r_{t+1}^i(s_{t+1}^i)$ from the environment. The reward function has a scalar value that represents some performance metric observed in the environment in state s_{t+1}^i. Cost equations can be constructed to represent the reward. The reward $r_{t+1}^i(s_{t+1}^i)$ is known as immediate or delayed reward, since it is received from the environment at the instant $t+1$. The discounted reward (or accumulated or future reward) represents the rewards expected to be received from the environment at discrete times $t+1$, $t+2$, ...

2.3.1 Q-Learning

Q-Learning is an approach to RL and has been applied in several works to improve some network characteristics (YAU; KOMISARCZUK; TEAL, 2012; AL-RAWI; NG; YAU, 2015;

GUO; YAN; LU, 2019; MAMMERI, 2019). Q-Learning defines a Q-function $Q_t^i(s_t^i, a_t^i)$, also known as a state-action function. The Q-function estimates Q-values, which are the long-term rewards that an agent expects to receive by taking a given action a_t^i in a given state s_t^i. The Q-values are updated at each iteration of the algorithm, taking into account the previously stored value and the new reward received. Each agent maintains a Q-table with $|S| \times |A|$ records where the Q-values are stored for each state-action pair. When the Q-values are learned in a non-dynamic environment, the π policy can be constructed simply by selecting the action a_t^i which has the largest Q-value in each state s_t^i (AL-RAWI; NG; YAU, 2015).

Equation 2.1 represents the Q-value update function. $0 \leq \alpha \leq 1$ is the learning rate and $0 \leq \gamma \leq 1$ is the discount factor. High values of α lead to faster learning and are usually dependent on the dynamism of the environment but may cause fluctuations in Q-values. When $\alpha = 1$, the agent only uses the new value given by $r_{t+1}^i\left(s_{t+1}^i\right) + \gamma \max_{a \in A} Q_t^i\left(s_{t+1}^i, a\right)$, and forgets the current value stored in $Q_t^i(s_t^i, a_t^i)$. In practical terms, $\alpha = 0.1$ is generally used for all t (SUTTON; BARTO, 2018).

$$Q_{t+1}^i\left(s_t^i, a_t^i\right) \leftarrow (1-\alpha)\, Q_t^i\left(s_t^i, a_t^i\right) + \alpha \left[r_{t+1}^i\left(s_{t+1}^i\right) + \gamma \max_{a \in A} Q_t^i\left(s_{t+1}^i, a\right) \right] \tag{2.1}$$

The discount factor γ allows an agent to adjust its preference for long-term rewards. When $\gamma = 0$, the agent considers only immediate rewards, whereas when $\gamma = 1$, the immediate and discounted rewards have the same relevance.

As Q-Learning is an iterative algorithm, it assumes an initial condition before the first update of the values occurs. The Q-table initialization is done during the project phase. Usually, Q-table is initialized with zero or random values. Algorithm 1 presents the traditional approach for implementing Q-Learning (AL-RAWI; NG; YAU, 2015).

2.3.2 Action selection: Exploration or exploitation in Q-Learning

One of the challenges of using RL is the choice between exploration and exploitation. The exploit selects the action $a_t^i = \arg\max_{a \in A} Q_t^i(s_t^i, a_t^i)$, which has the highest Q-value: when an agent exploits its environment, it chooses actions that it already knows that extend its rewards. The exploration selects a random action $a_t^i \in A$ to extend the knowledge regarding the Q-values for all the state-action pairs (AL-RAWI; NG; YAU, 2015).

The balance of exploitation and exploration helps to increase the rewards accumulated over time, and the agent must try a variety of actions and progressively favor those that seem to be better (SUTTON; BARTO, 2018). For the convergence of the Q-values, the exploitation can receive a higher priority, since the exploration may not discover better actions all the time. Several approaches have been proposed to balance exploration and exploitation:

Algorithm 1: Q-Learning algorithm. Adapted from Al-Rawi, Ng and Yau (2015).

Start $s_t^i \in S$; $a_t^i \in A$; $\alpha = [0.0, 1.0]$; $\gamma = [0.0, 1.0]$; $Q_t^i(s_t^i, a_t^i) = [Q_{min}, Q_{max}]$;
repeat
  Observe the state s_t^i;
  Select an exploitation or exploration action $a_t^i \in A$ following policy π;
  if *exploitation* **then**
    choose the best-known action $a_t^i = \arg\max_{a \in A} Q_t^i(s_t^i, a_t^i)$;
  end
  else
    choose a random action $a_t^i \in A$;
  end
  perform the action a_t^i on the operating environment;
  observe the next state s_{t+1}^i and reward and reward $r_{t+1}^i(s_{t+1}^i)$ at the next time instance $t + 1$;
  update Q-value $Q_t^i(s_t^i, a_t^i)$ using Equation 2.1;
until;

a) Greedy approach: the agent will always select the action with the highest Q-value. This approach leaves no room for exploration, and the agent will take a long time to adapt to changes (KOSUNALP et al., 2016).

b) ε-greedy: the agent probes with a probability $0 \leq \varepsilon \leq 1$ every possible action, to know the rewards of those actions. In tasks where bad actions exist, this approach can select actions with very low Q-values, which are unsatisfactory to the application. Even so, the approach tends to increase the long-term accumulated rewards in some applications (AL-RAWI; NG; YAU, 2015; SUTTON; BARTO, 2018). The exploration can be controlled by changing the value of ε. At the start of execution, ε may have a higher value, allowing deep exploration, and as time passes, or some performance parameter is reached, ε may be reduced to allow exploitation (KOSUNALP et al., 2016).

c) Softmax: the agent increases the probability of choosing actions that have a higher Q-value based on a Gibbs or Boltzmann distribution to determine the probability (DOWLING et al., 2005; SUTTON; BARTO, 2018).

2.3.3 Episodes

In some RL problems, the concept of an episode can be used. An episode has a limited number of iterations. At the end of an episode, a predefined initial state is set, and a new episode begins taking into account the previous learning. In problems where there is a well-defined initial and final state, the use of this concept allows the agent to make better use of his learning while exploring the environment and learning (SUTTON; BARTO, 2018).

2.4 THE WIRELESSHART PROTOCOL

This section describes the main features of the WirelessHART protocol focusing on specifications related to routing, as well as the simulators available for performance evaluation of the protocol.

2.4.1 WirelessHART devices

This section describes the devices specified by the standard. Figure 1 shows a representation of the devices and the typical connection of a WH network with an automation plant.

2.4.1.1 Field Devices

The field devices are the sensors and actuators connected to the process. Most of them are battery-powered, allowing quick installation and commissioning.

2.4.1.2 Adapters

The adapters have the function of connecting conventional (wired) HART devices to the WirelessHART network.

2.4.1.3 Routers

Routers are devices that have the task of forwarding messages. They are generally not required since all field devices have routing capability. However, they may be beneficial in expanding the size of the network and increasing the lifetime of battery-powered field devices (KÜNZEL, 2012).

2.4.1.4 Handheld

The handheld is used for commissioning the devices of the network.

2.4.1.5 Access point

The Access Point is a device that physically connects the wireless network to the gateway. The AP, gateway and NM are usually a single piece of equipment.

2.4.1.6 Network Manager

The NM is responsible for the management of the WirelessHART network and the configuration of the devices. Management is accomplished through several commands exchanged with the network nodes. Its main functions and characteristics are:

a) To initialize the network and provide means for the devices to join the network;

b) To monitor the network, obtaining information about the health conditions of the devices and communications;

c) To provide means for the network administrator to obtain run-time data and change the NM configuration;

d) To manage the network topology and routes used for data exchange;

e) To schedule communication resources according to the application;

f) To have a direct connection with the gateway, which enables it to communicate with the devices of the network;

g) To establish secure connections between devices, providing security keys used to encrypt information exchanged between NM, gateway, and devices. Typically, a software module known as Security Manager (SM) is responsible for these features.

2.4.1.7 Gateway

The gateway connects the WirelessHART network to the industrial automation plant. Its main characteristics are:

a) To have one or more APs that will make the physical connection to the wireless network;

b) To be the point of origin and destination for the data traffic of the WirelessHART network;

c) To connect the automation plant network to the wireless network through an interface containing different protocols;

d) To have a connection to the NM;

e) To forward commands generated and directed to NM (alarms, commands, health reports);

f) To store process data locally;

g) To be the clock synchronization source of the network;

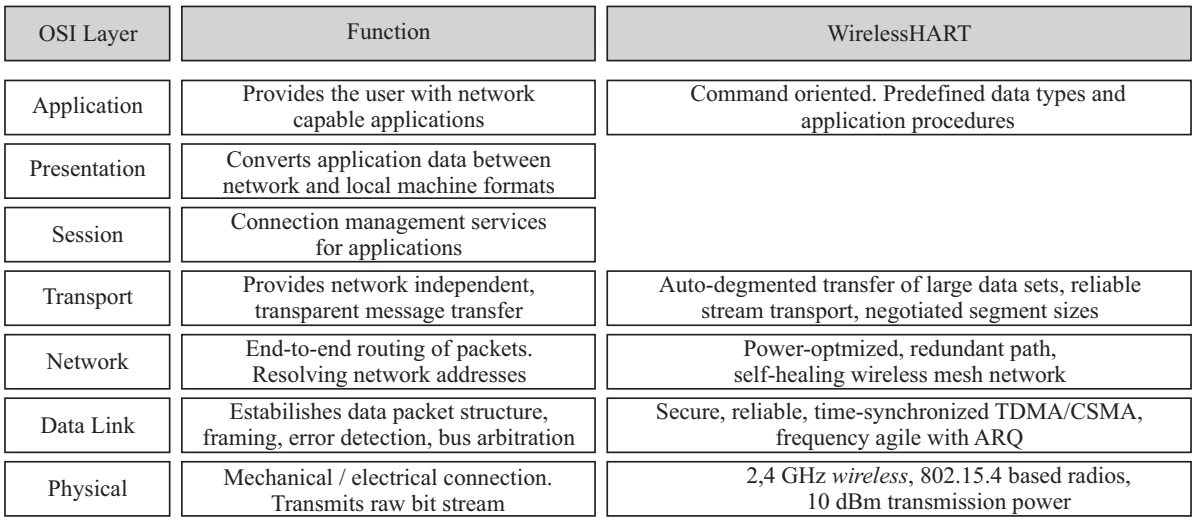
h) To support WirelessHART adapters;

i) To support universal and common standard HART commands.

2.4.2 WirelessHART layers

Figure 5 presents the architecture of the WirelessHART protocol according to the ISO / OSI model. The protocol has five layers: physical layer, data-link layer, network layer, transport layer, and application layer (HCF, 2007).

Figure 5 – WirelessHART ISO/OSI layers.



Source – Adapted from HCF (2008c).

2.4.2.1 Physical Layer

The physical layer of the protocol is responsible for transmitting and receiving data and incorporates most of the physical layer requirements of the IEEE 802.15.4 standard. The protocol operates over the Industrial, Scientific, Medical (ISM) frequency band in the range of 2.4 to 2.4835 GHz at a rate of 250 kbps. The channels are numbered from 11 to 25, with a 5 MHz bandwidth for each channel. Channel 26, available in IEEE 802.15.4, is not used because it is not allowed in some countries. All devices must have a transmitting power configurable between -10 and +10 dBm and a minimum sensitivity of -85 dBm (HCF, 2007).

Table 1 shows the expected communication distances at indoor and open environments, with and without a line of sight. Distances are estimated considering a unit-gain omnidirectional antenna with a transmission error rate of less than 1 %, without interference and reflective or obstacle attenuation effects, and with reception power of -82 dBm (HCF, 2007).

Table 1 – Communication distance.

Transmission power	Open field with line of sight	Indoor without line of sight
+10 dBm	200 m	75 m
0 dBm	100 m	35 m

Source – HCF (2007).

2.4.2.2 Data-Link Layer

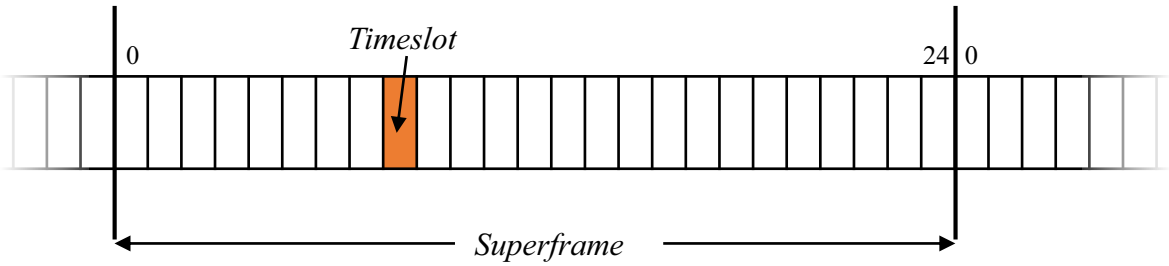
This layer defines reliable means for packet transmission between two devices, detecting transmission errors that may occur on the physical layer (HCF, 2008a). It can be split into two sublayers. The Logical Link Control (LLC) layer controls frame format, device address structure, security services to ensure message integrity and error detection. The Medium Access Control (MAC) layer controls when devices can transmit messages. The data-link layer uses Time Division Multiple Access (TDMA) to provide collision-free, deterministic communication. The communication channels are divided into 10 ms time frames (timeslots) in which the communications between the devices are performed. The MAC layer also keeps control of the number of timeslots that have already occurred since the network startup. This number is known as Absolute Slot Number (ASN).

2.4.2.2.1 Superframes

A superframe describes a sequence of consecutive timeslots repeated periodically. The period of a superframe is given by its length in timeslots. Figure 6 depicts an example of a superframe with a length of 25 timeslots. The cycle period has, therefore, 250 ms.

Timeslots can be simultaneously allocated in a superframe on different channels. It is assigned a channel offset number that associates the timeslot with a communication channel. A WirelessHART network can have multiple superframes active simultaneously, with different lengths. The superframe sizes must follow a harmonic chain. Multiple superframes can be used to allocate resources with different communication rates (HCF, 2008c).

Figure 6 – Superframe example.



Source – Adapted from HCF (2008a).

2.4.2.2.2 Timeslots

The NM allocates the timeslots available in a superframe for communication according to the configured routes and device demands. When two devices have a timeslot configured to communicate with each other, they have a link. Within the timeslot period, the transmitter is allowed to send a packet, and the receiver is allowed to send an acknowledgment (ACK) packet

back to confirm the reception correctness. A device sends a packet through a link if there are pending packets in its transmission stack.

Each link has the following properties:

a) Superframe number: the identifier (ID) of the superframe to which the link belongs;

b) Number: defines the index or position of the link timeslot within the superframe;

c) Type: defines the purpose of the link (normal, neighbor discovery, advertisement);

d) Origin and destination: identifies the transmitter and receiver nodes;

e) Options: defines, within the transmitter and receiver, if the link is used for transmission, reception or is shared;

f) Channel offset: provides the logical channel to be used in the link.

Shared links can be configured to save communication resources, where several devices compete for transmission to a receiving device. In this case, if two devices transmit simultaneously on the shared link, a collision will occur and will invalidate the contents of the received packet. The receiver will not send the ACK, and the transmitters will retry at the next available link. In broadcast links, receivers do not send the ACK.

The standard uses a mechanism to synchronize the clocks of nodes to ensure the operation of TDMA. Channel blacklisting is used to disable channels affected by interference. Each device has a table containing the active channels of the network. Channel hopping is also used, in which channel jumps are performed at each timeslot, which provides a diversity of frequencies that reduce the effects of attenuation by reflections and obstacles. Based on the active channel table, the channel offset value configured for the link and the ASN, the device can determine the physical channel to be used.

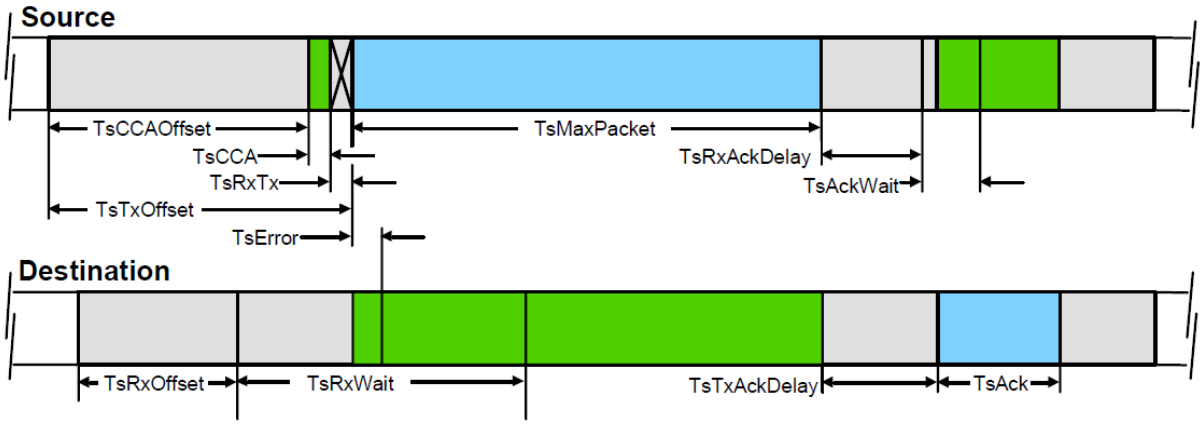
2.4.2.2.3 Timeslot structure

Figure 7 presents the detailed time structure of a timeslot, from the transmitting device and the receiver. Table 2 describes the utility of each of the fields within the timeslot.

2.4.2.2.4 Communication tables

The devices maintain a series of tables at the data-link layer. Superframe and link tables indicate the timeslots available on each superframe. Neighbor tables provide statistics on the transmissions and receptions for each neighbor and store information about potential neighbors discovered in the network. Graph tables contain IDs for different routes. Graph-neighbor tables associate the graph IDs with the neighbors. Data buffers store incoming and outgoing packets.

Figure 7 – Timeslot structure.



Source – HCF (2008a, p. 44).

Table 2 – Timeslot timing components.

Field	Definition
TsCCAOffset	Time for the timeslot start
TsCCA	*Clear Channel Assessment* (CCA), to check if the channel is available
TsRxTx	Radio switch between reception and transmission
TsTxOffset	Time between the timeslot start and the preamble transmission
TsMaxPacket	Maximum transmission time (133 bytes)
TsRxAckDelay	Wait for ACK reception start
TsAckWait	ACK reception time
TsError	TDMA synchronization reference
TsRxOffset	Time between timeslot start and reception activation
TsRxWait	Minimum time to wait for a packet
TsTxAckDelay	ACK generatiom time
TsAck	ACK transmission time

Source – HCF (2008a, p. 47).

The minimum size of these tables is shown in Table 3, and specifies the minimum number of data structures that a device should be able to support.

2.4.2.2.5 Frame types

The data layer of the link layer is also known as the Data-Link Protocol Data Unit (DLPDU). Advertise DLPDUs are used to disseminate network information to devices that want to join. ACK is used to confirm packet reception. Data DLPDUs are used to send application layer information. Keep-Alive (KA) DLPDUs are used for the devices to verify the conditions of the connection with linked neighbors and for synchronization. Disconnect DLPDUs indicate when a device is leaving the network.

Table 3 – Memory Requirements for Data-Link Layer Tables.

Table	Minimum number
Neighbors	32
Superframes	16
Links	64
Graphs	32
Graphs-neighbors	128
Message Buffers	16

Source – HCF (2008a, p. 47).

2.4.2.3 Network layer

The network layer provides functionality for reliable end-to-end communications between network devices. All devices must be able to forward messages on behalf of other devices.

WH networks can be constructed with different topologies. The protocol is flexible and allows the combination of star and multi-hop topologies in the same network (ZAND et al., 2014b). A star topology, with the AP positioned in the center of the network, allows high-performance applications with low latency. A multi-hop topology can be used when large areas must be covered, and low latency is not a concern.

There are four types of routing available in the protocol: Graph, superframe, source, and proxy. The standard does not specify algorithms for constructing routes. The only recommendation is that redundancy of paths should be available in the routes used (HAN et al., 2011). The network layer header of a message contains information about the routing mechanism to be used.

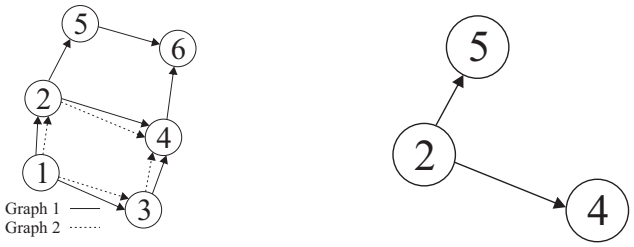
2.4.2.3.1 Graph routing

Graphs consist of a collection of paths that can be used to route a package from its source to the destination. To send a message the source device adds the graph ID to be used in the network layer header. The NM must configure devices on the destination path with information that specifies the neighbors to forward the packet with a given ID.

Figure 8 depicts the concept of graph routing. On the left, two graphs used by device 1 to send messages to device 6 (graph ID = 1) and to device 4 (graph ID = 2). On the right side, the neighbors configured in device 2 for the graph ID = 1 are shown.

As shown in Figure 8, this type of graph allows path redundancy, since more than one neighbor can be configured for sending messages to a destination node. At most, four neighbors can be used as next-hop for the message. Devices transmit packets using the first available link with any of the next neighbors, on any of the superframes. A representation of the typical graphs used was presented in Figure 3.

Figure 8 – Graph IDs and neighbors configured in node 2 for a given Graph ID.

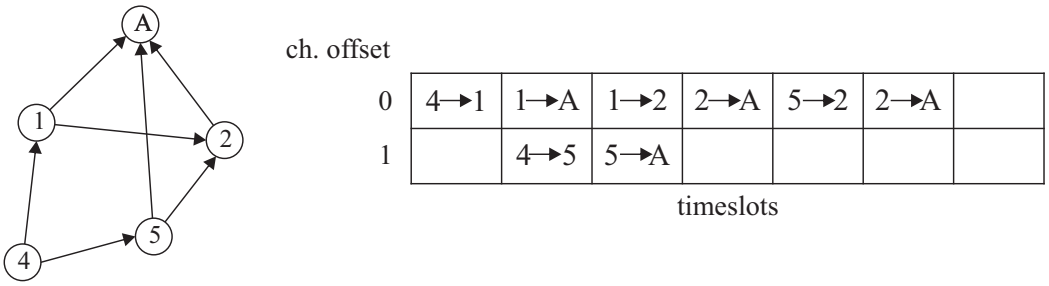


Source – Adapted from HCF (2009).

2.4.2.3.2 Superframe routing

Superframe routing is a special case of graph routing. Packets are designed to be sent by the device on any available link within a specific superframe. All existing links in a superframe must form a path that reaches the destination node. Figure 9 depicts an example of an uplink graph and the links configured in the superframe.

Figure 9 – Superframe routing example.



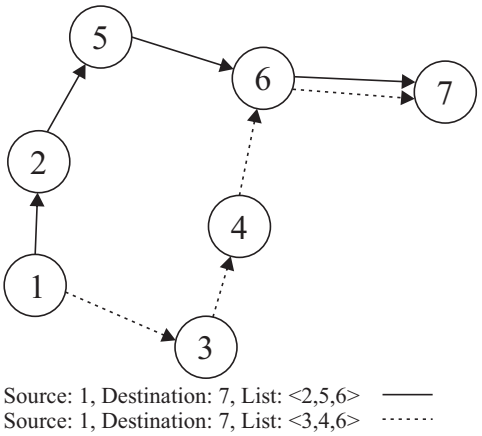
Source – Künzel (2012).

When superframe routing is used, the superframe ID is placed in the network layer header of the message. Superframe IDs have values different from those used for graph IDs.

2.4.2.3.3 Source routing

In this type of routing, the source device adds to the message header an ordered list of up to eight nodes through which the packet is routed. Each intermediate device reads the list to identify the next neighbor. Figure 10 depicts two possible source routes between devices 1 and 7. Routing by source provides a non-redundant path for sending the message. If one of the connections between the neighbor list is no longer available, the message is lost. This type of routing is mostly used by NM, which knows the complete topology of the network, to send configuration commands. Like graphs routing, source routing uses the first available link in any of the superframes. It is not used for process data.

Figure 10 – Source routing.



Source – Adapted from HCF (2009).

2.4.2.3.4 Proxy routing

This type of routing is only used when a device is joining the network. Another device, already in the network, is called proxy and has the function of mediating the communications between NM and a joining node.

2.4.2.3.5 Network tables

The devices maintain several tables at the network layer, which are also used in the transport layer. Session tables manage the security of the communications. Transport tables allow nodes to ensure message delivery. Route tables indicate the graphs and superframe IDs to be used for communication. Source tables contain the list of intermediate nodes that should be used when source routing is used. Service tables associate services with the routes for sending data, such as process variables. The minimum size of these tables, specified in the standard, is shown in Table 4.

Table 4 – Memory requirements for the network layer tables.

Table	Minimum number
Sessions	8
Neighbor	1 by session
Transport	2 by session
Routes	8
Source routes	2
Services	16

Source – HCF (2009, p. 66)

2.4.2.3.6 Routing requirements for the NM

Some of the NM tasks and recommendations regarding routing and specified in the standard are described below (HCF, 2008c, p. 117).

- NM keeps an updated internal representation of the topology of the network. This internal representation is used to generate the routes;
- NM collects network statistics and neighbors information through periodic reports and uses this information to choose between existing connections and make decisions about the formation of new ones;
- NM constructs graph routes. Graph routing is ideal for process data like sending sensor readings, reporting alarms, and sending commands to actuators;
- NM constructs a broadcast graph for all devices;
- NM constructs a downlink graph for each device;
- NM constructs source routes;
- NM avoids building routes with cycles.

The standard recommendands that there should be no cycles in the graphs, in order to prevent messages from circulating indefinitely in the network. However, some routing algorithms found in the literature insert cycles in the neighbors of the recipient in downlink graphs to increase reliability (HAN et al., 2011), as can be seen in Figure 3.

2.4.2.3.7 Comparison between the routing methods provided in WirelessHART

A comparison of the graph, source and superframe routing is presented by Künzel (2012) using four comparison criteria: traffic isolation; latency predictability; path redundancy; and memory resource usage. Table 5 presents a comparative summary between the routing methods.

Table 5 – Comparison between the routing methods provided in the WirelessHART standard

Routing method	Traffic isolation	Latency predictability	Path redundancy	Resource usage
Graph	Low	Low	Yes	Mean
Souce	Low	Low	No	Low
Superframe	High	High	Yes	High

Source – Adapted from Künzel (2012).

One of the differences pointed out is that in the graph and source methods the packets will use any link configured in the transmitter. Superframe routing can then be used to isolate traffic from management messages and process variables by creating, for example, an exclusive superframe for the flow of messages from devices to the gateway (sensor data) and from devices to the NM (configuration and command responses). The same concept can be used to communication from the gateway towards the devices (commands to the actuators) and from NM to devices (configuration and commands) (KÜNZEL, 2012).

The predictability of the latency is higher in the superframe routing since the process data traffic can be isolated in a superframe. The latency also depends on the number of links allocated between neighbors. The more links allocated, the more opportunities for communication a device will have. However, in networks with many devices, allocating too many links increases the channel occupancy, leads to higher power consumption, and reduces the expansion capacity of the network (KÜNZEL, 2012; CHEN; NIXON; MOK, 2010).

The redundancy is related to the reliability of communications, and is greater in graph and superframe routing since several neighbors can be configured, whereas in the source routing only one neighbor is used.

The utilization of memory resources of the devices in each routing method is analyzed using Tables 3 and 4. Resource utilization is higher in superframe routing because devices have a limited number of superframe entries in their memory, and more links must be allocated for communications. Resource use for graph routing is smaller, since the nodes have more entries for graphs and neighbor pairs in its memory, and a smaller number of links needs to be configured. Finally, source routing has the lower resource utilization, considering that the source node sends the list of nodes and that the NM must configure a small number of links between the list of nodes in the path.

2.4.2.4 Transport layer

The transport layer has the function of ensuring that packets are successfully routed between source and destination nodes. It also aggregates commands and multiple requests and responses into a single packet. The protocol supports services with and without acknowledgment of delivery. According to the standard, delivery confirmation services are used primarily in management commands. Packages with process data are generated periodically and do not require delivery confirmation.

2.4.2.5 Application layer

The application layer is based on commands, which are sent by the gateway or by the field devices. Devices must support some of the native HART commands, and there are specific commands for WH. Commands 768-1023 are used for network management and gateway functions and can be classified into the following categories: management of superframes and

links; management of graphs and source routing; management of bandwidth and services; status reports, and device diagnostics (ZAND et al., 2014b).

Several commands defined in the protocol are relevant for use in configuring and defining routes. Table 6 presents commands that provide relevant information to the definition of routes. These commands can be classified in dynamic, when they are periodically sent to NM and can change their parameters over time, or static when they do not change (characteristics of the device, for example).

Table 6 – Application layer commands related to route definition

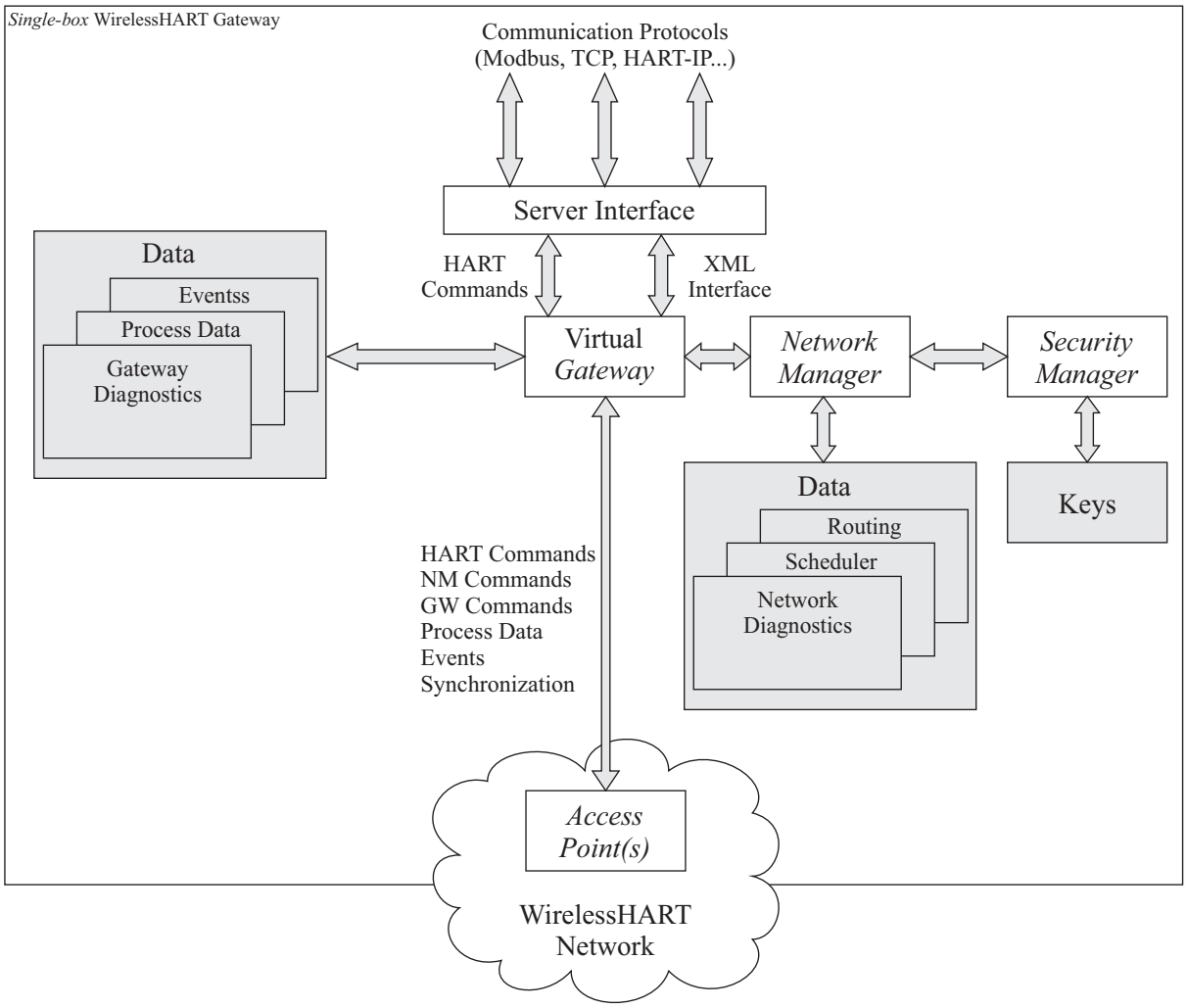
Command	Feature	Description	Information Available
777 - *Read Wireless Device Capabilities*	Static	Device characteristics	Power source type, *Received Signal Level* (RSL) sensitivity, maximum number of neighbors, buffers sizes and message interval
778 - *Read Battery Life*	Dynamic	Battery level	Remaining battery in days
779 - *Report Device Health*	Dynamic	Device status	Power source status, packets sent and received, data-link, network and transport layer failure counters
780 - *Report Neighbor Health List*	Dynamic	Status of neighbors with links	Neighbor's RSL, packets sent, received and failed
787 - *Report Neighbor Signal Levels*	Dynamic	Discovered neighbors (without links)	Neighbor's RSL
788 - *Alarm Path Down*	Dynamic	A neighbor connection is unavailable	Neighbor
789 - *Alarm Source Route Failed*	Dynamic	Source routing failed	Neighbor
790 - *Alarm Graph Route Failed*	Dynamic	Graph routingfailed	Graph ID

2.4.3 Single-box architecture for the gateway and the NM

The standard also presents possible implementations for the WH gateway. One of the suggested implementations integrates gateway, AP, NM, and SM functionalities into single hardware (single-box). Figure 11 presents the single-box architecture with its main elements. This architecture is used in the WH simulator presented by Zand et al. (2014b).

A virtual gateway (implemented purely in software) has the function of interconnecting all the other elements. It stores process data, events, and diagnostic data. It has connections with

Figure 11 – Single-box architecture.



Source – Adapted from HCF (2008c).

one or more APs, through which the transmission and reception of messages from the wireless network take place.

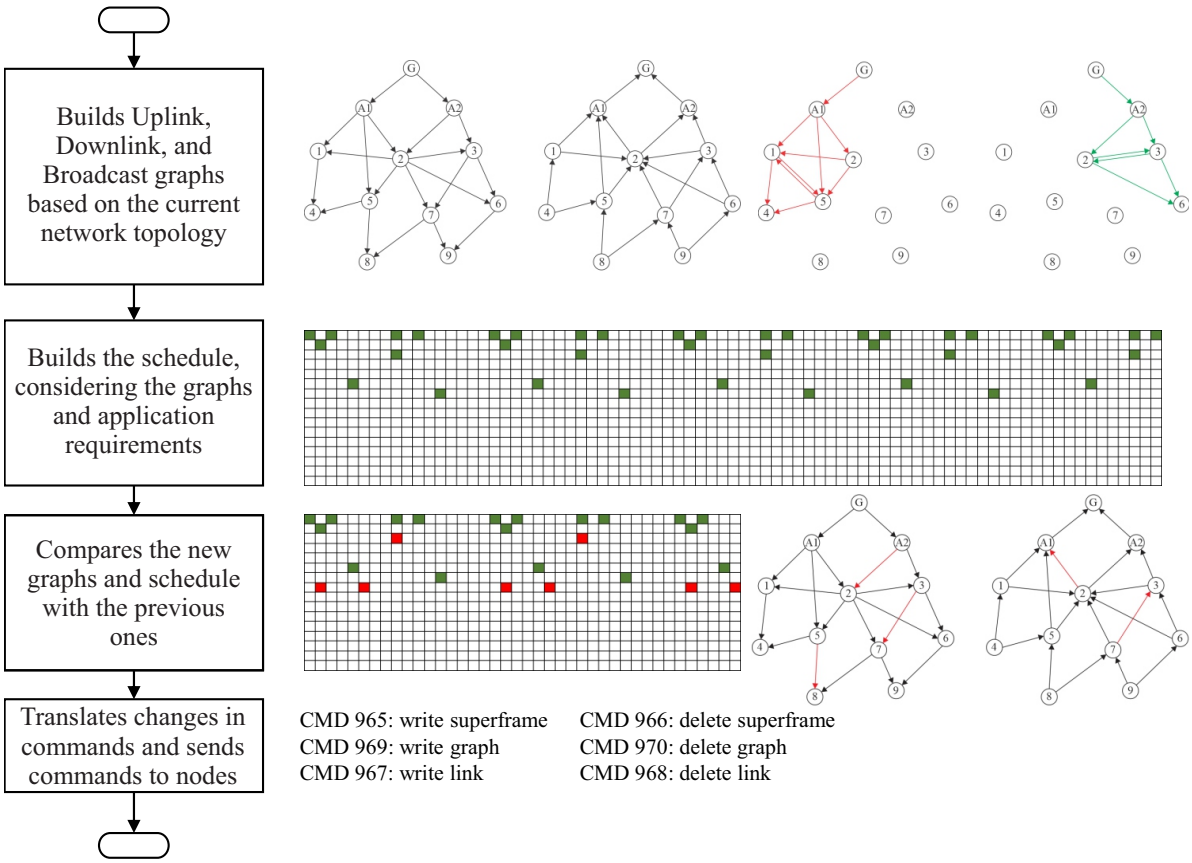
The virtual gateway has a connection to the NM, through which NM exchanges commands to the wireless network. The NM stores topology, routing, scheduling, and diagnostic data in its memory and has a connection with the SM to provide access to the keys used in the encryption of the messages.

The server interface provides, through different automation protocols made available by the manufacturer, access to the data of the sensors and actuators in the wireless network. Data exchange between the interface and the gateway is done through HART commands or through an Extensible Markup Language (XML) file.

2.4.4 The tasks of the Network Manager

As mentioned before, protocols such as WirelessHART use centralized management, where the NM is responsible for the overall configuration of the network. However, the WH standard does not define a specific algorithm or sequence of tasks for management. In general, management routines are performed when the network topology changes (whenever a node joins or leaves the network or a path down alarm occurs), when a node requests a service to the NM, or periodically, to optimize the use of network resources. In the works of Han et al. (2011) and Zand et al. (2014b), the network management routines follow the steps shown in Figure 12.

Figure 12 – Management routine of the NM according to Han et al. (2011), Zand et al. (2014b).



Source – The author.

The current topology is used by the routing algorithms to build the different routes and graphs needed, which are then translated by the scheduler into links, superframes, and graphs. Finally, the routes and schedules are converted into a sequence of commands sent to the nodes to update the network configuration (SHA et al., 2017; HAN et al., 2011). Sending these commands causes a communication overhead. NM reduces this overhead by comparing the old and new routes and schedules and by updating the changes only. To ensure path availability during reconfiguration, the new routes and schedules are first configured, and only then the old ones are removed (ZAND et al., 2014b).

2.5 THE WIRELESSHART SIMULATORS

This section present the WirelessHART simulators that have been developed over the past years to evaluate the performance and the applicability of the standard. The simulators were divided into subsections by frameworks and platforms, highlighting the development details and the limitations of each simulator.

In general, these simulators have a partial implementation, focusing on the physical and data-link layer to evaluate energy consumption, medium access control and synchronization. Other simulators have a complete implementation of the WirelessHART stack, allowing studies related to routing and scheduling, data transmission, security, protocol improvements and applications. To the best of our knowledge, the implementation developed by Zand et al. (2014b) over the NS-2 is the most complete approach, in terms of stack implementation.

2.5.1 Simulators using the COOJA framework

The work of Konovalov (2010) makes use of the discrete event-oriented operating system Contiki (DUNKELS; GRONVALL; VOIGT, 2004) and the COOJA simulation framework (OSTERLIND et al., 2006). It creates a hybrid simulation tool, which allows simulating a network with several WH devices, while real WH devices can communicate with the simulated devices. The COOJA simulator was installed on a computer to simulate some nodes and a bridge device operating on the physical and link layer of the WH was developed to interface the simulator and the real devices. In the bridge, Contiki was used to reach the timing requirements of the WH. A commercial NM was used with the simulator and the actual devices to validate the system.

2.5.2 Simulators using the OPNET framework

The OPNET simulator (CHANG, 1999) was used for the development of the simulation platform proposed in Gao, Zhang and Li (2012). Authors implemented the MAC mechanisms to evaluate performance, resource allocation, and scheduling. A comparison with the ZigBee standard was made to analyze the platform feasibility. Another work that uses OPNET is presented in Wang and Barac (2013). It implements the MAC layer and proposes an improvement in the standard through the use of the Carrier Sense Multiple Access - Collision Avoidance (CSMA/CA) mechanism in some shared slots.

2.5.3 Simulators using the Network Simulator 2 and 3 frameworks

The Network Simulator 2 (NS-2) is an open source framework environment for simulating discrete events for networks (MAHRENHOLZ; IVANOV, 2004). This simulator uses C++ for programming the protocols, and the Object-Oriented Tool Command Language (OTCL) to configure static and dynamic parameters of the simulation scenario.

Zand et al. (2014b) implement the complete WirelessHART stack in NS-2. It is possible to build a complete topology with NM, APs, and nodes. This simulator allows the collection of files and reports containing information about the nodes like energy consumption, neighbors list, connection list, latency, graphs, among others. The Han algorithms (routing and scheduling) are implemented in the NM. The simulator was validated through comparison with a real WH network, where authors evaluated different network parameters over time, such as transmission failure rates, RSL in some links and command response times. The implementation does not have realistic models of power consumption and transmission errors (NOBRE; SILVA; GUEDES, 2014). Authors in (BAYOU et al., 2015) also mention that the Zand simulator does not have a proper implementation of the security layer.

The Network Simulator 3 (NS-3) is an evolution of the NS-2. It was used by Nobre, Silva and Guedes (2015a) to develop a WH physical layer module. According to the authors, NS-3 has better scalability, memory management, documentation, and simulation-time performance when compared to NS-2. The developed physical layer module includes the Gilbert/Elliot transmission error model along with a signal attenuation model, battery consumption model, and a node positioning tool. The module was validated through simulation with different WH topologies. Routing and scheduling were statically configured in the experiments.

2.5.4 Simulators using the OMNET++ framework

The OMNET++ environment (VARGA; HORNIG, 2008) was used as the basis for several WH simulators. In Liu et al. (2016), the stack was partially implemented to integrate the network with a control system, allowing the study of the interactions between these two components and the evaluation of the system performance during design. In Bayou et al. (2015), authors partially implements the stack to investigate the security mechanisms used in the standard. Authors in Ferrari et al. (2013) develop a tool that combines OMNET++ with the MATLAB's TrueTime library (HENRIKSSON; CERVIN; ÅRZÉN, 2003), allowing the simulation of control systems using WH. The tool was used to analyze the performance of a control system and the coexistence with other wireless networks.

3 ROUTING AND REINFORCEMENT LEARNING IN IWSN

This chapter presents the analysis of the state of the art in the subjects of graph routing algorithms and RL approaches for routing in wireless networks. Each section of this chapter analyses one subject. The analysis is conducted in the following sequence: Initially, the main criteria used to define the pertinence of the works are described. Then, a description of each work is presented. A comparison is made by observing the main characteristics of each one. Finally, open aspects and possible contributions to the subjects are identified.

3.1 ROUTING IN IWSN

This section presents an analysis of the works related to reliable routing in IWSN, emphasizing those related to the WirelessHART and ISA SP100.11a protocols. The WirelessHART standard suggests that there may be path redundancy in the routes, making the network more reliable (CHEN; NIXON; MOK, 2010). The works considered in this analysis are mostly those that construct graphs or routes with redundant paths. In section 3.1.1, a comparison of the works is made observing the criteria used in the construction of routes, types of routes created, metrics used, experiments and validation forms used. Section 3.1.2 presents the considerations about the algorithms and identifies possible contributions.

Jindong, Zhenjun and Yaopei (2009) present the Exhenced Least-Hop First Routing (ELHFR) algorithm, which receives as input the topology graph and builds an uplink graph. A Breadth-First Tree (BFS) algorithm is used to find a tree that has the smallest path for each node to the gateway. BFS is also using to assign a level for each node, where the number of hops from the gateway is identified. Subgraphs with redundancy are generated using the smallest paths from a node to a destination. RSL information is used to choose connections.

Han et al. (2011) present algorithms for the construction of reliable broadcast, uplink and downlink graphs, and a scheduling algorithm for those graphs. According to Nobre, Silva and Guedes (2015b), it is one of the most relevant works for the WH protocol. The algorithms are of the greedy type, where each graph is constructed iteratively. At each iteration, a node in the topology is selected and added to the resulting graph, along with connections to its neighbors. The average number of hops from the gateway is used to choose nodes and connections. This criterion, also known as degree, reduces the number of hops between the gateway and devices, thus reducing the latency and the use of communication resources. Figure 3 presents examples of the graphs constructed by the Han algorithms. The routing algorithms are evaluated using criteria such as the number of reliable nodes and graph construction success rates. Subsequent

works such as Künzel (2012), Zand et al. (2014b), Wu et al. (2016), Cainelli, Künzel and Pereira (2017), Künzel et al. (2018) and Madduma-Bandarage (2020) use the Han algorithms in the comparisons.

Künzel (2012) proposes an environment for the evaluation of graph routing algorithms. Graphs are evaluated based on the connections and node characteristics. Topologies can be captured from an operational network or created in the environment. The author analyses the Han algorithms and identifies that in denser networks (with many devices), the algorithm causes an imbalance in the network, concentrating routing in a few devices. This concentration is not suitable for practical applications since the devices have memory and energy limitations. The Han algorithm is adapted to use a cost equation with weights that consider characteristics such as the power source of the devices, the distance from the gateway and the current number of neighbors in the graph.

Memon and Hong (2013) propose a load-balanced routing algorithm, splitting the routing responsibility between nodes and increasing network lifetime while maintaining path redundancy and a reduced number of hops. Routing is done through a division of the network topology into levels related to the distance from the gateway. With the level information, uplink, downlink, and broadcast graphs are built through the connection of devices of different levels. An equation defines the value of the routing responsibility of each device.

The Joint Routing Algorithm for Maximizing Network Lifetime (JRMLR) is presented by Zhang, Yan and Ma (2013) to maximize the lifetime of a WH network. An exponent-weighted cost function uses the energy consumption of a single transmission, the communication load factor of the nodes, and the destination's residual energy to choose routes. The best route to a node is the one in which the communication load and transmission power are minimal and the residual energy is maximum (NOBRE; SILVA; GUEDES, 2015a). The algorithm is implemented in a MATLAB simulator and then compared to the ELHFR.

The Re-Add algorithm, proposed by Zhang et al. (2014), is similar to JRMLR. The algorithm defines a priority for each link, using criteria such as link quality, residual energy and level differences between devices. A moving average estimator is used to define the quality of the link. Through a judgment matrix, weights are assigned to each criterion to determine the priority given to each link. The highest priority links are added to the graph. The performance is compared with the Han algorithms and JRMLR.

Wu et al. (2015) propose a real-time routing method for Wireless Sensor and Actuator Networks (WSAN) when there are conflicts between the transmissions (data flows). Conflicts between the transmissions that share a common router device contribute significantly to the communication delays in networks such as WH. By incorporating the conflicts in the routing decisions, a WSAN can support a greater number of real-time flows and to fulfill its deadlines. The source routing method is used in this work. The validation of the algorithms is based on simulations and practical experiments that use a TDMA protocol on the physical layer of

802.15.4. The work analyses parameters such as latency and acceptance rate, of greater relevance in the scheduling.

In the work of Wu et al. (2016), the authors formulate the problem of the maximization of the network lifetime using routing by graphs taking into account aspects of the scheduling and the flows of information. Three solutions are presented: an optimal solution based on integer programming; a relaxed approach using linear programming; and another with greedy heuristics. The heuristic solution builds the graph with the lowest normalized load, which is calculated based on the expected energy consumption rate divided by the initial capacity of the battery in each device. The main idea is to allow devices with higher battery capacity to carry more traffic when using graph routing. The authors mention that the approach with linear programming is complex and takes a long time to execute, which makes it unsuitable for practical applications.

Hong et al. (2015) propose an Energy-Balancing Graph-Routing (EBGR) algorithm that achieves longer network lifetimes by graph reshaping. The algorithm uses BFS to divide the WH network into levels and then uses a graph reshaping algorithm to redistribute the energy consumption to nodes with smaller routing responsibility.

A Multipath Routing Algorithm (MPAR) is proposed by Sepulcre, Gozalvez and Coll-Perales (2016). It identifies redundant routes that are needed to satisfy the end-to-end reliability and the latency of an application. MPAR uses probabilistic estimates of reliability and latency to guarantee the Quality of Service (QoS). Three types of routes are built: Routes that may have nodes and links in common; without links in common; and without nodes in common. The paper also mentions the need to change the WH standard so a message can be transmitted simultaneously to different neighbors. The approach was compared with other protocols with single and multiple paths and with a WH-based protocol.

The work of Cainelli, Künzel and Pereira (2017) adapts the broadcast algorithms of Künzel (2012) and Han et al. (2011) to build graphs using four characteristics: power type of devices, RSL, distance from the gateway, and the number of neighbors. A cost equation with weights adjusts the relevance of each parameter used in the graph construction. It is possible to prioritize the construction of graphs with the following characteristics: reduction of the routing function in battery-powered nodes, which increases the network lifetime; higher RSL between neighbors, which reduces possible transmission errors; smaller distance in of hops between gateway and device, which reduces latency and communication resources usage; and network balance, distributing the communication load between the devices. Different sets of weight values are evaluated on static topologies containing battery-powered and line-powered devices using the simulation tool of Künzel (2012).

Han, Ma and Chen (2019) present the EBREC algorithm. The BFS algorithm is used to define a network hierarchy, then the routing path is generated according to the energy consumption of each layer. Using criteria as the minimum hops and redundancy of paths, it balances the energy consumption of the whole network, prolonging the lifetime.

The work of Madduma-Bandarage (2020) present a routing and scheduling algorithm to improve the performance of IWSN. The Frame Level Optimized Reliable Graph (FLO-RG) is a graph-routing algorithm based on the approach for lifetime optimization in IWSN of Herrmann and Messier (2018). The FLOR-RG consists of primary path and backup paths. A MAC protocol is introduced to reduce the wasted energy in idle listening in backup slots. A petroleum refinery wireless sensor network model complying with WirelessHART/ ISA100.11a industrial standards is simulated using the Network Simulator 3 (NS-3) package to evaluate the proposed FLO-RG algorithm.

3.1.1 Algorithm comparison

A classification of the state-of-the art routing algorithms is made by Nobre, Silva and Guedes (2015b) with the following criteria: routing construction objectives; route definition criteria; constructed routes and graphs (downlink, uplink and broadcast); use of historical information (packet transmission statistics); and ways of implementation, presentation and validation of the algorithms. The authors pointed out that the (HAN et al., 2011) algorithms were the most complete, considering the evaluated aspects. This affirmation is reinforced by the number of citations of the paper in recent works.

To complement the analysis in this thesis, other characteristics were included in this comparison, which are described below:

a) Primary path: if the routing algorithm defines a primary path for each node, that is, the one by which a node will make the first attempt to send a packet, and what is the criterion for this choice. This feature is important because some scheduling algorithms use this information to allocate more links in the primary path;

b) Implementation on an industrial protocol: whether the algorithm was implemented in a simulation environment or in a real network that has the complete stack of an IWSN protocol. This feature is relevant because it indicates if the results of the experiments were obtained in a IWSN protocol;

c) Parameters and scenarios used: parameters such as area, number of devices, network layout, power characteristics of devices, communication range, among others used in the evaluation;

d) Performance metrics evaluated: information collected regarding the performance of the algorithms, such as latency, packet loss, network lifetime, reliability, energy consumption, among others.

Table 7 presents the comparison of the algorithms concerning their constructed routes and graphs, route construction objectives, route definition criteria, use of historical data and

the definition of a primary path for each node. It is observed that most of the works do not implement all the graphs suggested in the WH standard. Most algorithms aim to increase the network lifetime, since the use of battery-powered devices in predominant in IWSNs. The main route definition criterion uses the degree (distance in hops) from the gateway, because this metric reduces latency and the use of communication resources. Besides, some of the works define weights in equations that will determine the cost of each neighbor or route chosen. These weights are defined off-line, before running the simulations. Only one work presents an approach to adjust the weights automatically. Historical data such as link quality and packet delivery rates are used in three works. Finally, many works define the existence of a primary path.

Table 7 – Graphs, objectives and criteria used in the routing algorithms

Algorithm	Routes	Objectives	Construction criteria	Historical data	Primary path
(JINDONG; ZHENJUN; YAOPEI, 2009)	Uplink, downlink	Redundancy	Degree	No	Highest RSL
(HAN et al., 2011)	Uplink, downlink, broadcast	Lifetime, resource usage	Degree	No	Lowest Degree
(KÜNZEL, 2012)	Uplink, broadcast	Lifetime, load balance	Degree, power source, load	No	Lowest Cost
(MEMON; HONG, 2013)	Uplink, downlink, broadcast	Lifetime	Degree, load	No	No
(ZHANG; YAN; MA, 2013)	Uplink	Lifetime	Residual energy, transmission energy and traffic load	Yes	Lowest Cost
(ZHANG et al., 2014)	Uplink	Lifetime, robustness	Residual energy, transmission energy, degree	No	Highest flow priority
(WU et al., 2015)	Source	Real time requirements	Conflict delays in the flows	No	No
(WU et al., 2016)	Uplink, downlink	Lifetime	Battery consumption rate	No	Highest flow priority
(HONG et al., 2015)	Uplink	Lifetime	Traffic load	No	Load
(SEPULCRE; GOZALVEZ; COLL-PERALES, 2016)	Source, downlink	QoS	Latency, packet delivery ratio	Yes	Lowest latency

Table 7 - continued

Algorithm	Routes	Objectives	Construction criteria	Historical data	Primary path
(CAINELLI; KÜNZEL; PEREIRA, 2017)	Broadcast	Lifetime, transmission errors	Degree, power source, load, RSL	No	Lowest cost
(HAN; MA; CHEN, 2019)	Uplink	Lifetime	Residual energy, RSL, degree	No	Highest energy
(MADDUMA-BANDARAGE, 2020)	Uplink	Lifetime	Packet reception ratio, expected energy consumption, normalized load	Yes	Lowest degree, bit and level optimization

Table 8 presents characteristics about the presentation, validation and implementation of the algorithms. The algorithms are all implemented in simulation environments and only two works presenting experiments using a TDMA protocol on the data-link layer of the 802.15.4 stack. The Han et al. (2011) algorithm was implemented in the Zand et al. (2014b) simulation environment over the WH stack. The experiments.

Table 8 – Presentation, validation and implementation of the routing algorithms

Algorithm	Presentation	Validation	Implementation in an IWSN protocol
(JINDONG; ZHENJUN; YAOPEI, 2009)	Textual description	Simulation in OMNET++	No
(HAN et al., 2011)	Pseudocode	Own simulator	WH in (ZAND et al., 2014b)
(KÜNZEL, 2012)	Pseudocode	Own simulator	No
(MEMON; HONG, 2013)	Fluxogram	Own simulator	No
(ZHANG; YAN; MA, 2013)	Textual description	Author's simulator	No
(ZHANG et al., 2014)	Textual description	Author's simulator	No
(WU et al., 2015)	Pseudocode	Own simulator and real experiment	No
(WU et al., 2016)	Pseudocode	Own simulator and real experiment	No
(HONG et al., 2015)	Textual description, fluxogram	Simulation in MATLAB	No

Table 8 - continued

Algorithm	Presentation	Validation	Implementation in an IWSN protocol
(SEPULCRE; GOZALVEZ; COLL-PERALES, 2016)	Fluxogram	Simulation in MATLAB	No
(CAINELLI; KÜNZEL; PEREIRA, 2017)	Textual description	Künzel (2012) tool	No
(HAN; MA; CHEN, 2019)	Pseudocode	Simulation in MATLAB tool	No
(MADDUMA-BANDARAGE, 2020)	Pseudocode, fluxogram	Simulation in NS-3	Yes

Table 9 presents the main parameters of the scenarios used for validation, as well as the performance metrics used in the analyses. All works use from 0 to 300 devices. The arrangement of the devices is generally random, and the communication range is defined by a distance in meters, or by a signal strength threshold. For the latter case, it is used a propagation model with specific parameters (not presented in this comparative table). The performance metrics evaluated are latency, network lifetime, packet delivery ratio, reliability and average characteristics of the generated graphs. Most of the works present the performance results using an average of several experiments or simulations.

Table 9 – Main parameters and performance metrics used for performance evaluation of the routing algorithms

Algorithm	Parameters	Performance metrics
(JINDONG; ZHENJUN; YAOPEI, 2009)	Area: 1000 x 600 m; Communication range: 200 m; Nodes: 0 - 300; Node position: random	Latency; packet loss; bandwidth
(HAN et al., 2011)	Area: 450 x 450 m; Communication range: 25 - 200 m; Nodes: 50 - 150; Node position: random; Failed links: 0 - 95 %	Rate of successful route construction; Percentage of reliable nodes; Broadcast graphs: average number of links per node, number of connected nodes; Downlink graphs: average number of links and nodes per graph; Latency.

Table 9 - continued

Algorithm	Parameters	Performance metrics
(KÜNZEL, 2012)	Area: 450 x 450 m; Communication range: 100 m; Nodes: 150; Battery-powered nodes: 50 %; Gateway position: center.; Node position: random	Average and maximum number of hops of the graph; Percentage of reliable nodes in the graph; Distant nodes (greater than 4 hops); Percentage of routing nodes; Percentage of routing battery-powered nodes.
(MEMON; HONG, 2013)	Area: 250 x 250 m; Communication range: 50 m; Nodes: 100; Node position: uniform; Simulation time: First battery-powered node down.	Energy consumption per node.
(ZHANG; YAN; MA, 2013)	Area: 200 x 200 m; Nodes: 30 - 100; Gateway position: top left corner.	Network lifetime; Average transmission power.
(ZHANG et al., 2014)	Area: 450 x 450 m; Communication range: 100 m; Nodes: 50 - 150.	Number of edges; Number of success transmissions; Average residual energy; Latency; Packet loss ratio.
(WU et al., 2015)	Area: Building floor; Nodes: 63; APs: 2; Channel number: 4, 8, 12, 16 (simulations), 8 (experiments).	Latency max. and min.; Accept ratio.
(WU et al., 2016)	Area: Building floor; Nodes: 63; APs: 2; Channel number: 4, 8, 12, 16 (simulations), 8 (experiments).	Network lifetime; Delivery rate.
(HONG et al., 2015)	Area: 150 x 150 m; Communication range: 50 m;	Energy consumption; Network lifetime.
(SEPULCRE; GOZALVEZ; COLL-PERALES, 2016)	Area: 200 x 200 m; Nodes: 50 - 150; Node position: random;	End-to-end reliability; Latency; Number of redundant routes to reach the desired reliability.
(CAINELLI; KÜNZEL; PEREIRA, 2017)	Area: 450 x 450 m; Communication range: 100 m; Nodes: 150; Battery-powered nodes: 50 %; Gateway position: center.	Average and maximum number of hops of the graph; Percentage of reliable nodes in the graph; Distant nodes (greater than 4 hops); Percentage of routing nodes; Percentage of routing battery-powered nodes; Average network RSL.
(HAN; MA; CHEN, 2019)	Area: 300 x 300 m; Communication range: 100 m; Nodes: 40-130;	Network lifetime; Remaining energy.
(MADDUMA-BANDARAGE, 2020)	Area: 84 - 276 m; Channel number: 1; Nodes: 25 - 50; Battery-powered nodes: 100 %	Average latency; Network lifetime; Energy consumption; Average hops; Reliability; Reachability.

3.1.2 Contributions identified

The following items indicate possible contributions and future works in the area of routing for IWSN:

a) The implementation and evaluation of the state-of-the-art algorithms in a simulation environment that has the whole stack of a standard IWSN protocol, to make experiments and comparisons in similar conditions;

b) To define scenarios (benchmarks) that may represent common applications of IWSN, allowing further works to have a baseline for comparisons. Examples of scenarios are, for example, open areas, offices, and factory floors;

c) The use of data collected over time in a real network to feed a simulated scenario. Data collection could be done, for example, using the passive monitoring tool presented in Künzel (2012). Health reports sent by the devices could be used to access characteristics such as RSL and the packet delivery ration over time, and use this data to feed the simulation;

d) The development of NM architectures that allow the implementation of the state-of-the-art algorithms and the performance comparison;

e) As indicated in the work of Nobre, Silva and Guedes (2015b), to dynamically execute weight adjustments of the routing algorithms dynamically, through heuristics or other methods;

f) To use ML models to create and optimize routes according to the demands of the application. These methods can be used to adjust the weights of the routing algorithms or to choose the routes and neighbors.

3.2 RL APPLIED TO ROUTING IN WIRELESS NETWORKS

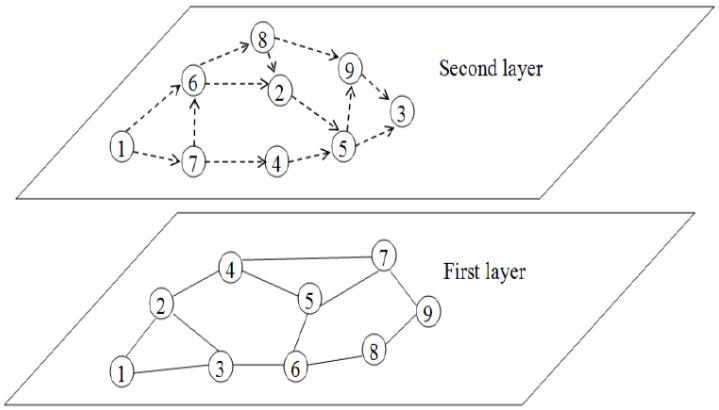
This section presents an analysis of works related to the construction of routes using RL. Relevant works were found for WSN, mobile ad-hoc wireless networks, underwater sensor networks, among others. Section 3.2.1 presents a comparison of the works according to their characteristics, and section 3.2.2 identifies possible contributions in the area.

Al-Rawi, Ng and Yau (2015) review the use of routing RL models in ad-hoc wireless networks, WSNs, cognitive networks, and delay-tolerant networks. Three models relevant to routing in wireless networks are described: Q-routing, Multi-Agent Reinforcement Learning (MARL) and Partial Observable Markow Decision Process (POMDP). In Q-routing, states represent the destination node, while actions represent the next-hop neighbor to forward the message towards the destination. MARL provides global optimization across the network. Agents

learn local information using the traditional RL model and share their rewards with neighbors. It allows nodes to consider their own performance and also of the other nodes. In this way, a global optimization problem is decomposed into a set of locally-solved problems. The POMDP model extends both Q-routing and MARL. In POMDP, a node may not be able to clearly observe its environment. As the state is unknown, the node can estimate the state by incorporating, for example, neighbor parameters such as residual energy, congestion level, and others.

Ye, Zhang and Yang (2015) present a multiagent framework that aims to improve WSN performance regardless of the protocol used. A two-layer architecture is proposed: the first layer represents the complete topology of the network, while the second layer represents a cooperation network between agents, as shown in Figure 13. Each device has an agent who, using local information, decides on the best routes for their messages. Each node in the network builds a cooperation group with some of its neighbors based on their characteristics and the routing transmissions experienced over time with its group.

Figure 13 – Two-layer architecture. Network topology and agent cooperation network.



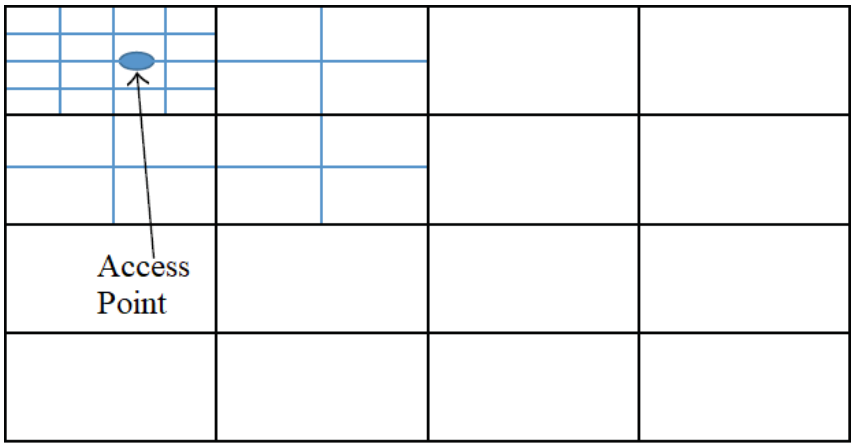
Source – Ye, Zhang and Yang (2015).

The relationships between the agents and their cooperation group are defined considering: the energy consumption, related to the distance and the size of packets being transmitted; storage consumption, related to the quantity and the time that the messages are stored in the transmission buffers and the number of neighbors cooperating; and sensing coverage, which relates the area covered by the sensors to the total area of interest. With these metrics, the authors sought to address the three types of routing approaches mentioned above in a single framework. Agents exchange information and calculate a reward for cooperation. Therefore, each agent is able to decide with which neighbors it cooperates and to which ones it should forward its packets. Different weights can be applied to the three performance metrics, depending on the application. A Q-Learning algorithm with ε-greedy exploration was used in the decision-making of the agents. The use of this learning algorithm allows agents to explore other possibilities for cooperation beyond those that only maximize their reward. The framework was implemented in a simulator developed by the authors in Java language. The performance was compared with traditional

algorithms for each type of routing approach in two scenarios: a static topology, where all devices are already in the topology and do not move; and a dynamic topology, where devices may join and move over time. The comparison evaluated the changes over time in communications latency, packet delivery rate, number of active nodes and total network coverage.

Kiani et al. (2015) propose a protocol that uses RL in WSN to reduce the energy consumption, balance the load and reduce latency. The learning and the route definition are carried out simultaneously in order to reduce the wasted energy in the learning phase, and consequently the communication overhead. Devices close to the AP are grouped into small clusters, as shown in Figure 14, reducing the power consumption on nearby devices when compared to a division into equal-size clusters. The cluster heads are chosen using Q-Learning, where Q-values are determined from an equation that balances the residual energy and the distance from the AP, and the nodes exchange information to know their costs and rewards. The simulations show an increase in packet delivery and lifetime compared to other traditional routing algorithms for WSN.

Figure 14 – Cluster divisions used in Kiani et al. (2015).



Source – Kiani et al. (2015).

In Debowski, Spachos and Areibi (2016) a routing protocol for WSN based on gradients using Q-Learning is presented. The rewards at each node are calculated by taking into account the average number of transmissions performed between a node and the base station as well as the residual energy. Each node maintains a Q-value for itself and sends it to its neighbors. When deciding a neighbor to forward a packet, a node will choose the neighbor that will bring the highest Q-value to itself. Compared with other protocols based on gradients, it was possible to reduce the latency in the communications and increase the network lifetime.

A multiagent Q-Learning-assisted backpressure routing algorithm is presented in Gao et al. (2017). Each node has multiple Q-Learning agents, and each agent continually updates its estimates of route congestion using queue length and congestion information from neighbor nodes. Based on the estimated congestion, each node routes packets through less-congested routes.

As the main advantages of this approach, the authors mention the distributed implementation, low complexity of computation, and optimization of transfer rates.

In Ghaffari (2017), Q-Learning is used in Mobile Ad-Hoc Networks (MANET). MANET consist of a set of mobile nodes, and the routes used for packet transmission are not static. RL is used to predict the node behavior and reduce packet transmission delays. The states of the agents are defined by the destination node, and the actions are defined from the division of the neighbors into groups. The purpose of grouping nodes is to reduce the number of available actions. When selecting actions, the agent first selects a group as a transmitter among the other available groups. The reward is an estimate of the delay and packet transmission success rate that is obtained from the nodes available in the group. After selecting a group that has the best performance in packet transmission, the node with the lowest number of hops is selected as the next neighbor. In this way, the probability of selecting a better performing node is improved. The results obtained in simulation indicate that, over time, delays decrease and packet delivery rates increase.

In Jin et al. (2017), Q-Learning Based Delay-Aware Routing (QDAR) is proposed to extend the lifespan and reduce latency in underwater sensor networks. When a node wants to send a packet to the sink, it first sends a request to the sink to indicate the future transmission. The sink device collects some information and plot a virtual topology between the source node and the sink. The sink then applies the QDAR algorithm to choose the route and sends a virtual packet through this route so that all intermediate nodes are aware of the future transmission. In QDAR, each packet can be viewed as an agent, where states are mapped as the current node where the packet is located, and the actions are the next neighbors to forward the message. The reward is calculated in the sink through cost equations associated with the latency and residual energy of the intermediate nodes, as well as the history of transmissions. The Q-table is maintained in the sink and updated with each new transmission. The QDAR mechanism is adaptive and can be distributed in the dynamic underwater environment.

Tozer, Mazzuchi and Sarkani (2017) use Q-Learning in problems where there are multiple and conflicting objectives for the selection of paths and routes. The authors present an adaptation to the Q-Learning model, known as Voting Q-Learning (VoQL). Q-value vectors store a value for each objective. For a given state, all possible actions are identified as well as the Q-values associated. These vectors are mapped from actions to objectives and then sorted from largest to smallest, as shown in Figure 15. Through this transformation, it is possible to identify which actions are most interesting to be taken for each objective. This information is then used with different voting systems. In the experiments, a robot chooses a path inside an area and has five conflicting objectives, which are presented in Table 10. The results indicate that all the voting methods exceed the compared works in the quality of the solutions found, in the total reward obtained and in the time to define the route.

An agent changes the value of the weight of the power source type of nodes in a cost equation used to build a broadcast graph in (KÜNZEL et al., 2018). States store the current

Figure 15 – Transforming a Q-value vector to actions $a_1, ..., a_4$ and objectives $O_1, ..., O_3$.

$$\begin{array}{l} Q(s,a_1) = [-10, 7, 2] \\ Q(s,a_2) = [-6, 5, 0] \\ Q(s,a_3) = [-5, 6, 0] \\ Q(s,a_4) = [-1, 4, 0] \end{array} \rightarrow \begin{array}{c} O_1 = [-10, -6, -5, -1] \\ O_2 = [7, 5, 6, 4] \\ O_3 = [2, 0, 0, 0] \end{array} \rightarrow \begin{array}{c} O_1 = [a_4, a_3, a_2, a_1] \\ O_2 = [a_1, a_3, a_2, a_4] \\ O_3 = [a_1, a_2 \& a_3 \& a_4] \end{array}$$

Source – Tozer, Mazzuchi and Sarkani (2017).

weight, actions keep or change the state, and rewards are given only when the agent reduces latency and increases lifetime. The use of actions that lead to the same state, and the given rewards, increase the worst-case complexity of the RL problem and require more exploration (KOENIG; SIMMONS, 1992). Also, link quality information is not used to define routes, the simulations do not use an error model in the physical layer, and therefore do not provide proper information about reliability.

In Lu et al. (2020), Q-learning is utilized in a new protocol for underwater sensor networks to learn and adapt to the dynamic environment. Factors such as residual energy, empty spaces and depth difference of sensor nodes are used to calculate the Q-value. The simulation demonstrates that the approach improves the performance in terms of energy efficiency, packet delivery ratio and average network overhead.

Surveys on RL routing approaches for networks and protocols were presented by Mammeri (2019) and Habib, Arafat and Moh (2019), but the centralized approaches described are not related to IWSN and does not consider graph routing.

3.2.1 Comparison of RL approaches for routing in wireless networks

The works described were compared using the following criteria:

a) Centralized approach: if the work addresses the definition of routes in a centralized way, where a single device is responsible to define the routes;

b) Route type: if the work uses graph routing, source routing or defines the next neighbor to forward messages;

c) Reward exchange: if the nodes share information about their current rewards, Q-values or Q-tables;

d) Rewards: the metrics used to calculate the rewards;

e) Performance metrics: information and statistics collected regarding the performance of the approaches, such as latency, packet loss, packet delivery rates, network lifetime, power consumption, rewards, among others.

Table 10 – Comparison of RL applications in routing in wireless networks

Work	Centralized approach	Route type	Reward exchange	Rewards	Performance metrics
(YE; ZHANG; YANG, 2015)	No	Next hop	Yes	Energy and storage consumption, sensor coverage	Latency, delivery ration, number of active nodes, coverage area
(KIANI et al., 2015)	No	Next hop	Yes	Residual energy, degree	Lifetime, delivery ratio, latency, load balance
(DEBOWSKI; SPACHOS; AREIBI, 2016)	No	Next hop	Yes	Residual energy and transmission number	Minimum battery level, average latency
(GAO et al., 2017)	No	Next hop	No	Packet queue size, neighbor congestion levels	Latency
(GHAFFARI, 2017)	No	Next hop (in a group)	No	Latency, delivery ratio	Average latency, delivery ratio
(JIN et al., 2017)	Yes	Next hop	No	Average latency, residual energy	Network lifetime, total energy consumption, average latency
(TOZER; MAZ-ZUCHI; SARKANI, 2017)	Yes	Next quadrant (hop)	No	Distance, signal loss, travel time, energy used, adversary avoidance	Total rewards, episode time
(KÜNZEL et al., 2018)	Yes	Broadcast, uplink graph	No	Average network latency, expected network lifetime	Network latency, network lifetime
(LU et al., 2020)	No	Next hop	Yes	Residual energy, voids and depth	Energy consumption, packet delivery, delay, overhead

The comparison shows that most of the works are decentralized, where each node has a learning agent responsible for choosing the routes to be used. Generally, states represent a recipient and actions the next-hop neighbor. In the centralized approaches, the agent knows the topology, chooses the routes, and distributes them to be used by the nodes. None of the works

creates path-redundant graphs or routes. In some works, the agents share their Q-tables and rewards with the neighbors, similar to the MARL approach. Regarding rewards, most of the works reward agents when they reduce energy consumption, latency, and network congestion. The performance metrics involve latency, network lifetime, packet delivery ratio, similar to the works presented in section 3.1.

To the best of our knowledge, the current decentralized approaches are not suitable for centralized IWSN, since they require nodes to choose routes independently. Also, the available centralized approaches are used for other communication technologies or do not build uplink graphs suitable for IWSN applications.

3.2.2 Contributions identified

The following contributions were identified for RL approaches for routing in IWSN:

a) To use of RL for the construction of uplink, broadcast and downlink graphs in a centralized manner, allowing their application in IWSN protocols;

b) To develop approaches with RL that can build graphs with path redundancy, increasing network reliability;

c) To develop RL approaches to adjust weights of the state-of-the-art routing algorithms;

d) To develop case studies that involve the evaluation of these approaches in IWSN protocols, in applications such as process monitoring and control;

e) To discuss expected results, considerations that must me taken, reward, and the possible limitations of RL approaches in a protocol such as WH;

f) To evaluate and compare the performance of these approaches with the state-of-the-art routing algorithms presented in section 3.1.1.

4 THE Q-LEARNING RELIABLE ROUTING APPROACHES

The main contribution identified in the state-of-the-art analysis involves the development of approaches to build reliable graphs in a centralized fashion using RL models such as Q-Learning, trying to enhance or balance the IWSN performance. Other relevant contributions are: the discussion of the aspects that impact the use of RL for graph routing in IWSN; the comparison of these new approaches with the state-of-the-art graph routing algorithms; and the use of a simulation environment that provides a complete IWSN protocol stack for performance evaluation.

To give detailed information about how the contributions are achieved, this thesis is divided into two chapters. This chapter details two novel routing algorithms. Section 4.1 presents the scope, definitions and overall characteristics of the approaches developed. Sections 4.2 and 4.3 describe two algorithms to build the uplink graphs using Q-Learning. Section 4.4 discusses the use of the QLRR approaches in IWSN. Chapter 5 presents the improvements in the WirelessHART simulator, simulation parameters and scenarios, details of the performance evaluation methodology, and discussion of the results.

4.1 SCOPE AND DEFINITIONS

It is considered that in centralized IWSN protocols such as WH and ISA SP100.11a, the management routines are executed as the sequence previously presented in Figure 12. The NM runs the management routines when a node joins or leaves the network, when the topology changes (based on reports and alarms sent by nodes), or periodically for optimization. The NM maintains a complete representation of the network topology as well as information about the network nodes and operation conditions (HAN et al., 2011; ZAND et al., 2014b)

The NM routines have four basic steps:

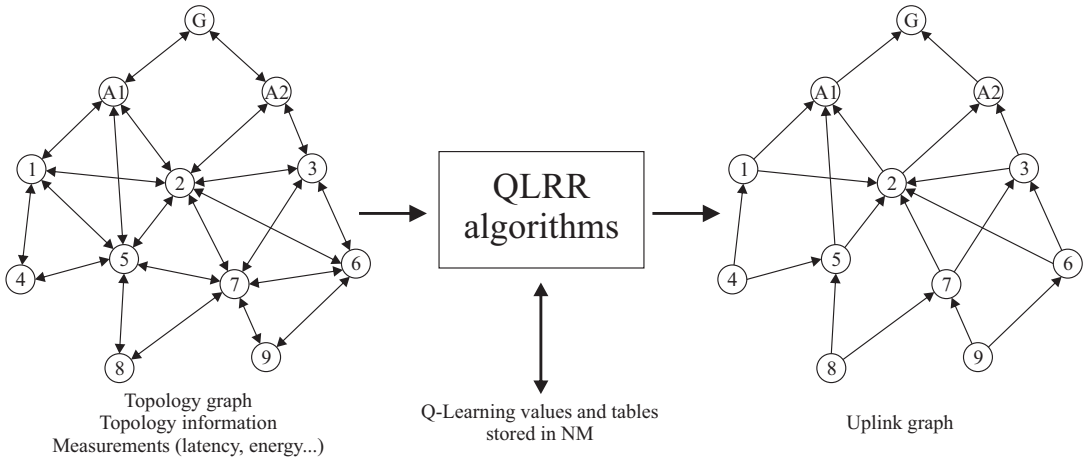
- to build the routes that will be used for data transmission based on the current network topology;
- to build the data communication schedule;
- to compare the new routes and schedule with the ones currently being used in the network;
- and to send commands through the network to update routes and schedule.

This thesis is focused on the construction of the uplink graph that will be used by nodes to send sensor readings towards the gateway in IWSN monitoring applications. The approach can be further developed for the construction of the broadcast and downlink graphs.

The proposed algorithms will be named Q-Learning Reliable Routing (QLRR). The QLRR algorithms execute as a function that receives as input a graph $G(V, E)$ that contains the complete network topology. Set V contains the vertexes that represent the devices in the network (gateway g, the set of access points V_{AP}, and the nodes). Set E contains the edges representing the connections available between devices.

Figure 16 depicts the QLRR data flow used to build the uplink graphs. Updated information about the topology graph, nodes and neighbors, latency, lifetime and Q-Learning tables are available to the QLRR algorithms.

Figure 16 – QLRR data flow used to build the uplink graph



Source – The author.

At the end of the execution, QLRR returns the uplink graph $G_U(V_U, E_U)$, where V_U is the set of nodes added during the uplink graph construction, and E_U is a subset of E with edges connecting the nodes towards the gateway. QLRR is a greedy algorithm, which means that G_U is built iteratively. At each iteration, QLRR adds a node and the selected edges with neighbors to G_U, until $V_U = V$.

The connection between g and V_{AP} is considered wired and therefore not prone to transmission failures (NOBRE; SILVA; GUEDES, 2015b). It is assumed that the graph $G(V, E)$ is connected, that is, all nodes have at least one edge with a neighbor. Nodes disconnected from the topology must be removed from G prior running the QLRR algorithms because they represent devices that are still joining or are no longer available in the network. It is considered that the network has a fixed topology during the operation, since IWSN topologies are typically planned for each application.

As IWSN are subject to different conditions of the wireless channel, it is considered a

general path loss model for RSL estimation and a packet loss probability associated with the RSL. Nodes can inform the NM about poor connections with any linked neighbors through path down alarms. NM permanently removes these connections from the network topology, and do not allow the QLRR algorithms to use them anymore. The discussion of the use of other loss models and the processing of path down alarms is beyond the scope of this thesis.

A percentage of nodes is considered to be powered by batteries, while the others are line powered. Battery-powered nodes can estimate their battery lifetime, and they start the simulations with the battery level at 100 %. It is considered that no nodes will be powered down during the simulation because IWSN nodes typically use batteries that are intended to provide an expected lifetime up to ten years (CHEN; NIXON; MOK, 2010).

The exploration phase of the QLRR algorithms starts at the beginning of each simulation and ends after a given period of simulation time. During the exploration phase, the NM will make several reconfigurations over the network. No exploration will occur during the last hours of simulation, allowing the network to stabilize in a certain operation condition that will allow the measurement of the performance metrics used for comparison.

A description of the main characteristics and objectives is made for each QLRR approach. The execution sequence for the construction of G_U, the mapping of the Q-Learning states and actions and the rewards given to the learning agents are also presented.

4.1.1 Metrics used for performance evaluation

QLRR is evaluated considering three requirements of IWSN applications: low latency, low energy consumption, and reliable communications (SHA et al., 2017; NOBRE; SILVA; GUEDES, 2015b). They were chosen because they are the most relevant performance measurements identified in the literature review when considering IWSN monitoring applications. The metrics are described in the following subsections because of the necessity to explain some details of the algorithms and rewards.

4.1.1.1 Average Network Latency (ANL)

The latency of a data packet is defined here as the time between the generation of a data packet at the sensor's Network Layer and the reception at the gateway's Medium Access Control layer (CHEN; NIXON; MOK, 2010).

To measure the Average Network Latency, the NM stores the latency of all data packets received from all nodes at the gateway over the last time interval t_s. The current ANL is denoted by d_{t+1} and is obtained from the average value of the samples collected over t_s. An array D is used to store the last k measurements of d. The D array keeps a trace of the ANL over time and can be used to give rewards based on the network's historical information.

The average latency of a specific node or agent v is calculated in the same way as the ANL, but considering only the latency of the data packets sent by v to the gateway over t_s. The current average latency of v is denoted by d^v_{t+1}. An array D^v also keeps the last k measurements of d^v.

4.1.1.2 Expected Network Lifetime (ENL)

The Expected Network Lifetime is defined as the minimum expected lifetime value between all battery-powered nodes (WU et al., 2016). For the measurement of the ENL, NM periodically requests the current lifetime expectation of all battery-powered nodes. The current ENL is denoted by l_{t+1}, and NM stores in an array L the last k measurements of l.

The battery lifetime of a node is expected to reduce at each measurement because of the energy consumption pattern of the nodes, but it is assumed that the battery life is reported through an integer value that represents days and does not reduce significantly from one measurement to another (HCF, 2008c).

4.1.1.3 Packet Delivery Ratio (PDR)

The Packet Delivery Ratio is defined as the ratio between all data packets generated at the sensors and those effectively received at the gateway (WU et al., 2016). Packets may be discarded at the MAC layer of a node after several retransmission retries from a node to the next neighbor (ZAND et al., 2014b).

4.1.1.4 Percentage of Reliable Nodes (PRN)

The Percentage of Reliable Nodes is calculated as a ratio between the number of nodes that have at least two neighbors to forward data on the uplink graph and the total number of nodes on the uplink graph (HAN et al., 2011).

4.2 Q-LEARNING RELIABLE ROUTING WITH A WEIGHTING AGENT

In this approach, called Q-Learning Reliable Routing with a Weighting Agent (QLRR-WA), a single learning agent is used to adjust the set of weights of a state-of-the-art routing algorithm that builds the uplink graph. The weights are related to a cost equation used to define how nodes and successors will be selected during the uplink graph construction. The agent will act globally, searching for a set of weights that optimizes the overall performance of the network. The agent takes actions that increase or decrease the weights at each execution of the QLRR-WA algorithm, measures the performance metrics and receives a reward.

In Section 4.2.1, it is explained how G_U is built and how the weights influence the construction of the uplink graph. In Section 4.2.2 it is described how the Q-Learning model is applied to adjust the weights.

4.2.1 QLRR-WA Uplink Graph Construction

QLRR-WA uses the routing algorithm in Künzel, Cainelli and Pereira (2017) as a baseline, which is a greedy algorithm that builds reliable broadcast and uplink graphs. During the uplink graph construction, nodes and edges with successors are iteratively added to G_U and selected through a cost equation. The pseudo-algorithm of QLRR-WA is presented in Algorithm 2. Further details of the execution sequence, pseudoalgorithm, and related equations can be found in Han et al. (2011), Künzel, Cainelli and Pereira (2017).

Algorithm 2: Q-Learning Reliable Routing with a Weighting Agent

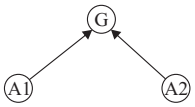
Input: $G(V, E)$ // topology graph
Output: $G_U(V_U, E_U)$ // uplink graph

Calculate reward r_{t+1} according to Eq. 4.3
Update $Q_{t+1}(s_t, a_t)$ according to Eq. 2.1
Select action $a_{t+1} \in A$ using ε-greedy
Take action a_{t+1}, changing the weights according to the new state s_{t+1}
$V_U = g \cup V_{AP}$ and E_U contains all edges from V_{AP} to g.
while $V_U \neq V$ **do**
    Find $S' \subseteq V - V_U : \forall v \in S'$, v has at least two outgoing edges to V_U
    if $S' \neq \emptyset$ **then**
        forall $v \in S'$ **do**
            Store in E_v the outgoing edges to V_U
            Store in U_v the destination vertexes of E_v
            Sort U_v according to Eq. 4.1
            Choose e_{v,u_1} and e_{v,u_2} from U_v
            $h_v = 1 + \frac{h_{u_1} + h_{u_2}}{2}$
        end
        Find $v \in S'$ with smaller c according to Eq. 4.1
        $V_U \subseteq V_U \cup v$ and $E_U \subseteq E_U \cup e_{v,u_1} \cup e_{v,u_2}$
    end
    else
        Find $S'' \subseteq V - V_U : \forall v \in S''$, v has one outgoing edge to V_U
        forall $v \in S''$ **do**
            $h_v = h_{u_1} + 1$
            Determine n_v, the number of ingoing edges from $V - V_U$ to v
        end
        Find $v \in S''$ with smaller cost according to Eq. 4.2.
        $V_U \subseteq V_U \cup v$ and $E_U \subseteq E_U \cup e_{v,u_1}$
    end
end
Return G_U

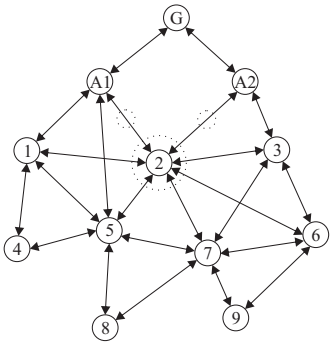
In Line 5, g, V_{AP} and the edges from the APs to the gateway are added to G_U, as depicted in Fig. 17a. The algorithm looks for candidate nodes in $V - V_U$ to be inserted in G_U until all nodes in V are added to V_U in Line 6. The algorithm adds first nodes considered reliable, which have at least two possible neighbors (successors) in G_U, as exemplified in the sequence of Fig. 17b-17f. These reliable nodes are identified in Line 7. The cost c of the candidate nodes and their possible successors are calculated using Equation 4.1.

Figure 17 – Construction sequence of G_U in QLRR-WA.

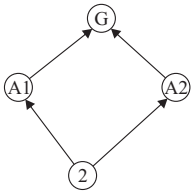
(a) Initial $G_U(V_U, E_U)$



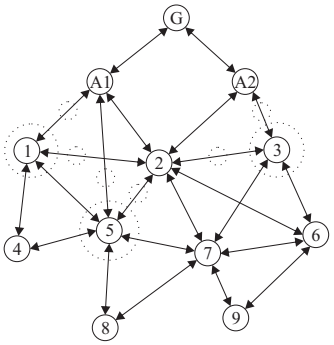
(b) $G(V, E)$ and a reliable candidate v



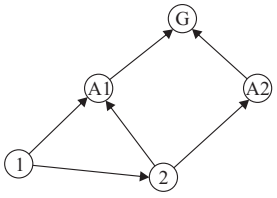
(c) Insertion of v and edges in $G_U(V_U, E_U)$



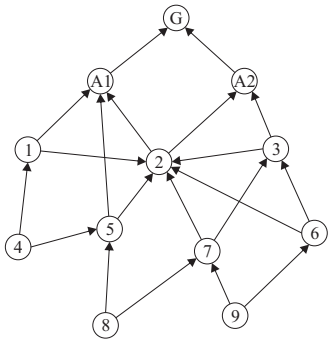
(d) Candidates and edges on the 2nd iteration



(e) Insertion of the lowest cost node and edges in $G_U(V_U, E_U)$ in 2nd iteration



(f) Finished G_U



Source – Adapted from Künzel, Cainelli and Pereira (2017).

The costs of the possible successors of a candidate node are calculated in Line 8. When evaluating a successor, the parameters of Equation 4.1 are considered as follows. The average number of hops h of a successor u is given by the average hops of its successors in G_U plus 1 (HAN et al., 2011). h_{max} stores the largest value of h of all the successors of a candidate node. p is a constant value that is associated with the energy source type of the successor. s is the RSL

value of the edge between the candidate node and the successor. s_d is a constant value which gives a desirable level for the RSL. When all successors have their costs evaluated, they are sorted and then the two lower-cost successors u_1 and u_2 are chosen for the candidate node in Line 13. The average number of hops of node v is then calculated in Line 14.

$$c = w_h \frac{h}{h_{max}} + w_p p + w_s \min\left(\frac{s}{s_d} - 1, 0\right) \tag{4.1}$$

Then, the candidate nodes are evaluated to select one of them to be added to G_U with the edges to its selected successors as described in Line 16. For the evaluation of a candidate node v, h is given by h_v, h_{max} is the maximum number of hops of the candidate nodes, p is associated with the energy source type of the node, and s is given by the average RSL of the edges with its selected successors. The lowest cost candidate node is then added with the selected edges in Line 17.

By changing the values of the weights w_h, w_p and w_s, it is possible to define how the topology and node characteristics will influence the costs and the connections established: increasing w_h will reduce the distance in hops from nodes to the gateway and thus the use of communication resources (HAN et al., 2011); increasing w_p will cause nodes to avoid forwarding data to battery-powered nodes; increasing w_s will make nodes connect to successors with greater RSL, thus reducing the probability of packet transmission failures.

If a reliable candidate is not found, the algorithm then identifies the candidates with a single successor towards the current G_U in Line 20. The costs of the candidates are calculated using Equation 4.2, which will assign lower costs to candidates that do not have energy restrictions and increase the probability of finding a higher number of reliable nodes in the next iteration. In this equation, n is the number of edges that the nodes have with $V - V_U$, which indicates how many nodes may connect to the candidate in the next iteration. n_{max} is the highest value for n for all candidates. p is the power restriction of the candidate.

$$c = w_n\left(1 - \frac{n}{n_{max}}\right) + w_p p \tag{4.2}$$

By increasing the values of the weight w_n, it is possible to choose candidates that will have a greater number of connections with the nodes in $V - V_U$. By increasing w_p, battery-powered candidates will be avoided, so as the use of battery-powered nodes to act as routers. In this thesis, the weights of Equation 4.2 were fixed. The agent will focus only on adjusting the weights of Equation 4.1.

4.2.2 Q-Learning and the Weighting Agent

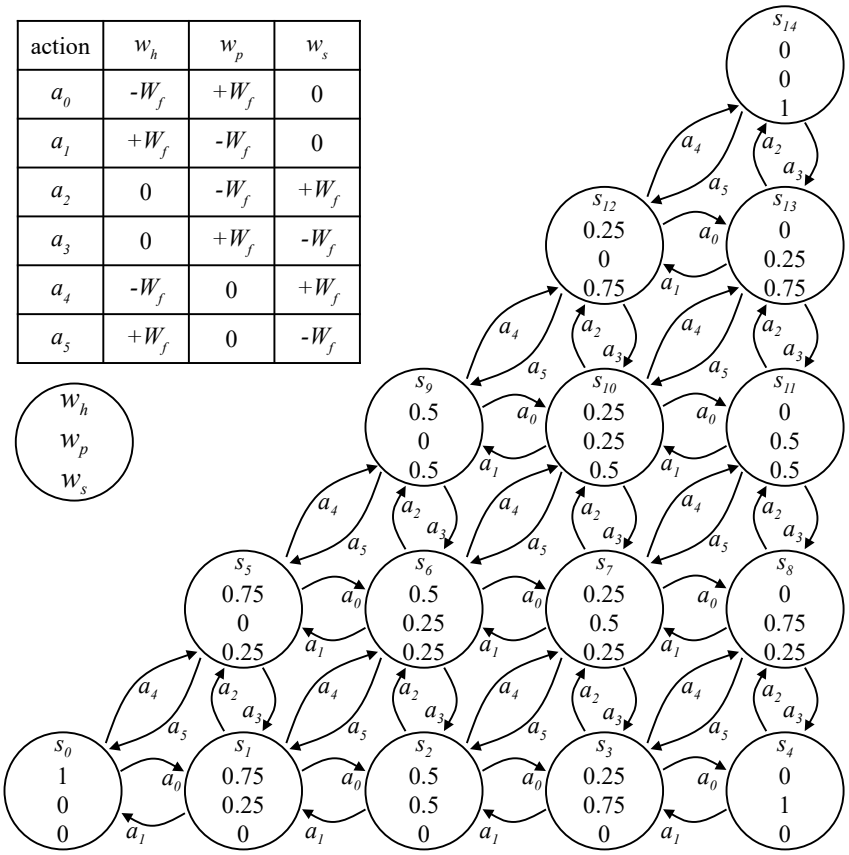
A set of states was defined for the Q-Learning model, where each state has a fixed set of values for w_h, w_p and w_s, and $w_h + w_p + w_s = 1$ on each state. N_w is the set of weights being

used in the cost equation. For our approach, $|N_w| = 3$ because $N_w = w_h \cup w_p \cup w_s$.

A weight factor $0 \leq W_f \leq 1$ defines how much the value of the weights may change from state to state and is given by $W_f = \frac{1}{M}$, where M is an integer number that represents how many transitions between values each weight will have in the model. Actions represent the increment or reduction of the weights from one state to another, but actions available in one state allow transitions only for states where the values of the weights change at maximum $\pm W_f$. Fig. 18 depicts an example of the states, actions, and weights when $M = 4$ and $|N_w| = 3$ and Table 11 presents the number of states and actions of the model for different values of M and $|N_w|$.

Figure 18 – Actions and states when $M = 4$, $|N_w| = 3$

action	w_h	w_p	w_s
a_0	$-W_f$	$+W_f$	0
a_1	$+W_f$	$-W_f$	0
a_2	0	$-W_f$	$+W_f$
a_3	0	$+W_f$	$-W_f$
a_4	$-W_f$	0	$+W_f$
a_5	$+W_f$	0	$-W_f$



Source – The author.

A higher value of M increases the possibility to find an optimal set of weights. However, a lower value of M reduces the number of states and thus the exploration time, because it reduces the number of iterations required for the learning process and the worst-case complexity of the RL problem (KOENIG; SIMMONS, 1992). Another concern is that the costs of the candidates will change abruptly from one state to another when M has a lower value, thus increasing the number of changes in the connections in G_U from one execution of QLRR-WA to another.

Table 11 – Number of states and actions for different values of $|N_w|$ and M.

$\|N_w\|$	M	W_f	$\|S\|$	$\|A\|$
2	2	0.50	3	2
2	3	0.33	4	2
2	4	0.25	5	2
3	2	0.50	6	6
3	3	0.33	10	6
3	4	0.25	15	6

Source – The author.

4.2.3 Reward calculation

Every time the QLRR-WA algorithm is executed, it first calculates the rewards in Line 1. The rewards given to the agent in this approach have as main objectives to reduce the ANL of process data sent from sensors to the gateway and to increase the ENL of the network. These two metrics are the most used for evaluating the performance of the routing algorithms presented in section 3.1. To determine the reward, it is necessary to measure the ANL and ENL at each execution of the QLRR-WA during the NM management tasks.

Equation 4.3 describes the rewards given to the agent. A positive reward of value R is given if the ANL has decreased and the ENL has increased in comparison with $\min(D)$ and $\min(L)$; a positive reward of value $R/2$ is given if ANL has decreased or ENL has increased; no reward otherwise. This reward will cause the agent to explore the action-state pairs and discover a state that it should go to increase its rewards.

$$r_{t+1} = \begin{cases} R, & \text{if } l_{t+1} > \min(L) \text{ and } d_{t+1} < \min(D) \\ \frac{R}{2}, & \text{if } l_{t+1} > \min(L) \text{ or } d_{t+1} < \min(D) \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

In Line 2 of Algorithm 2, the Q-values regarding the last action and state are updated. Then, a new action is chosen in Line 3 and the action is taken on Line 4, changing the current state. The values of the weights are updated according to the weight values of the new state.

4.3 Q-LEARNING RELIABLE ROUTING WITH MULTIPLE AGENTS

In this approach, called Q-Learning Reliable Routing with Multiple Agents (QLRR-MA), each node v has a learning agent. During the construction of G_U, each agent performs actions that define the successors that will be used to send messages towards the gateway.

4.3.1 States and actions mapping

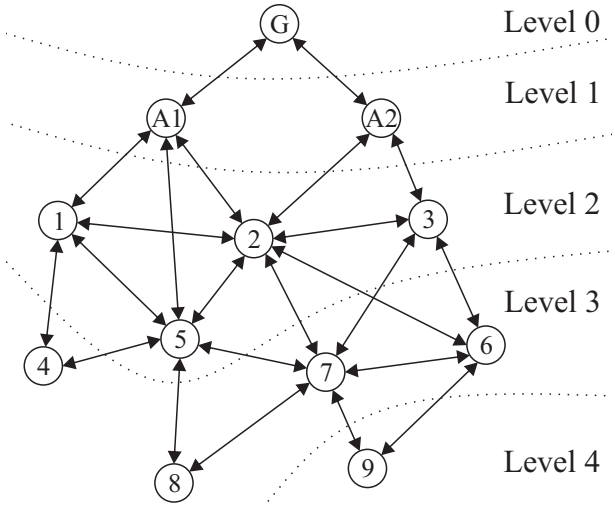
The states and actions are based on the Q-routing model. In Q-routing, each destination of a message corresponds to a state and actions represent the successor chosen to send the message. However, QLRR-MA has differences regarding Q-routing: as the focus is the construction of G_U, the state $s^v \in S^v$ of a node will always represent the gateway destination; each action $a^v \in A^v$ has a pair of successors u_1 and u_2; each agent selects an action $a^v \in A^v$ and adds the node v and the edges e_{v,u_1} and e_{v,u_2} of the selected action to G_U; some actions may be enabled or disabled; finally, the approach is centralized, returning G_U at the end of the algorithm execution. These characteristics will be clarified in the following sections.

4.3.2 Uplink graph construction

The QLRR-MA pseudocode is presented in Algorithm 3. This algorithm was based on some aspects of the ELHFR and Han algorithms (JINDONG; ZHENJUN; YAOPEI, 2009; HAN et al., 2011). NM keeps a global Q-table $Q(S, A)$, which has all Q-values, states, and actions currently stored for all agents.

In Line 1, the algorithm performs a search using BFS to identify the level of each node v in the topology. The BFS level represents the minimum number of hops h_v from v to g. This information will be needed to define to which neighbors each node can connect and the set of actions A^v available in the later steps of the algorithm. Figure 19 depicts the levels for the example topology.

Figure 19 – BFS levels for the example topology



Source – The author.

In Line 2, the algorithm adds into G_U the gateway g, the V_{AP} and the edges from V_{AP} to g. This step is similar to the Han algorithm and is depicted in Figure 17a. Then, the algorithm

enters a loop that will insert all the nodes present in V to V_U and the edges are chosen by the agents to E_U.

Algorithm 3: QLRR-MA uplink graph construction

Input: $G(V, E)$ // Topology graph
Input: $Q(S, A)$ // Q-table of all agents
Output: $G_U(V_U, E_U)$ // Uplink graph

1 $\forall v \in V$, store in h_v its level using BFS
2 $V_U = g \cup V_{AP}$ and E_U has the edges from V_{AP} to g
3 **while** $V_U \neq V$ **do**
4 **forall** $v \in V - V_U$ **do**
5 Find $U_v \subseteq V : \forall u \in V, \exists e_{v,u} \in E$ **and** $h_u \leq h_v$ **and** u has at least two outgoing edges to $V - v$ with level $\leq h_u$ **and** $e_{v,u}$ don't create cycles in G_U if added
6 Sort all successors in U_v by power source type cost p
7 **if** $|U_v| > 2$ **then**
8 Calculate reward r^v_{t+1}
9 Update $Q^v_{t+1}(s^v_t, a^v_t)$ according to Equation 2.1
10 Disable all actions in A^v
11 Create actions related to the successor pairs combinations in U_v not yet in A_v
12 Enable the actions in A^v related to the successor pairs combinations in U_v, limiting to N_u the number of enabled actions
13 Select next action a^v between the enabled ones in A^v using ε-greedy
14 $V_U \subseteq V_U \cup v$ **and** $E_U \subseteq E_U \cup e_{v,a^v_{u_1}} \cup e_{v,a^v_{u_2}}$
15 **end**
16 **else**
17 $V_U \subseteq V_U \cup U_v$ **and** $E_U \subseteq E_U \cup e_{v,u} \forall u \in U_v$
18 **end**
19 **end**
20 **end**

The neighbors U_v, which are potential successors of v to send their messages towards g are identified in Line 5. Some rules have been defined so that a neighbor u can be considered a possible successor for v:

a) There must be an edge in E from v to u, otherwise it is not possible to use u as a successor;

b) The BFS level of u must be equal to or less than the level of v, avoiding messages to be sent to higher levels, farther from g;

c) u must have at least two edges towards $V - v$ in G, and the vertices of these edges must have a level less than or equal to h_u. This is because, if u is added in a later iteration, it will not be able to guarantee the path redundancy, thus reducing the reliability of G_U;

d) If $h_u = h_v$, it is necessary to check if cycles will be created at this level if the edge $e_{v,u}$ is inserted into G_U. In some cases, the neighbor u or some of its successors may be sending messages to v, then closing a cycle that can cause the propagation of messages indefinitely

in the network. It is also avoided, along with Rule b, that all devices connect to neighbors only of the same level, and that the graph G_U is disconnected or does not have as final destination g.

Once all the potential successors U_v are identified, they are sorted by their power source cost p on Line 6. Line-powered successors may have attributed a lower cost value, while battery-powered may have a higher cost. When $|U_v|$ is high, this will allow node v to avoid the battery-powered successors.

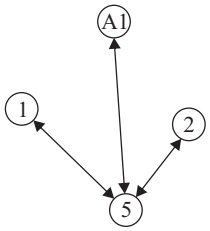
If $|U_v| > 2$, the algorithm begins to perform the steps related to the agent of node v. If $|U_v| <= 2$, this indicates that only one action is available for node v, and can not explore other possibilities of connections.

In Line 7, the reward r^v_{t+1}, corresponding to the execution of the last action at the instant t, is calculated. In Line 8, the Q-table is updated according to the equation 2.1.

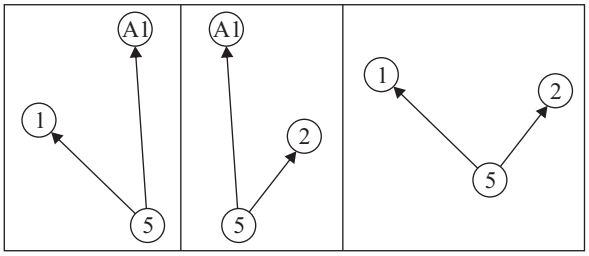
Each pair of neighbors in the set U_v will be combined to create an action. Each action will have two components: The two successors (u_1 and u_2) and an action condition (enabled or disabled). To clarify, Figure 20a depicts a possible set U_v for node 5 and Figure 20b the possible actions arising from the U_v set.

Figure 20 – Example of set U_v for node 5 and actions available

(a) Possible neighbors for node 5 ($U_5 = AP1 \cup 1 \cup 2$)



(b) Available actions for node 5



Source – The author.

Line 10 disables all actions A^v available in the state s^v. If an action for a specific pair of successors does not exist yet in A^v, it must be created in Line 11. Then, in Line 12 all actions available relative to the neighbors in U_v and existing in A^v are enabled. This activation control has two main objectives: first, it prevents node v from choosing actions that are not allowed due to changes in the topology or because the action has a successor that doesn't respect the rules

stated in Line 5; second, the number of available actions for a specific node at a given instant t depends on $|U_v|$. In highly connected topologies the number of enabled actions for a node can be limited to N_u. As U_v is sorted by p on Line 6, the actions related to successors with lower power costs will be activated first.

Then, on Line 13, the agent v selects the next action a^v_{t+1}. On Line 14 it inserts the v node in V_U and the edges for the neighbors that belong to a^v_{t+1} in E_U. Finally, in Line 17, if $|U_v| <= 2$, the node v will be connected to the one or two successors available. In this case, the learning agent is not executed.

4.3.3 Reward calculation

QLRR-MA will try to reduce the average latency of communications in the network by reducing the average latency of each node. Latency measurement is performed at each iteration of the Q-Learning algorithm, but the reward of each v agent is given by measuring the average latency of its process data only. The reward r^v_{t+1} is proposed as follows:

$$r^v_{t+1} = \begin{cases} R, & \text{if } d^v_{t+1} < \min(D^v) \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

This reward makes each node explore the available actions, and finds one where it can reduce its latency.

4.4 USING QLRR IN IWSN

The characteristics of the application and the selected communication protocol must be analysed when using QLRR. G_U may change at each execution of QLRR during the exploration phase, and the NM must send reconfiguration commands over the network. In the current IWSN protocols, the reconfiguration time will take from seconds to tens of minutes. The reconfiguration time must be considered to define the time interval between the actions of the agent.

Fluctuations in the latency will occur during the reconfiguration because of the following reasons: the overhead caused by messages with commands; the commands may have priority over data process messages (SHEN et al., 2014); the number of hops that a message takes to reach the gateway may change; the number of messages waiting in the transmission stacks of the nodes may increase; the Q-Learning parameters used; and the exploration randomness.

Several data process messages should be received to measure the ANL, considering these fluctuations and the wireless characteristics. The time interval between the iterations of the agent should allow the network to reconfigure correctly and the measurements to represent stabilized conditions for the data process messages.

The reconfiguration will also reduce the ENL because of the energy spent with overhead, and this influence will be more significant when low-capacity batteries are used. At the same time, battery-powered nodes closer to the gateway will have their batteries depleted first, as they may work as routers.

Another observation regarding the current topology is that the changes in the topology are usually informed by health reports and path down alarms, thus influencing the time needed for detecting changes (CHEN; NIXON; MOK, 2010).

The fluctuation in ANL caused by exploration may be tolerable only during the network's startup and maintenance, but not during process monitoring and control. In applications where variations in the ANL are tolerable, the agent can continuously explore. When using ε-greedy to control exploration, it is possible to stop the exploration by changing the ε value to 0 during execution.

The complexity of the learning model also depends on the number of states, actions, and rewards given (SUTTON; BARTO, 2018; KOENIG; SIMMONS, 1992). The agent should explore several times the available actions in each state. Learning, therefore, may require many iterations to converge (MAMMERI, 2019; WANG; CHAI; WONG, 2016; KOENIG; SIMMONS, 1992). When using QLRR-WA, the value of M allows a trade-off between reducing the need for exploration and avoiding abrupt changes in the weights, consequently reducing the number of significant changes in G_U from state to state. When using QLRR-MA, the value of N_u can limit the number of available actions for each node.

The choice of the reward function also influences the behavioral pattern of the agent and must be defined based on the metrics that want to be enhanced (SUTTON; BARTO, 2018; AL-RAWI; NG; YAU, 2015; WANG; CHAI; WONG, 2016).

The actions of the agent will also be associated with the schedule changes because the changes in G_U will cause different reconfiguration patterns on the timeslots depending on the scheduling algorithms. The scheduling algorithms will also influence performance because of the strategy used for the allocation of timeslots. Thus, a proper combination of routing and scheduling algorithms is needed to enhance the performance, as indicated by (NOBRE; SILVA; GUEDES, 2015b).

5 QLRR PERFORMANCE EVALUATION

This chapter discusses details of the performance evaluation and the results. Section 5.1 presents the changes of the WirelessHART simulator used. Section 5.2 describes the performance evaluation methodology and Section 5.3 the simulation parameters used. Section 5.4 presents the results for the 20 and 40-node topology examples depicted in Figure 22. Section 5.5 presents the general results for several random topologies.

5.1 IWSN PROTOCOL AND SIMULATION ENVIRONMENT

The WirelessHART protocol was chosen for the development of the simulations, taking into account the following criteria: this protocol is widely known and used in industrial applications; the experience of the Graduate Program in Electrical Engineering (PPGEE) group of Federal University of Rio Grande do Sul (UFRGS), recognized through publications on various aspects of the protocol; the availability of official documentation, equipment for experiments and data collection at UFRGS's laboratories, and open software tools for simulation.

The simulator of Zand et al. (2014b) version 9.4.1 was chosen to conduct the performance evaluations. The main advantages of this simulator are: the complete implementation of the WirelessHART stack; the availability of source codes, documentation, examples, scenarios; the use of object-oriented language; the validation compared to real experiments; the academic relevance; the complete implementation of the routing and scheduling algorithms of Han et al. (2011); and the ease of implementation of new algorithms.

However, as pointed out by Nobre, Silva and Guedes (2014), this simulator lacks more realistic models for power consumption and transmission errors. The initial simulations performed during the thesis development also identified that some relevant application layer commands were not implemented, while others were partially implemented according to the standard, as discussed in section 2.5.

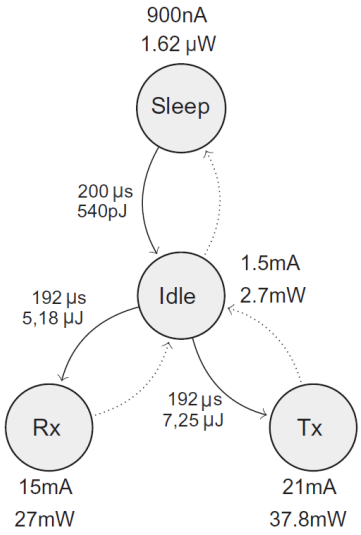
Changes and adaptations were proposed in the simulator in order to perform simulations that provide a better representation of an operational WH network. Some features have been added for the execution of the routing algorithms and for the automated generation of reports. These adaptations and improvements in the simulator are other contributions of this thesis since they will allow subsequent studies and performance analysis over different aspects of the protocol. The main changes in the simulator are described in the following subsections, according to the layers of the protocol where they were implemented.

5.1.1 Data-link Layer

5.1.1.1 Energy consumption model

It was necessary to adapt the simulator to the use of an energy model that allows a more accurate comparison and evaluation of the power consumption of the devices. The new model is based on the work of Nobre, Silva and Guedes (2015a), where the power consumption of the CC2500 radio transceiver was considered. This radio has four operating states: Sleep, when the radio is with its clock off and waiting to be turned on; Idle, when the clock is active; Tx, when the radio is transmitting; and Rx when the radio is receiving data. The possible state transitions are shown in Figure 21, as well as the power consumption in each state and during the transitions considering a 1.8 V battery.

Figure 21 – CC2500 energy model



Source – Adapted from Nobre, Silva and Guedes (2015a).

The time spent on each state is influenced directly by the data-link layer, more precisely by the scheduling algorithm. At each timeslot, a device may be transmitting or receiving a packet, or if it has no allocated task, the device remains in Sleep mode. When transmitting or receiving a packet, transitions between states occur according to the timeslot time structure, previously shown in Figure 7.

This model has some limitations and unclear aspects related to the size of the packets and the destination of a packet (unicast or broadcast). They influence some of the state transitions of the radio:

- The model assumes that whenever there is a transmission, the packet's source node will send a packet of data equal to 90 bytes. In practice, data packets may vary in size;
- The model assumes that even for received broadcast packets, a node will send an ACK;

- The size considered for an ACK packet is 9 bytes in the model while is 26 bytes in the standard;

Considering these limitations, the following improvements are proposed:

- The calculation of energy consumption in a timeslot, for the transmitter, takes into account the size of the packet;
- The calculation of power consumption in a timeslot, for the transmitter, takes into account if the packet is of the broadcast type. In this case, after sending the data, the radio goes to the Sleep state. Otherwise, it waits in the Idle state and the then transitions to the Rx state and waits for the ACK to be received;
- The calculation of power consumption in a timeslot, for the receiver, takes into account the size of the received packet;
- The energy consumption calculation in a timeslot, for the receiver, takes into account the size of the ACK defined in the standard;
- The calculation of the power consumption in the receiver takes into account if the packet is of the broadcast type. In this case, after receiving the data, the radio goes to the Sleep state.

5.1.1.2 Battery Lifetime Calculation

The data-link layer of the nodes was adapted to allow them to estimate the battery lifetime. Each node monitors its activity on each timeslot and calculates the total energy consumed during that timeslot. With this information, the battery life is estimated as follows: Every time the node receives a Read Battery Life command (778) from the of NM, it calculates the amount of energy consumed since the last command received. Using the time interval between the commands, the amount of energy spent in this time interval and the residual energy of the battery, it can determine the expected battery lifetime and respond to the NM.

The WH standard does not define any method for estimating power consumption and only indicates that the manufacturer must take the necessary precautions in the design so that the information is consistent with the characteristics of the batteries (HCF, 2008b).

5.1.1.3 Transmission Error Model

A probability of failure for transmission of a packet was added to the simulator, following the study carried out by Bildea et al. (2013). This model was experimentally obtained using the same radio transceiver family of the CC2500. In this study, experiments were conducted to relate the RSL between to nodes with the probability of communication failures.

5.1.2 Application layer

The application layer of the devices was also adapted so that some information and characteristics of the nodes could be obtained by the NM. Some commands were also not properly implemented in version 9.4.1 of the simulator Zand et al. (2014b). These commands are listed below.

- Command 777 (Read Device Capabilities) - implemented;
- Command 778 (Read Battery Life): - implemented;
- Command 779 (Report Device Health): - adapted to the standard;
- Command 780 (Report Neighbor Health List): adapted to the standard;
- Command 787 (Report Neighbor Signal Levels): - adapted to the standard;
- Command 788 (Alarm Path Down): - implemented.
- Command 799 (Request Service): - adapted to the standard.

Using the Alarm Path Down command, nodes can report to the NM any broken links with neighbors. A broken link with a neighbor is detected when a node does not receive a Keep-Alive message during a specified time interval defined in the standard (HCF, 2008b).

The nodes were configured to start sending the process data immediately after receiving the response to the Request Service command when the NM confirms the allocation of resources for sending process data. To evaluate PDR in the simulations, a sequence number was used in the data field of the process data packets sent from nodes to the gateway, so the gateway can identify missing packets. A data packet is considered missed when it is not received in the proper sequence by the gateway.

5.2 PERFORMANCE EVALUATION

The performance evaluation has the following objectives:

- To compare the performance of QLRR approaches with state-of-the-art routing algorithms, using the defined metrics;
- To evaluate how the Q-Learning parameters influence the performance of QLRR;
- To evaluate the performance using simulation parameters similar to those used in the state-of-the-art works and, at the same time, representing process monitoring applications of IWSN;

- To use statistical analysis to verify if the performance enhancements are significant.

The metrics considered in the performance evaluation are ANL, ENL, PDR, and PRN of the uplink graph. These metrics were already defined in Section 4.1.1.

During the thesis development, several repetitions of the simulations were conducted for a given set of topologies and routing algorithms. For each topology tested, samples of the metrics were collected during the last simulation hour. These samples were collected in the last hour to ensure that the measurements were stable, with no perturbations and changes in the routes caused by the learning agents. The analysis of these samples, for a given topology, indicated that they follow a normal distribution, while the variance presented similar values for all the routing algorithms. The average values and variances were different between each topology tested (KÜNZEL et al., 2018).

Analysis of Variance (ANOVA) is one of the statistical tools available for the analysis of data prone to errors (variations in the measurements). The basic assumptions of ANOVA are that errors are random variables that follow a normal distribution and are independent and that the variance is considered constant or similar for all levels of controllable factors (MONTGOMERY, 2006).

ANOVA defines that the outputs of an experiment (response variables, or in this case, the performance metrics) depend on a set of controllable factors (input variables) that can be applied at different levels (values). Each combination of levels for the controllable factors is known as a treatment. By collecting samples of the response variables for each treatment, it is then possible to verify the hypothesis that the controllable factors affect the response variables in a statistically-significant way (MONTGOMERY, 2006). The controllable factor is the routing algorithm used by the NM.

The main hypothesis verified was if the QLRR algorithms presented a statistically significant difference for ANL, ENL, PDR, or PRN, when compared to the state-of-the-art routing algorithms for a given topology. If a significant difference is verified between two treatments for a given response variable, the average value of this response variable is used to indicate the reduction or increase of the performance metric.

Once the hypotheses are tested for each topology, it can be calculated a ratio related to how many topologies (of a given set of topologies) QLRR enhanced the response variables, when compared side-by-side with each state-of-the-art routing algorithm. ANL is enhanced when it has a reduction, ENL, PDR, and PRN are enhanced when they have an increase in the average value.

With the information of the ratio (percentage) of topologies where QLRR presented significant enhancements, it is also possible to calculate an average percentage of reduction or increase of the performance metrics, allowing an estimation of the enhancements that a given routing algorithm gives when compared to another.

It is also verified if the QLRR algorithms present enhancements when the network has a different density of nodes (the number of nodes per area unit). In denser networks, more possibilities of connections will be available in the topology, thus providing more routing options for the nodes. Two different numbers of nodes per area are used in the simulations.

Another hypothesis verified is if the α and ε parameters of the Q-Learning algorithm significantly influenced the performance metrics on each algorithm. Sets were defined for α and ε. It was then identified the sets that presented significant enhancements for each topology.

The sample collection of the response variables was performed during a simulation time where the agents were not exploring and the network topology was not changing, and thus no reconfiguration was occurring. The last hour of the simulation was chosen for these measurements because the values of the response variables do not suffer significant variations since it is defined that the exploration phase will not happen during the last hours of the simulation.

As mentioned by Han et al. (2011), in a few cases, some topologies may have schedulability problems caused by the topology characteristics and because the communication resources (timeslots) may become unavailable during the scheduling process. In this case, the simulations finish before expected. If this situation occurs on a given test, the samples of this repetition are discarded from the evaluation.

5.3 SIMULATION PARAMETERS

The parameters used in the simulations represent typical WH industrial monitoring applications, where nodes are scattered over an industrial plant. The simulation parameters and the number of nodes used are also related to works described in Section 3.1. Table 12 presents the parameters used in the simulation environment, according to the layers of the WH protocol. Subsection 5.3.1 describes the parameters related to QLRR algorithms.

The results are analyzed from a set of 30 topologies, which allows generalizing the performance of the algorithms. The number of tests for each topology was defined as 15, based on the sample size formula considering $z = 1.96$, $s = 0.2$ s, $e = 0.1$ s for the ANL measurement.

The gateway is positioned in the center of the area along with the APs, and the connection between the gateway and the APs is considered to be wired and reliable (HAN et al., 2011). This positioning favors the formation of star topologies. However, as one of the objectives of the experiments is to allow the exploration of routes by the agents, the formation of more complex topologies must occur, where some nodes will act as routers. Wireless nodes are positioned randomly within the area.

Simulations were conducted with 20 and 40 nodes to verify the performance when IWSN have a different density of nodes. Nodes were numbered according to the distance from the gateway. The number of nodes used in the scenarios is defined based on the argument that

Table 12 – Parameters of the simulation

Statistical parameters	
Number of random topologies for each node density	30
Number of tests (simulation repetitions) for each topology	15
Number of ANL, ENL, PDR and PRN samples collected over the last hour of simulation	6 samples
Node and area parameters	
Area	100 x 100 m
Number of nodes (node density)	20, 40 nodes
Number of APs	2
Gateway position	Center
APs positions	5 m left and 5 m right from gateway
Nodes positions	Random
Nodes enumeration and startup sequence	According to gateway distance
Battery-powered nodes	50 %
Battery capacity	3.6 V 17 Ah
Physical layer parameters	
Channel number	16
Propagation model	Two-Ray Ground Model
Transmission power	0 dBm
Communication range	40 m
Signal level sensibility threshold	-85 dBm
Transmission error model	Bildea et al. (2013)
Time parameters	
Gateway startup time	0 min
APs startup time	$AP_1 = 2$ min, $AP_2 = 4$ min
First node startup	5 min
Nodes startup interval	1 min
Process data publish interval	32 s
Total simulation time	12 h
Health report interval (787, 779, 780)	15 min
Battery lifetime reading interval (778)	1 min
NM's periodic tasks interval	10 min

Source – The author.

IWSN applications generally have a maximum number of 50 nodes since latency is the main parameter in these applications, not the scalability (ÅKERBERG; GIDLUND; BJöRKMAN, 2011). The simulator used has limitations of scalability due to the availability of timeslots for the communication demands of the nodes and the scheduling algorithm used (ZAND et al., 2014b).

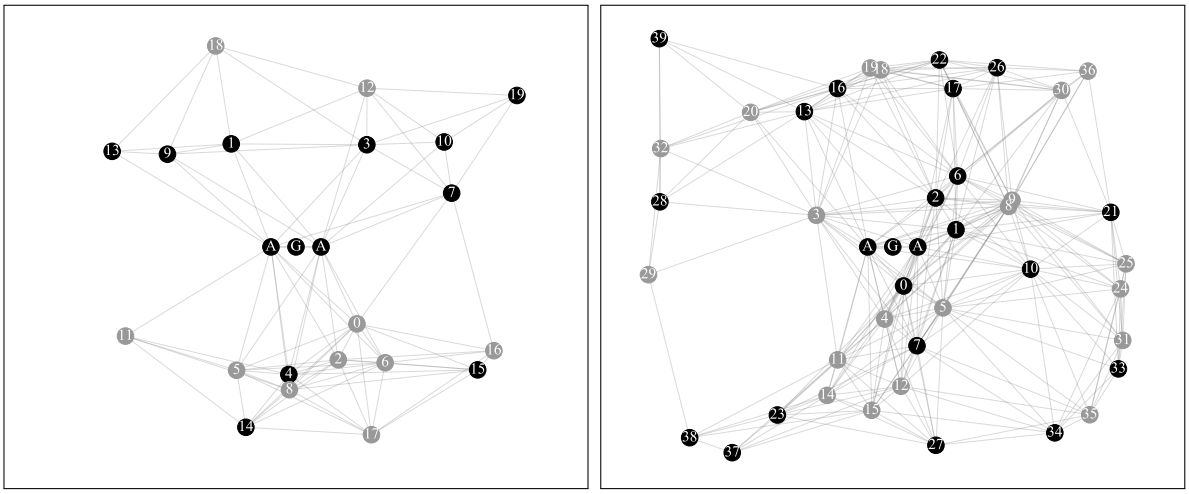
A proportion of 50 % of the nodes was powered with industrial-standard batteries (3.6 V, 17 Ah) with an expected lifetime of 10 years (CHEN; NIXON; MOK, 2010). The other nodes

were considered to be line powered.

Fig. 22 depicts the 20 and 40-node topologies used as examples for the performance evaluation, where black nodes are line-powered nodes, gray nodes are battery-powered nodes and lines represent that nodes are within the communication range of each other. The selected topologies presented the best results for the QLRR algorithms.

Figure 22 – 20 and 40-node topologies used for performance evaluation

(a) 20 nodes (b) 40 nodes



Source – The author.

The Two-Ray Ground-Reflection model with power transmission of 0 dBm was used for RSL estimation (ZAND et al., 2014b). The packet transmission error model used in the physical layer follows the analytic model for indoor environments by Bildea et al. (2013). The maximum communication range is a parameter required by the simulator and represents a value close to the RSL sensitivity threshold defined in the WH standard (-85 dBm) and also the point where the probability of failures is maximum in the fault model used. It was considered that two devices that are within the communication range of each other are able to send and receive messages to each other, following the same criteria of (HAN et al., 2011).

Each simulation starts with the startup of the NM/gateway and APs. The first node is turned on after 5 minutes, and then each node is turned on in one-minute intervals, according to the sequence number given during the topology construction. A new node listens the channels looking for an advertisement packet from its neighbors, and then starts the join process. When its join process is over, the node requests bandwidth to NM for sending process data. After receiving configurations from the NM (graphs, routes, superframes, links), it starts sending sensor readings towards the gateway. The period for sending process data was set to 32 seconds, following the work of (ZAND et al., 2014a). The health reports (787, 779, 780) are sent every 15 minutes and command 778 is polled by NM in one-minute intervals. The NM runs the management routines when a new device joins the network or in 10-minute intervals. The simulations run for 12 hours for each topology.

The QLRR-WA agent and the QLRR-MA agents will have the opportunity to perform an action in the environment only when the management routines are performed by period. This prevents reconfiguration from occurring in the graphs during the join process of the nodes, which may cause routes to be unavailable for sending configuration to the new nodes.

5.3.1 QLRR parameters

Table 13 presents the QLRR parameters. The ε - greedy policy is used because it is a commonly used strategy in the related works in Section 3.2. The exploration phase starts at the beginning of the simulation, where ε assumes the value of the set of learning parameters being used. The exploration ends when the simulation reaches 8 hours, when $\varepsilon = 0$. After the exploration phase, the agents go to the next state related to the state-action pair with greater value in $Q(s, a)$. For QLRR-WA, this means that the agent exploits the best set of weights found for Equation 4.1. For QLRR-MA, this means that an agent will connect to the pair of successors that provided the lowest average latency.

This exploration time was chosen for two reasons: it represents a feasible exploration time for practical applications, considering protocol characteristics, network setup and maintenance; and it represents the average number of iterations needed considering the complexity of RL problems for goal-oriented domains (KOENIG; SIMMONS, 1992). Considering 6 iterations per hour (due to the NM periodic tasks each 10 min), and the exploration time of 8 hours, it allows the agents to iterate approximately 48 times per simulation.

5.3.2 QLRR-WA parameters

It was considered $M = 7$ and all states that have weights with value 0 were removed in order to avoid abrupt changes in G_U that may occur by changes in the cost value. The initial state was set to s_7. States and actions used are depicted in Fig. 23. The value of s_d was adjusted so that neighbors with very low signal levels presented a higher cost. Equation 4.2 had the weights w_n and w_p set to 0.5, following the previous work of Künzel, Cainelli and Pereira (2017).

5.3.3 QLRR-MA parameters

The number of pairs of successors created from U_v was limited to a maximum of 15, to avoid a large number of available actions that would require an agent to explore for a long time. The successors in U_v are sorted according to the type of power supply, giving preference for line-powered successors. Nodes with no energy restrictions are preferably used and the battery-powered nodes are included in U_v only if the number of pairs is lower than 15.

Table 13 – QLRR parameters

QLRR common parameters	
Policy	ε-greedy
Exploration period	0 to 8 h
Exploitation period	8 h to 12 h
Q-table initial values	$Q(s,a) = 0, \forall a \in A, s \in S$
Reward value, R	1
t_s	5 min
k	2
Power energy cost p	1 (battery-powered), 0 (line-powered)
γ	0.80
QLRR-WA	
M	7
s_d	-45 dBm
N_w	3 (w_h, w_p, w_s)
Initial state	s_7 $(w_h = 0.28, w_p = 0.42, w_s = 0.28)$
Equation 4.2	$w_n = 0.5, w_p = 0.5$
Set of α and ε parameters	
Set A	$\alpha = 0.10$, $\varepsilon = 0.30$
Set B	$\alpha = 0.20$, $\varepsilon = 0.10$
Set C	$\alpha = 0.20$, $\varepsilon = 0.20$
Set D	$\alpha = 0.20$, $\varepsilon = 0.30$
Set E	$\alpha = 0.20$, $\varepsilon = 0.05$
Set F	$\alpha = 0.30$, $\varepsilon = 0.10$
Set G	$\alpha = 0.50$, $\varepsilon = 0.10$
Set H	$\alpha = 0.50$, $\varepsilon = 0.20$

Source – The author.

5.3.4 Q-Learning Parameter Sets

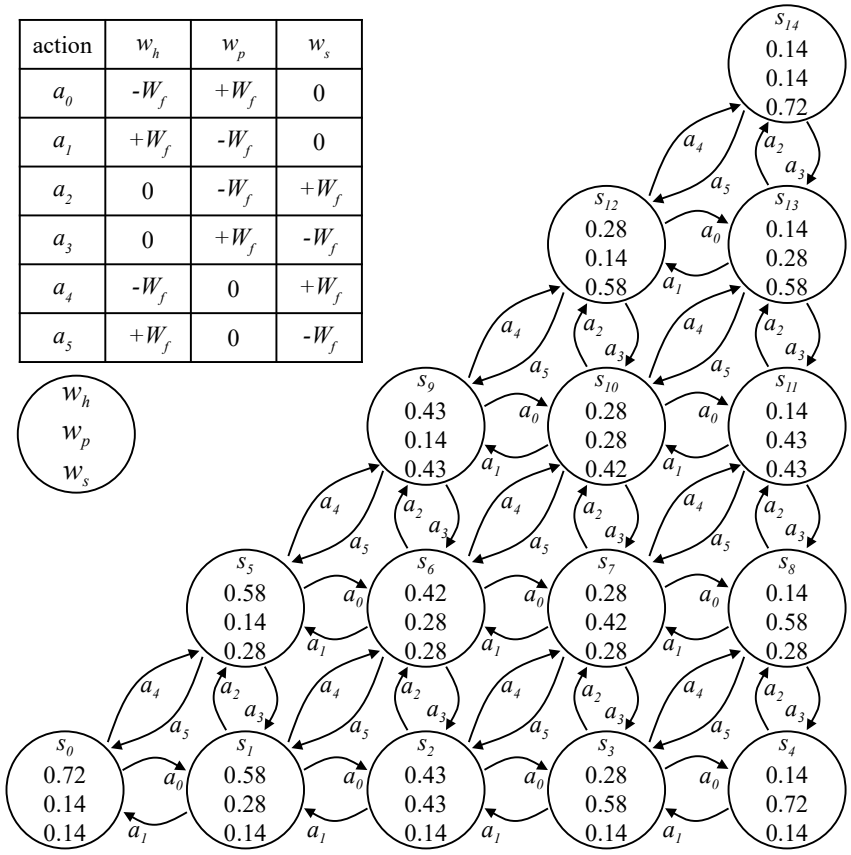
Table 13 also shows the sets A-H of parameters used in the simulations to evaluate how α and ε influence the results. The selected values are based on the related works.

5.3.5 State-of-the-art uplink graphs compared

The performance of QLRR was compared with the following baseline uplink algorithms:

- Han (HAN et al., 2011), which builds graphs trying to reduce the number of hops from the gateway;
- ELHFR (JINDONG; ZHENJUN; YAOPEI, 2009), which builds graphs based on the RSL of neighbors;

Figure 23 – States and actions used for QLRR-WA (rounded values)



Source – The author.

- Künzel (KÜNZEL; CAINELLI; PEREIRA, 2017), where weights were set to $w_h = 0, w_p = 1, w_s = 0$ on all related equations trying to build graphs that avoid battery-powered nodes as routers;
- The previous work in (KÜNZEL et al., 2018) was also used for comparison.

5.3.6 Downlink graphs used in the experiments

Downlink graphs were created using a BFS tree, where edges are chosen based on the RSL of neighbors. Downlink graphs without path redundancy were used in the experiments to reduce the number of links required during the scheduling process to ensure scalability, as scalability is a known limitation of the simulator (NOBRE; SILVA; GUEDES, 2015b).

5.3.7 Scheduling algorithm used in the experiments

The Han's scheduling algorithm was used for the allocation of links (HAN et al., 2011). It allocates each link on the paths from the source to the destination in a depth-first manner and splits the traffic from a node among all its successors by reducing the bandwidth requirement on each successor. The communication schedule of the successors is designed so that their

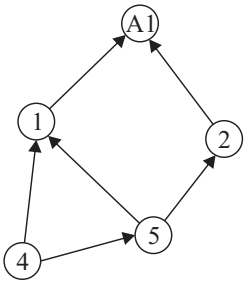
combination has the same pattern as the original node (NOBRE; SILVA; GUEDES, 2015b; HAN et al., 2011).

Links are allocated in the path between a source node u and a destination v through a depth-search algorithm on a constructed graph. For each link of u, the first timeslot t_i available in the superframe F in the position i is allocated, where the superframe has size l_F, equal to the data transmission period (DICKOW, 2014).

Considering that a WH network is typically multihop, the bandwidth required to perform the scheduling of all links from source to a recipient is large. This dramatically reduces scheduler performance and resource availability across the network. To solve this problem, the Han algorithm proposes a particular strategy when a device has more than one neighbor to route its data. Such strategy consists of creating a parallel F' superframe, with size $l'_F = 2l_F$, and allocating links in this superframe, reducing the publication period of the data. Therefore, only two paths are created for the neighbors of u and not for all possible paths from the node to the recipient (DICKOW, 2014).

Figure 24 represents an allocation of the Han scheduling algorithm to an example graph. The superframe F repeats twice while the superframe F' repeats only once in the same time period. Both are active simultaneously, and it turns out that there are no timeslot conflicts. The superframe F' is only used when the device to be scheduled has two successors. In the example above, only devices 4 and 5 have two successors, and therefore only such communications appear in the superframe F. The F superframe is used when the device has only one successor, or for routing communications using DFS.

Figure 24 – Han scheduling for an uplink graph



Superframe F — Superframe F+1

Timeslot	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Ch. Offset 0	1>A1	2>A1		1>A1	2>A1		1>A1	5>2	2>A1		1>A1	2>A1		1>A1	2>A1		1>A1	5>2	2>A1	
Ch. Offset 1																				
Ch. Offset 2																				

Superframe F'

Timeslot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Ch. Offset 0			5>1			4>1							5>2			4>5				
Ch. Offset 1																				
Ch. Offset 2																				

Source – Adapted from Dickow (2014)

It is worth noting that when the downlink and uplink graphs change, graphs and links must be reconfigured in the WirelessHART network, causing a significant communication overhead. The implementation of Han's scheduling algorithm by (ZAND et al., 2014b) tries to reduce the overhead by keeping in the schedule the links already allocated for a path that did not change, releasing links that are no longer necessary, and writing the links of the new paths. The NM allocates links in the schedule in the following sequence:

- Advertisement links are allocated for all devices currently in the network with a period t_{ADV}. They are used for neighbor discovery and for new nodes to join the network. The total number of advertisement links for each node is n_{ADV};
- Permanent links are allocated in the downlink graph with a period t_{DP}. These links are used for management communication and to keep a minimum connection between the gateway and a node. They are constrained to n_{DP} links between two nodes in the downlink graphs;
- Normal links are allocated in the downlink graph with a period t_{DN}. These links are used for management communication. They are limited to n_{DN} links between two nodes in the downlink graphs;
- Permanent links are allocated in the uplink graph with a period t_{UP}. These links are also used for management communication and to keep a minimum connection with the gateway. They are limited to n_{UP} links between two nodes in the uplink graphs;
- Normal links are allocated in the uplink graph with a period t_{UN}. These links are used for management communication. They are limited to n_{UN} links between two nodes in the uplink graph;
- Normal links are allocated for the requested services (send process data) according to the service's data transmission period.

The scheduling periods of these links are presented in Table 14.

Table 14 – Scheduler parameters

$t_{ADV} = 8$ s	$t_{DP} = 4$ s	$t_{DN} = 2$ s	$t_{UP} = 4$ s	$t_{UN} = 2$ s
$n_{ADV} = 3$	$n_{DP} = 1$	$n_{DN} = 6$	$n_{UP} = 1$	$n_{UN} = 6$

Source – The author.

5.4 RESULTS FOR THE 20 AND 40-NODE EXAMPLE TOPOLOGIES

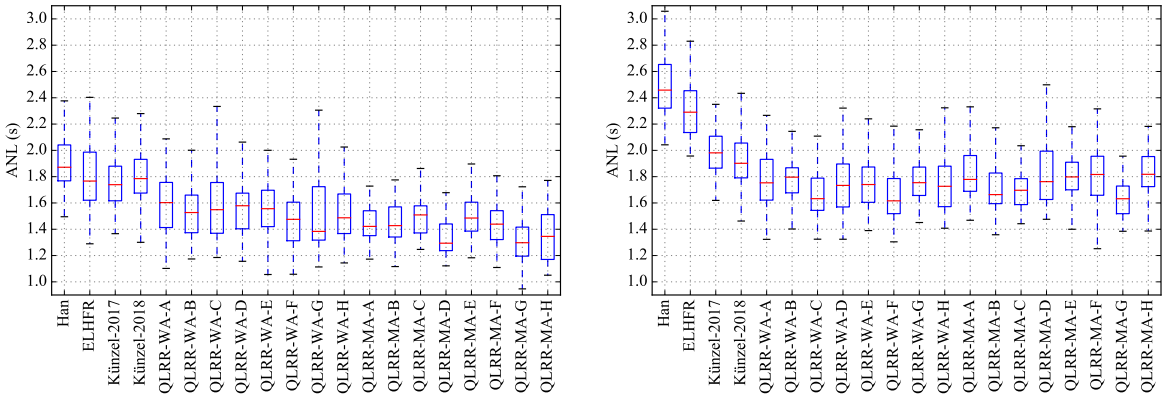
The results presented in this section are related to two topology examples. Samples of ANL, ENL, PRN, and PDR were collected over the last simulation hour (1 sample every 10 minutes) for each topology and algorithm tested. The values presented here are the average value of the samples collected for all the simulation repetitions for each topology.

5.4.1 Average Network Latency

Figure 25 presents the ANL boxplots for the example topologies.

Figure 25 – Average Network Latency boxplots for the example topologies

(a) 20 nodes (b) 40 nodes



Source – The author.

5.4.1.1 20-node example topology

For the 20-node topology, QLRR-WA and QLRR-MA significantly reduced ANL when compared to the state-of-the-art algorithms. It was observed that the parameter sets presented a few differences in the final results. Sets F, G, and H presented the lowest ANL values for QLRR-WA, although the variances in set G presented higher value. QLRR-MA showed a greater reduction in latency compared to QLRR-WA. Sets D, G, and H presented relevant results for the ANL in QLRR-MA.

5.4.1.2 40-node example topology

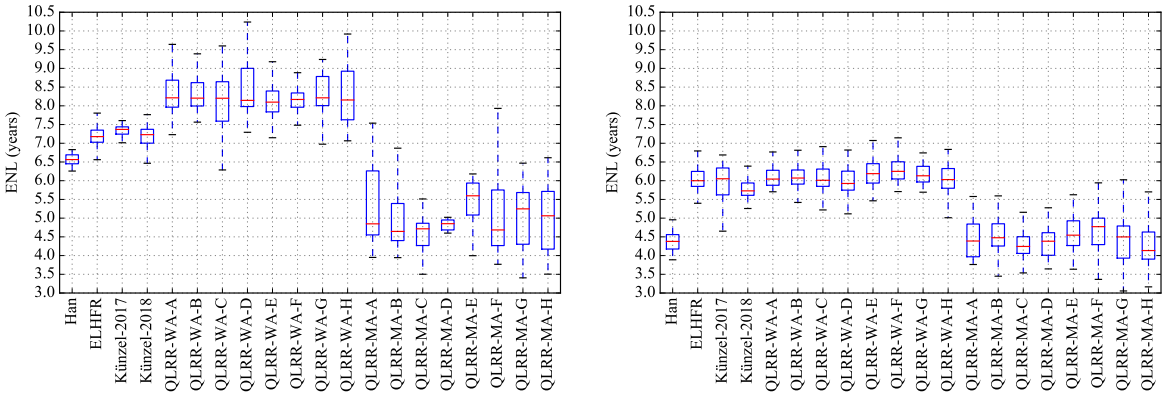
For the 40-node topology, QLRR-WA and QLRR-MA also showed significant reductions in the ANL when compared to the state-of-the-art algorithms. Considering the ANL, the most relevant sets for QLRR-WA were C and F, and sets B and G for QLRR-MA.

5.4.2 Expected Network Lifetime

Figure 26 presents the ENL boxplots for the example topologies. For the 20-node topology, QLRR-WA showed a significant improvement over the state-of-the-art algorithms. For the 40-node topology, QLRR-WA presented an ENL higher or equal than the state-of-the-art algorithms.

Figure 26 – Expected Network Lifetime boxplots for topology examples

(a) 20 nodes (b) 40 nodes



Source – The author.

In general, QLRR-MA presented the lowest ENL. This is because the reward given to the agents prioritizes the latency reduction only, and nodes will eventually connect to battery-powered devices to maximize the rewards. The ε and the use of one agent per node may frequently change the uplink graph, causing greater reconfiguration and energy consumption.

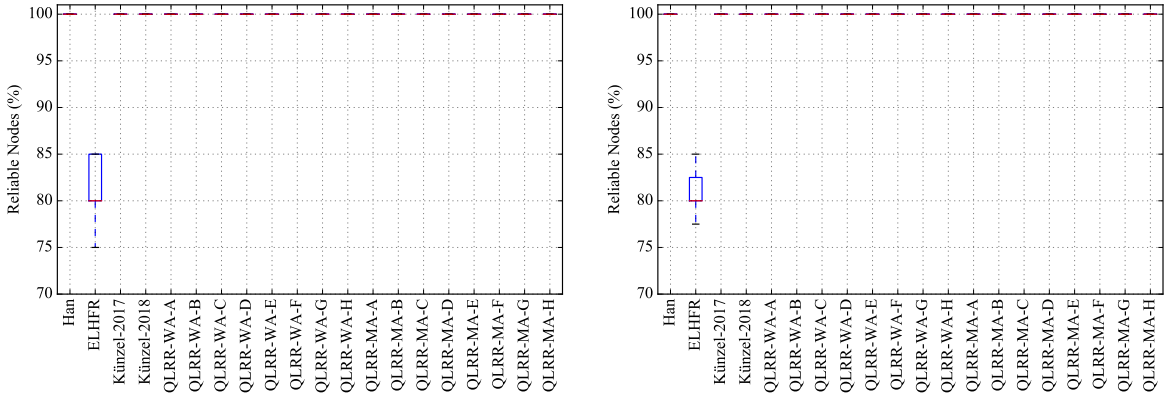
It was observed that the parameter sets used in QLRR did not cause significant differences in the ENL in both topologies. but variances presented different values in the 20-node topology.

5.4.3 Percentage of Reliable Nodes

Figure 27 presents the PRN boxplots for the example topologies. It can be seen that, except for ELHFR, all algorithms were able to provide path redundancy in the uplink graph, ensuring that each node had two successors for sending messages. All algorithms kept 100 % of reliable nodes in the example topologies. ELHFR presented 80 % of reliable nodes in the 20 and 40-node topology. ELHFR presented the lowest PRN because it allows nodes to connect only to neighbors from lower levels in the BFS tree, and usually a few neighbors are available on these levels. No variations were identified in the PRN value in those experiments except for ELHFR.

Figure 27 – Percentage of Reliable Nodes for topology examples

(a) 20 nodes (b) 40 nodes



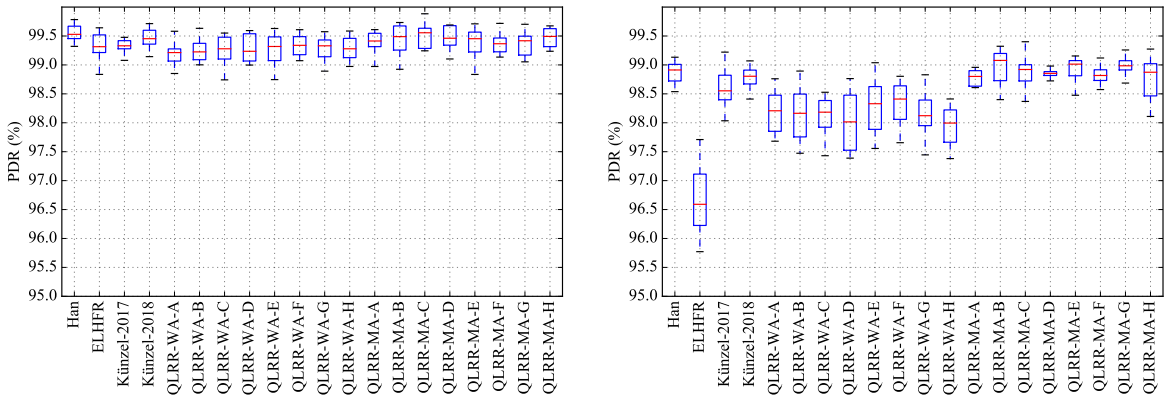
Source – The author.

5.4.4 Packet Delivery Ratio

Figure 28 presents the PDR boxplots for the example topologies. For the 20-node topology, all algorithms had similar values for the packet delivery rate. In the 40-node topology, QLRR-MA presented the highest PDR. The ELHFR algorithm presented the lower results, mostly because it does not provide path redundancy for all nodes.

Figure 28 – Packet Delivery Ratio boxplots for topology examples

(a) PDR for 20 nodes (b) PDR for 40 nodes



Source – The author.

5.4.5 Average Network Latency over time

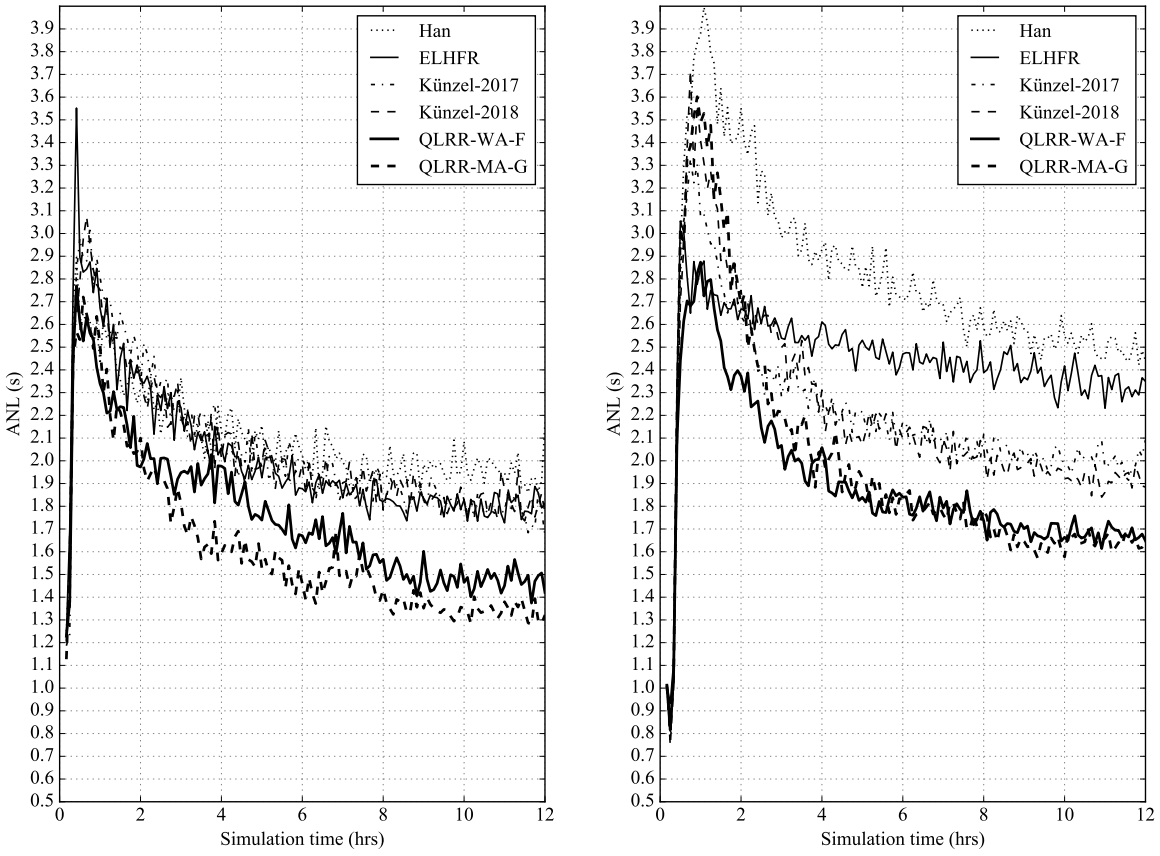
Fig. 29 depicts the ANL of the 20 and 40-node topologies over the simulation time. During the startup of the network (from 0 to 4 hours), the ANL increased because nodes were

joining the network and exchanging commands with the Network Manager. After 6 hours, all algorithms started to stabilize the ANL. Slight variations are presented for all algorithms, mostly caused by packet retransmission, paths taken during packet propagation, or when a Path Down Alarm is received by the NM, causing reconfiguration of the uplink graph.

The ANL stabilized in the Han, ELHFR, and Künzel (KÜNZEL; CAINELLI; PEREIRA, 2017) after the join process because the topology stopped changing. In QLRR, the ANL presented slight variations from 0 to 8 hours because of the exploration phase. After the exploration, QLRR algorithms proceeded to the best state, stabilizing the ANL in a reduced value when compared to the other algorithms.

Figure 29 – Average Network Latency over simulation time

(a) ANL for 20 nodes (b) ANL for 40 nodes



Source – The author.

5.5 GENERAL RESULTS OVER SEVERAL TOPOLOGIES

The same simulations were conducted to verify if QLRR presented a similar performance over random topologies for each node density level (30 random topologies for the 20-node level and 30 for the 40-node level). Each topology has unique characteristics for the performance

metrics because of the spatial distribution and the power source of the nodes. The ANL, ENL, PDR, and PRN samples of the last simulation hour were collected for several repetitions of the simulations for each topology and then used One-way Analysis of Variance with a 95 % significance to verify in how many topologies QLRR enhanced ANL, ENL and PDR on each topology when compared side-by-side to the other routing algorithms. For this analysis, the set F ($\alpha = 0.3, \varepsilon = 0.05$) was considered for QLRR-WA, and set G ($\alpha = 0.5, \varepsilon = 0.2$) for QLRR-MA.

5.5.1 Results for QLRR-WA

Table 15 shows the percentage of topologies where QLRR-WA significantly reduced ANL or increased ENL and PDR. We omitted the PRN because it was always over 85 % for Han, Künzel and QLRR-WA and over 75 % for ELHFR in all topologies, not presenting significant differences.

To clarify the information presented in Table 15, we highlighted the values of the comparison of QLRR-WA-F with the Han algorithm, and explain below how to interpret the highlighted information:

- For the 20-node topologies, QLRR-WA-F improved the ANL in 86 % of the topologies, when compared to Han;
- For the 20-node topologies, QLRR-WA-F improved the ENL in 53 % of the topologies, when compared to Han;
- For the 20-node topologies, QLRR-WA-F improved the PDR in 36 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-WA-F improved the ANL in 100 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-WA-F improved the ENL in 90 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-WA-F improved the PDR in 46 % of the topologies, when compared to Han;

The results also indicate that QLRR-WA presented a better performance in denser networks, mostly because nodes have more neighbors to provide routes.

Table 16 shows the average percentage of reduction of the ANL value and the average percentage of increase of the ENL value of the QLRR-WA algorithm for the topologies where QLRR-WA presented significant improvements when compared to the state-of-the-art algorithms. The PDR improvements were less than 1 %. To clarify the information presented in Table 16, we

Table 15 – Percentage of topologies where QLRR-WA-F improved ANL, ENL, PDR

Parameters		Routing algorithm compared with QLRR-WA-F				
Nodes	Metric	Han	ELHFR	Künzel-2017	Künzel-2018	QLRR-MA-G
20	ANL	**86** %	83 %	70 %	80 %	6 %
	ENL	**53** %	23 %	33 %	43 %	80 %
	PDR	**36** %	66 %	43 %	36 %	20 %
40	ANL	**100** %	93 %	96 %	96 %	20 %
	ENL	**90** %	86 %	16 %	63 %	90 %
	PDR	**46** %	93 %	73 %	63 %	40 %

Source – The author

highlighted the comparison of QLRR-WA-F with the Han algorithm, and explain below how to interpret the highlighted information:

- For the 20-node topologies where QLRR-WA-F reduced the ANL when compared to Han, the average ANL reduction was of 15 %;
- For the 20-node topologies where QLRR-WA-F increased the ENL when compared to Han, the average ENL increase was of 29 %;
- For the 20-node topologies where QLRR-WA-F increased the PDR when compared to Han, the average PDR increase was of 0.3 %;
- For the 40-node topologies where QLRR-WA-F reduced the ANL when compared to Han, the average ANL reduction was of 28 %;
- For the 40-node topologies where QLRR-WA-F increased the ENL when compared to Han, the average ENL increase was of 79 %;
- For the 40-node topologies where QLRR-WA-F increased the PDR when compared to Han, the average PDR increase was of 0.2 %;

Table 16 – Reduction of ANL, increase of ENL and PDR with QLRR-WA-F

Parameters		Routing algorithm compared with QLRR-WA-F				
Nodes	Metric	Han	ELHFR	Künzel-2017	Künzel-2018	QLRR-MA-G
20	ANL	**-15** %	-19 %	-12 %	-10 %	-7 %
	ENL	**29** %	29 %	9 %	9 %	34
	PDR	**0.3** %	0.8 %	0.2 %	0.3 %	0.2
40	ANL	**-28** %	-18 %	-19 %	-16 %	-11 %
	ENL	**79** %	64 %	10 %	10 %	52 %
	PDR	**0.2** %	1 %	0.3 %	0.2 %	0.2 %

Source – The author.

5.5.2 Results for QLRR-MA

Table 17 shows the percentage of topologies where QLRR-MA significantly reduced ANL or increased ENL and PDR. In most cases, QLRR-MA improves ANL and PDR. The results also indicated that QLRR-MA presented a better performance in denser networks. In all topologies, PRN was over 85 % for Han, Künzel and QLRR-WA and over 75 % for ELHFR and did not present significant differences between the algorithms.

- For the 20-node topologies, QLRR-MA-G improved the ANL in 76 % of the topologies, when compared to Han;
- For the 20-node topologies, QLRR-MA-G improved the ENL in 13 % of the topologies, when compared to Han;
- For the 20-node topologies, QLRR-MA-G improved the PDR in 70 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-MA-G improved the ANL in 96 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-MA-G improved the ENL in 63 % of the topologies, when compared to Han;
- For the 40-node topologies, QLRR-MA-G improved the PDR in 50 % of the topologies, when compared to Han;

Table 17 – Percentage of topologies where QLRR-MA-G improved ANL, ENL, PDR

Parameters		Routing algorithm compared with QLRR-MA-G				
Nodes	Metric	Han	ELHFR	Künzel-2017	Künzel-2018	QLRR-WA-F
20	ANL	**76** %	80 %	70 %	66 %	53 %
	ENL	**13** %	3 %	3 %	0 %	0 %
	PDR	**70** %	73 %	43 %	43 %	26 %
40	ANL	**96** %	93 %	96 %	93 %	26 %
	ENL	**63** %	46 %	3 %	3 %	3 %
	PDR	**50** %	93 %	86 %	73 %	46 %

Source – The author.

Table 18 shows the percentage of reduction of the ANL for the topologies where QLRR-MA presented significant improvements when compared to the state-of-the-art algorithms. The table also shows the average reduction of the ENL for the QLRR-MA algorithm. The PDR improvements were less than 1 %.

- For the 20-node topologies where QLRR-MA-G reduced the ANL when compared to Han, the average ANL reduction was of 22 %;

- For the 20-node topologies where QLRR-MA-G reduced the ENL when compared to Han, the average ENL reduction was of 19 %;
- For the 20-node topologies where QLRR-MA-G increased the PDR when compared to Han, the average PDR increase was of 0.3 %;
- For the 40-node topologies where QLRR-MA-G reduced the ANL when compared to Han, the average ANL reduction was of 29 %;
- For the 40-node topologies where QLRR-MA-G reduced the ENL when compared to Han, the average ENL reduction was of 15 %;
- For the 40-node topologies where QLRR-MA-G increased the PDR when compared to Han, the average PDR increase was of 0.2 %;

Table 18 – Reduction of ANL and ENL, increase of PDR with QLRR-MA-G

Parameters		Routing algorithm compared with QLRR-MA-G				
Nodes	Metric	Han	ELHFR	Künzel-2017	Künzel-2018	QLRR-WA-F
20	ANL	**-22** %	-22 %	-17 %	-18 %	-10 %
	ENL	**-19** %	-31 %	-22 %	-22 %	-23 %
	PDR	**0.3** %	0.8 %	0.3 %	0.3 %	0.3 %
40	ANL	**-29** %	-19 %	-19 %	-18 %	-7 %
	ENL	**-15** %	-16 %	-33 %	-28 %	-31 %
	PDR	**0.2** %	0.1 %	0.2 %	0.2 %	0.3 %

Source – The author.

6 CONCLUSIONS

In this thesis, a Reinforcement Learning model known as Q-Learning was used to build routes for IWSN with centralized management. Two approaches were presented, considering the requirements of IWSN protocols such as WirelessHART. The approaches were named QLRR-WA and QLRR-MA. Both approaches run in the Network Manager. The QLRR-WA approach uses a learning agent that adjusts the weights of a cost equation that defines how nodes will connect to neighbors in the uplink graph. In the QLRR-MA approach, each node has a learning agent that chooses which neighbors to connect to.

The rewards proposed have different objectives in each approach: In QLRR-WA, the reward aims to find a set of weights that reduces the Average Network Latency and increases the Expected Network Lifetime. In QLRR-MA, the reward aims to reduce the average latency for each node, consequently reducing the Average Network Latency.

One of the available simulation environments for WirelessHART was selected for the evaluation of the QLRR approaches. The environment was improved using state-of-the-art energy and transmission models and providing WirelessHART application layer commands that were not yet implemented in the simulator of Zand et al. (2014a).

A methodology was presented to evaluate the performance of the approaches. The methodology included scenarios, node configuration, timing parameters, routing, and scheduling algorithms and statistical analysis using ANOVA. The QLRR algorithms and some of the relevant state-of-the-art graph routing algorithms were implemented in the simulation environment to evaluate and compare their performance.

Considering the given scenarios and simulations performed, both QLRR approaches presented relevant results for most of the topologies tested. When compared to the state-of-the-art algorithms, QLRR-WA maintained a lower ANL while increased or kept the ENL similar to the other works, thus balancing these two characteristics. QLRR-MA was able to reduce the ANL to values lower than the other approaches, but with a reduction of the ENL. QLRR-MA slightly increased the Packet Delivery Rate in most cases, when compared to the other algorithms. Both QLRR approaches maintained high reliability in the uplink graphs, ensuring path redundancy.

The results indicate that RL and QLRR may be useful for future centralized-management approaches, IWSN applications, and also for other centralized protocols targeting IoT, IIoT, and Industry 4.0. But it is important to mention that the use of RL, Q-Learning, and QLRR for routing may require significant efforts in evaluation and simulation for each protocol and application to ensure feasibility.

Based on the discussion and the results of this thesis, Section 6.1 presents relevant

research possibilities and further discussion on IWSN protocols and the use of Artificial Intelligence (AI) on those protocols.

6.1 FUTURE WORKS

6.1.1 IWSN protocols

The performance of RL routing approaches will be influenced by the number of iterations spent in the exploration phase, the time interval between iterations of the learning agent, parameters, rewards, and protocol characteristics. The centralized management protocols or applications with complex topologies (with several hops from nodes to gateway) must consider the reconfiguration times involved as a limiting factor for the use of RL, as discussed in Section 4.4. The tests conducted with WirelessHART indicated that this protocol takes long periods to configure nodes during the joining process and to properly reconfigure after a change in routes, as verified by (ZAND et al., 2014b). The reduction of the reconfiguration times in IWSN centralized protocols is a relevant issue.

6.1.2 QLRR evaluation

The evaluation of QLRR can be conducted for other IWSN, IoT and IIoT protocols. Also, the evaluation can use other topologies, scenarios, and rewards. Experiments can also be performed in real-world applications.

6.1.3 New QLRR approaches

QLRR approaches can be further developed to cope with node mobility, transmission power adjustments, coexistence, and interference.

6.1.4 Scheduling algorithms

The scheduling algorithms influence QLRR's performance because of the strategy used for the allocation of links and because the actions that are taken in the environment change the current schedule. Thus, a proper combination of routing and scheduling algorithms is needed to enhance the performance, as indicated by (NOBRE; SILVA; GUEDES, 2015b). Future works can treat routing and scheduling as a single RL problem, where states and actions could represent conditions and changes in routes and in the allocation of timeslots.

6.1.5 Network Manager architectures

Network Manager architectures can be developed to provide an optimized configuration for a given operational network, using information about the current topology and network

conditions. These architectures could include an optimization/simulation module inside the NM. A cloud computing processing service could also be used to reduce NM's hardware complexity. The idea of using real data to feed a simulation is also suggested in (DEDE et al., 2018).

6.1.6 Development of routing algorithms using other Artificial Intelligence models

Other AI models such as Deep Reinforcement Learning, Neural Networks, and Deep Neural Networks could be used to build graphs based on data gathered and learned from the network.

6.1.7 Analysis of the solutions provided by RL and AI

The main goal of an RL agent is to increase its rewards, and it will explore and exploit the environment to reach this goal. Depending on the application, it can be difficult to predict how the actions of the agent will affect the environment to increase its rewards. Eventually, it may cause adverse situations that can compromise the functionality of a system.

An example of a situation that could happen in a centralized management IWSN would be a case where an RL agent chooses routes that could accidentally remove some nodes from the periphery of the network (by choosing poor links with neighbors in QLRR-WA, for example), causing that these nodes could not send data packets anymore. These lost data packets may not be considered during the ANL measurement. This may cause the agent to receive positive rewards since ANL was reduced (because the nodes in the periphery usually have increased latency in the system and are not able to send data packets anymore).

These situations must be evaluated previously and rules should be created to verify if the agent is respecting the application requirements. This brings importance to the development of proper simulation and experiment environments to test the RL approaches before using in real-world applications. At the same time, RL can be used to discover and predict these problems in the protocols and applications.

As mentioned by authors in (MAMMERI, 2019), it is difficult to guarantee or estimate the performance gain of using RL on each application, due to the conditions of the physical environment and the learning characteristics themselves. Again, proper simulation tools can be useful.

Given the AI characteristics, the requirements of the application, the complexity of the protocols and the relevant number of parameters associated with the network and devices, the development and use of simulation tools that reproduce in detail the network operating conditions is fundamental to evaluate new AI approaches in the future.

6.2 PUBLICATIONS

The first results and simulations with the weighted routing algorithm used as baseline for the QLRR-WA were presented in:

- CAINELLI, G. P. ; KÜNZEL, G. ; PEREIRA, C. E. Algoritmo de Broadcast com Pesos para Redes WirelessHART. In: **XIII Brazilian Symposium on Intelligent Automation (SBAI)**, 2017, Porto Alegre;
- KÜNZEL, G; CAINELLI, G. P. ; PEREIRA, C. E. A Weighted Broadcast Routing Algorithm for WirelessHART Networks. In: **VII Brazilian Symposium on Computing Systems Engineering (SBESC)**, 2017, Curitiba;
- KÜNZEL, G; CAINELLI, G. P. ; MULLER, I. ; PEREIRA, C. E. Simulação e Análise de Desempenho de um Algoritmo de Roteamento com Pesos para Redes Industriais sem Fio. In: **VIII Brazilian Symposium on Computing Systems Engineering (SBESC)**, 2018, Salvador.

Works related to industrial wireless communication protocols and concepts were presented in:

- VALADÃO, Y. N.; KÜNZEL, G.; MULLER, I.; PEREIRA, C. E. Industrial Wireless Automation: Overview and Evolution of WIA-PA. In: **III IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)**, 2018, Faro, Portugal;

  SILVA, M. J. ; LIMA, J. ; KÜNZEL, G. ; PEREIRA, C. E. ; FREITAS, E. P. A Study on Interference Between Industrial and Assistive Body Area Wireless Networks. In: **XXII Congresso Brasileiro de Automatica**, 2018, Joao Pessoa;

The initial results with the QLRR-WA approach were presented in:

- KÜNZEL, G; CAINELLI, G. P. ; MULLER, I. ; PEREIRA, C. E. Weight Adjustments in a Routing Algorithm for Wireless Sensor and Actuator Networks Using Q-Learning. In: **III IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT)**, 2018, Faro, Portugal.

The full QLRR-WA and QLRR-MA approaches, the performance evaluation and the comparison with state-of-the-art algorithms were presented in:

- KÜNZEL, G.; INDRUSIAK, L. S.; PEREIRA, C. E. Latency and lifetime enhancements in IWSN: a Q-learning approach for graph routing. **IEEE Transactions on Industrial Informatics**,v. 16, n. 8, p. 5617–5625, 2020;

- KÜNZEL, G.; CAINELLI, G.P,; MULLER, I.; INDRUSIAK, L. S.; PEREIRA, C. E. A Reliable and Low-Latency Graph-Routing Approach for IWSN using Q-Routing. In: **X Brazilian Symposium on Computing Systems Engineering (SBESC)**, 2020, Online Event.

6.3 SOURCE CODES

The source codes are available at: <http://bit.do/gustavo-kunzel>.

BIBLIOGRAPHY

ÅKERBERG, J.; GIDLUND, M.; BJöRKMAN, M. Future research challenges in wireless sensor and actuator networks targeting industrial automation. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 9., 2011, Caparica. **Proceedings [...]**. Caparica: IEEE, 2011. p. 410–415.

AKYILDIZ, I. F.; WANG, X.; WANG, W. Wireless mesh networks: A survey. **Computer Networks ISDN Systems**, Elsevier Science Publishers B. V., Amsterdam, Netherlands, v. 47, n. 4, p. 445–487, mar. 2005. ISSN 0169-7552.

AL-RAWI, H. A.; NG, M. A.; YAU, K.-L. A. Application of reinforcement learning to routing in distributed wireless networks: A review. **Artifficial Intelligence. Reviews**, Kluwer Academic Publishers, Norwell, MA, USA, v. 43, n. 3, p. 381–416, mar. 2015. ISSN 0269-2821.

BAYOU, L. et al. WirelessHART netsim: A WirelessHART SCADA-based wireless sensor networks simulator. In: WORKSHOP ON SECURITY OF INDUSTRIAL CONTROL SYSTEMS AND CYBER PHYSICAL SYSTEMS, 1., 2015, Vienna. **Proceedings [...]**. Vienna: Springer, 2015. p. 63–78.

BILDEA, A. et al. Link quality metrics in large scale indoor wireless sensor networks. In: IEEE ANNUAL INTERNATIONAL SYMPOSIUM ON PERSONAL, INDOOR, AND MOBILE RADIO COMMUNICATIONS, 24., 2013, Toronto. **Proceedings [...]**. London, UK: IEEE, 2013. p. 1888–1892.

CAINELLI, G. P.; KÜNZEL, G.; PEREIRA, C. E. Algoritmo de broadcast com pesos para redes WirelessHART. In: BRAZILIAN SYMPOSIUM ON INTELLIGENT AUTOMATION, 13., 2017, Porto Alegre. **Proceedings [...]**. Porto Alegre: IEEE, 2017. p. 95–102.

CHANG, X. Network simulations with OPNET. In: CONFERENCE ON WINTER SIMULATION: SIMULATION, 31., 1999, Aveiro. **Proceedings [...]**. New York: ACM, 1999. p. 307–314.

CHEN, D.; NIXON, M.; MOK, A. **WirelessHART: Real-Time Mesh Network for Industrial Automation**. 1st. ed. New York: Springer Publishing Company, Incorporated, 2010. ISBN 1441960465.

DEBOWSKI, B.; SPACHOS, P.; AREIBI, S. Q-Learning enhanced gradient based routing for balancing energy consumption in wsns. In: IEEE INTERNATIONAL WORKSHOP ON COMPUTER AIDED MODELLING AND DESIGN OF COMMUNICATION LINKS AND NETWORKS, 21., 2016, Toronto. **Proceedings [...]**. Toronto: IEEE, 2016. p. 18–23.

DEDE, J. et al. Simulating opportunistic networks: Survey and future directions. **IEEE Communications Surveys Tutorials**, v. 20, n. 2, p. 1547–1573, 2018.

DICKOW, V. H. **Avaliação de Algoritmos de Roteamento e Escalonamento de Mensagens para Redes WirelessHART**. 2014. 89 f. Dissertation (Master in Electrical Engineering) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

DOWLING, J. et al. Using feedback in collaborative reinforcement learning to adaptively optimize manet routing. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, v. 35, n. 3, p. 360–372, May 2005. ISSN 1083-4427.

DUNKELS, A.; GRONVALL, B.; VOIGT, T. Contiki-a lightweight and flexible operating system for tiny networked sensors. In: IEEE INTERNATIONAL CONFERENCE ON LOCAL COMPUTER NETWORKS, 29., 2004, Tampa. **Proceedings [...]**. Tampa: IEEE, 2004. p. 455–462.

FERRARI, P. et al. Improving simulation of wireless networked control systems based on WirelessHART. **Computer Standard Interfaces**, Elsevier Science Publishers B. V., Amsterdam, Netherlands, v. 35, n. 6, p. 605–615, nov. 2013. ISSN 0920-5489.

GAO, G.; ZHANG, H.; LI, L. An OPNET-based simulation approach for the deployment of WirelessHART. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY, 9., 2012, Chongqing. **Proceedings [...]**. Chongqing: IEEE, 2012. p. 2120–2124.

GAO, J. et al. Multi-agent Q-Learning aided backpressure routing algorithm for delay reduction. **arXiv preprint arXiv:1708.06926**, 2017.

GHAFFARI, A. Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms. **Wireless Networks**, v. 23, n. 3, p. 703–714, Apr 2017. ISSN 1572-8196.

GUO, W.; YAN, C.; LU, T. Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing. **International Journal of Distributed Sensor Networks**, v. 15, n. 2, 2019.

GVR. **Industrial Wireless Sensor Network Market Worth $8.67 Billion By 2025**. 2018. Available: <https://www.grandviewresearch.com/press-release/global-industrial-wireless-sensor-networks-iwsn-market>. Access: 27 Apr 2018.

HABIB, M. A.; ARAFAT, M. Y.; MOH, S. Routing protocols based on reinforcement learning for wireless sensor networks: A comparative study. **Journal of Advanced Research in Dynamical and Control Systems**, p. 427–435, jan 2019.

HAN, S. et al. Reliable and real-time communication in industrial wireless mesh networks. In: IEEE REAL-TIME AND EMBEDDED TECHNOLOGY AND APPLICATIONS SYMPOSIUM, 17., 2011, Chicago. **Proceedings [...]**. Chicago: IEEE, 2011. p. 3–12.

HAN, X.; MA, X.; CHEN, D. Energy-balancing routing algorithm for WirelessHART. In: IEEE INTERNATIONAL WORKSHOP ON FACTORY COMMUNICATION SYSTEMS, 15., 2019, Sundsvall. **Proceedings [...]**. Sundsvall: IEEE, 2019. p. 1–7.

HCF. **HCF_SPEC-065: 2.4 GHz DSSS O-QPSK Physical Layer Specification**. Austin, 2007. 20 p.

HCF. **HCF_SPEC-075: TDMA Data Link Layer Specification**. Austin, 2008. 76 p.

HCF. **HCF_SPEC-155: Wireless Command Specification**. Austin, 2008. 140 p.

HCF. **HCF_SPEC-290: WirelessHART Device Specification**. Austin, 2008. 159 p.

HCF. **HCF_SPEC-085: Network Management Specification**. Austin, 2009. 98 p.

HENRIKSSON, D.; CERVIN, A.; ÅRZÉN, K. erik. TrueTime: Real-time control system simulation with MATLAB/Simulink. In: NORDIC MATLAB CONFERENCE, 2003, Copenhagen. **Proceedings [...]**. Copenhagen: Fema, 2003.

HERRMANN, M. J.; MESSIER, G. G. Cross-layer lifetime optimization for practical industrial wireless networks: A petroleum refinery case study. **IEEE Transactions on Industrial Informatics**, v. 14, n. 8, p. 3559–3566, 2018.

HONG, S. H. et al. An energy-balancing graph-routing algorithm for WirelessHART networks. In: IEEE ASIA PACIFIC CONFERENCE ON WIRELESS AND MOBILE, 2015, Bandung. **Proceedings [...]**. Bandung: IEEE, 2015. p. 239–245.

IKRAM, W. et al. Adaptive multi-channel transmission power control for industrial wireless instrumentation. **IEEE Transactions on Industrial Informatics**, v. 10, n. 2, p. 978–990, May 2014. ISSN 1551-3203.

JIN, Z. et al. A Q-Learning-based delay-aware routing algorithm to extend the lifetime of underwater sensor networks. **Sensors**, v. 17, n. 7, 2017. ISSN 1424-8220.

JINDONG, Z.; ZHENJUN, L.; YAOPEI, Z. Elhfr: A graph routing in industrial wireless mesh network. In: INTERNATIONAL CONFERENCE ON INFORMATION AND AUTOMATION, 2009, Macau. **Proceedings [...]**. Macau: IEEE, 2009. p. 106–110.

KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. **Journal of Artificial Intelligence Research**, AI Access Foundation, USA, v. 4, n. 1, p. 237–285, maio 1996. ISSN 1076-9757.

KIANI, F. et al. Efficient intelligent energy routing protocol in wireless sensor networks. **International Journal of Distributed Sensor Networks**, v. 11, n. 3, 2015.

KOENIG, S.; SIMMONS, R. G. **Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains**. Carnegie Mellon University, School of Computer Science, Pittsburg, US, 1992.

KONOVALOV, I. **A Framework for WirelessHART Simulations**. 16. ed. Computer Systems Laboratory, Swedish Institute of Computer Science, Kista, Sweden, 2010. (SICS Technical Report, 2010:06).

KOSUNALP, S. et al. Use of Q-Learning approaches for practical medium access control in wireless sensor networks. **Engineering Applications of Artificial Intelligence**, v. 55, p. 146–154, 2016. ISSN 0952-1976.

KÜNZEL, G. **Ambiente de Avaliação de Estratégias de Roteamento em Redes WirelessHART**. 2012. 95 f. Dissertation (Master in Electrical Engineering) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

KÜNZEL, G. et al. Weight adjustments in a routing algorithm for wireless sensor and actuator networks using Q-Learning. In: IFAC CONFERENCE ON EMBEDDED SYSTEMS, COMPUTATIONAL INTELLIGENCE AND TELEMATICS IN CONTROL, 3., 2018, Faro. **Proceedings [...]**. Faro: IFAC-PapersOnLine, 2018. v. 51, n. 10, p. 58–63.

KÜNZEL, G.; CAINELLI, G. P.; PEREIRA, C. E. A weighted broadcast routing algorithm for WirelessHART networks. In: BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING, 7., 2017, Curitiba. **Proceedings [...]**. Curitiba: IEEE, 2017. p. 187–192.

LI, Y. et al. Adaptive optimization-based routing in wireless mesh networks. **Wireless Personal Communications**, v. 56, n. 3, p. 403–415, 2011. ISSN 1572-834X.

LIU, Y. et al. A simulation framework for industrial wireless networks and process control systems. In: IEEE WORLD CONFERENCE ON FACTORY COMMUNICATION SYSTEMS, 12., 2016, Aveiro. **Proceedings [...]**. Aveiro: IEEE, 2016. p. 1–11.

LU, Y. et al. Energy-efficient depth-based opportunistic routing with Q-Learning for underwater wireless sensor networks. **Sensors**, v. 20, n. 4, 2020. ISSN 1424-8220.

MADDUMA-BANDARAGE, R. D. K. **Design and Analysis of Routing and Scheduling Algorithms for Industrial Wireless Sensor Networks**. 113 f. Dissertation (Master in Engineering) — University of Calgary, Calgary, 2020.

MAHRENHOLZ, D.; IVANOV, S. Real-time network emulation with NS-2. In: IEEE INTERNATIONAL SYMPOSIUM ON DISTRIBUTED SIMULATION AND REAL-TIME APPLICATIONS, 8., 2004, Chicago. **Proceedings [...]**. Chicago: IEEE, 2004. p. 29–36.

MAMMERI, Z. Reinforcement learning based routing in networks: Review and classification of approaches. **IEEE Access**, v. 7, p. 55916–55950, 2019. ISSN 2169-3536.

MEMON, A. A.; HONG, S. H. Minimum-hop load-balancing graph routing algorithm for wireless hart. **International Journal of Information and Electronics Engineering**, v. 3, n. 2, p. 221–225, 2013. ISSN 2010-3719.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. USA: John Wiley & Sons, Inc., 2006. ISBN 0470088109.

NIU, J. et al. R3e: Reliable reactive routing enhancement for wireless sensor networks. **IEEE Transactions on Industrial Informatics**, v. 10, n. 1, p. 784–794, Feb 2014. ISSN 1551-3203.

NOBRE, M.; SILVA, I.; GUEDES, L. A. Reliability evaluation of wirelesshart under faulty link scenarios. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS, 12., 2014, Porto Alegre. **Proceedings [...]**. Porto Alegre: IEEE, 2014. p. 676–682.

NOBRE, M.; SILVA, I.; GUEDES, L. A. Performance evaluation of WirelessHART networks using a new network simulator 3 module. **Computers & Electrical Engineering**, v. 41, p. 325–341, 2015. ISSN 0045-7906.

NOBRE, M.; SILVA, I.; GUEDES, L. A. Routing and scheduling algorithms for WirelessHART networks: A survey. **Sensors**, v. 15, n. 5, p. 9703–9740, 2015. ISSN 1424-8220.

OSTERLIND, F. et al. Cross-level sensor network simulation with cooja. In: IEEE INTERNATIONAL CONFERENCE ON LOCAL COMPUTER NETWORKS, 31., 2006, Tampa. **Proceedings [...]**. Tampa: IEEE, 2006. p. 641–648.

SAVAGLIO, C. et al. Lightweight reinforcement learning for energy efficient communications in wireless sensor networks. **IEEE Access**, Mar 2019.

SEPULCRE, M.; GOZALVEZ, J.; COLL-PERALES, B. Multipath qos-driven routing protocol for industrial wireless networks. **Journal of Network and Computer Applications**, v. 74, n. Supplement C, p. 121–132, 2016. ISSN 1084-8045.

SHA, M. et al. Empirical study and enhancements of industrial wireless sensor-actuator network protocols. **IEEE Internet of Things Journal**, v. 4, n. 3, p. 696–704, June 2017. ISSN 2327-4662.

SHEN, W. et al. Prioritymac: A priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks. **IEEE Transactions on Industrial Informatics**, v. 10, n. 1, p. 824–835, Feb 2014. ISSN 1551-3203.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. Cambridge, MA, USA: A Bradford Book, 2018. ISBN 0262039249.

TOZER, B.; MAZZUCHI, T.; SARKANI, S. Many-objective stochastic path finding using reinforcement learning. **Expert Systems with Applications**, v. 72, n. Supplement C, p. 371–382, 2017. ISSN 0957-4174.

VARGA, A.; HORNIG, R. An overview of the OMNeT++ simulation environment. In: INTERNATIONAL CONFERENCE ON SIMULATION TOOLS AND TECHNIQUES FOR COMMUNICATIONS, NETWORKS AND SYSTEMS AND WORKSHOPS, 1., 2008, Marseille. **Proceedings [...]**. Marseille, 2008.

WANG, K.; CHAI, T. Y.; WONG, W.-C. Routing, power control and rate adaptation: A Q-Learning-based cross-layer design. **Computer Networks**, v. 102, p. 20–37, 2016. ISSN 1389-1286.

WANG, Y.; BARAC, F. Implementation of the WirelessHART MAC layer in the OPNET simulator. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND NETWORK TECHNOLOGY, 6., 2013, Paris. **Proceedings [...]**. Paris: IEEE, 2013. p. 663–668.

WINTER, J. M. et al. Towards a WirelessHART network with spectrum sensing. **IFAC Proceedings Volumes**, v. 47, n. 3, p. 9744–9749, 2014. ISSN 1474-6670.

WINTER, J. M. et al. Wireless coexistence and spectrum sensing in industrial internet of things: An experimental study. **International Journal of Distributed Sensor Networks**, SAGE Publications Sage UK: London, England, v. 11, n. 11, 2015.

WU, C. et al. Maximizing network lifetime of WirelessHART networks under graph routing. In: IEEE INTERNATIONAL CONFERENCE ON INTERNET-OF-THINGS DESIGN AND IMPLEMENTATION, 1., 2016, Berlin. **Proceedings [...]**. Berlin: IEEE, 2016. p. 176–186.

WU, C. et al. **Conflict-Aware Real-Time Routing for Industrial Wireless Sensor-Actuator Networks**. Washington, 2015.

XU, L. D.; HE, W.; LI, S. Internet of things in industries: A survey. **IEEE Transactions on Industrial Informatics**, v. 10, n. 4, p. 2233–2243, Nov 2014. ISSN 1551-3203.

YAU, K.-L. A.; KOMISARCZUK, P.; TEAL, P. D. Review: Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. **Journal of Networks and Computer Applications**, Academic Press Ltd., London, UK, v. 35, n. 1, p. 253–267, jan. 2012. ISSN 1084-8045.

YE, D.; ZHANG, M.; YANG, Y. A multi-agent framework for packet routing in wireless sensor networks. **Sensors**, v. 15, n. 5, p. 10026–10047, 2015. ISSN 1424-8220.

ZAND, P. et al. A distributed management scheme for supporting energy-harvested i/o devices. In: IEEE EMERGING TECHNOLOGY AND FACTORY AUTOMATION, 7., 2017, Barcelona. **Proceedings [...]**. Barcelona: IEEE, 2014. p. 1–10.

ZAND, P. et al. Implementation of WirelessHART in the NS-2 simulator and validation of its correctness. **Sensors**, v. 14, n. 5, p. 8633–8668, 2014. ISSN 1424-8220.

ZHANG, Q. et al. Reliable and energy efficient routing algorithm for WirelessHART. In: SUN, X.-h. et al. (Ed.). **Algorithms and Architectures for Parallel Processing**. Cham: Springer International Publishing, 2014. p. 192–203.

ZHANG, S.; YAN, A.; MA, T. Energy-balanced routing for maximizing network lifetime in WirelessHART. **International Journal of Distributed Sensor Networks**, v. 9, n. 10, p. 173–185, 2013.