3R: a reliable multi agent reinforcement learning based routing protocol for wireless medical sensor networks.

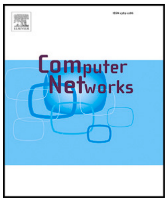
HAJAR, M.S., KALUTARAGE, H.K. and AL-KADRI, M.O.

2023

Contents lists available at ScienceDirect

Computer Networks

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



3R: A reliable multi agent reinforcement learning based routing protocol for wireless medical sensor networks

Muhammad Shadi Hajar [*], Harsha Kumara Kalutarage, M. Omar Al-Kadri

School of Computing, Robert Gordon University, Garthdee Road, Aberdeen, AB10 7GJ, Scotland, UK

ARTICLE INFO

Keywords:
Routing
Reinforcement learning
Trust management
Energy
Blackhole attacks
Selective forwarding attacks
Sinkhole attacks
On-off attacks
Security
Q-learning

ABSTRACT

Interest in the Wireless Medical Sensor Network (WMSN) is rapidly gaining attention thanks to recent advances in semiconductors and wireless communication. However, by virtue of the sensitive medical applications and the stringent resource constraints, there is a need to develop a routing protocol to fulfill WMSN requirements in terms of delivery reliability, attack resiliency, computational overhead, and energy efficiency. This paper proposes 3R, a reliable multi agent reinforcement learning routing protocol for WMSN. 3R uses a novel resource-conservative Reinforcement Learning (RL) model to reduce the computational overhead, along with two updating methods to speed up the algorithm convergence. The reward function is re-defined as a punishment, combining the proposed trust management system to defend against well-known dropping attacks. Furthermore, an energy model is integrated with the reward function to enhance the network lifetime and balance energy consumption across the network. The proposed energy model only uses local information to avoid the resource burdens and the security concerns of exchanging energy information. Experimental results prove the lightweightness, attacks resiliency and energy efficiency of 3R, making it a potential routing candidate for WMSN.

1. Introduction

Wireless Medical Sensor Network (WMSN) offers innovative applications to the healthcare field ranging from providing monitoring tools to sense the body's physiological signs to drug delivery. This revolutionized technology provides a potential solution to ease patients' lives, meet aging population healthcare needs, and support overloaded medical staff. However, despite the rapid development of this emerging technology, security concerns are still holding back the wide adoption [1]. Any security breach may disrupt the network operation and threaten the patient's life.

WMSN comprises a set of tiny bio-medical Sensor Nodes (SNs) distributed on the body surface, inside the body, or in the vicinity of the body where one of them acts as a sink. The wireless nature and the critical applications provided by WMSN make it vulnerable to a variety of security attacks and misconduct activities, the most important of which are the packet dropping attacks. These kinds of attacks are called internal attacks because they are launched by the Sensor Nodes (SNs) themselves for different reasons. For instance, an attacker could compromise a functioning SN and launch dropping attacks to disrupt the overall network operations. Another example is when an SN acts selfishly or gets overloaded and stops relaying packets for others with a view to conserving energy or gaining extra resources unfairly [2]. In both cases, the consequences would be detrimental and could endanger the patient's life. Moreover, many dropping attacks discussed in the literature have different characteristics and dropping patterns, such as selective forwarding [3], blackhole [4], and sinkhole [5]. It is worth mentioning that the proposed methods address various dropping patterns regardless of the underlying reason, whether it is due to an attack by malicious nodes or misbehavior of selfish or overloaded nodes.

In addition to the security concerns inherited from Wireless Sensor Networks (WSNs), WMSN has additional unique characteristics, such as resource constraints, critical applications, network topology, and low traffic rates. While routing in WSN is still challenging, with much research is being put forward constantly to produce an efficient routing protocol [6], designing a suitable routing protocol for WMSN is even more challenging, considering its unique characteristics. Reinforcement Learning (RL) based routing protocols have been introduced in the literature to address the routing problem in WSN [7–9]. Although this approach allows SNs to learn the optimal path to the destination, it has few limitations. To the best of our knowledge, the learning agent in all these proposed schemes has to receive a reward for each sent packet and then update its estimation to find the optimal path for future packets. This mechanism is voracious in terms of resource consumption and may not fit the resource-constrained SNs of WMSNs.

[*] Corresponding author.
E-mail addresses: m.hajar@rgu.ac.uk (M.S. Hajar), h.kalutarage@rgu.ac.uk (H.K. Kalutarage), omar.alkadri@bcu.ac.uk (M.O. Al-Kadri).

https://doi.org/10.1016/j.comnet.2023.110073
Received 1 February 2023; Received in revised form 31 August 2023; Accepted 17 October 2023
Available online 21 October 2023

Moreover, choosing the lowest cost path does not guarantee delivery reliability as the chosen path may contain one or more malicious nodes. Therefore, in our proposed 3R, a novel RL model is used to produce a lightweight, efficient routing protocol. Moreover, an effective Trust Management (TM) scheme is integrated with the 3R to ensure high delivery reliability. The reward function has been redefined as a punishment function based on the trustworthiness of potential routes. Furthermore, a novel energy model that only uses local information has been developed and integrated with the reward function to optimize the network lifetime and load balance the energy consumption across the network.

The main contribution of this paper is fourfold. First, the unique requirements of designing an efficient and reliable routing protocol for WMSN are specified. Second, proposing a resource-conservative RL model to overcome the WMSN resource limitations. Third, an efficient, lightweight, and reliable routing protocol based on the proposed RL model and combined with an effective trust management scheme is proposed. Fourth, a comprehensive analysis is carried out to prove the merit of our routing protocol against well-known dropping attacks.

The remainder of this paper is organized into five sections as follows. Related work is given in Section 2. The proposed routing protocol for WMSN is presented in Section 3, followed by evaluation and performance results in Section 4. Finally, Section 5 concludes the paper and highlights future work.

2. Related work

Routing is quite a challenging task in WMSN. The main challenge is to achieve reliable data delivery with minimum resource consumption in order to ensure high longevity of network operation [10]. Various routing protocols have been proposed in the literature to ensure reliable data transfer in WSN using different metrics and algorithms. However, only a few schemes targeted WMSN. Moreover, WMSN has unique characteristics and requirements, making inherited routing protocols from WSN not necessarily fit WMSN. Therefore, there is still an imperative research gap to design a routing protocol that fits WMSN and meets its requirements. Generally speaking, routing in WMSN could be classified into non-learning-based and learning-based protocols.

2.1. Non-learning routing protocols

Non-learning routing protocols involve using various methods to find the optimal path between the source and destination. Different classifications are proposed in the literature for this kind of routing methods [11,12]. Those methods could be classified into posture-based, thermal-aware, cluster-based routing and other routing approaches.

2.1.1. Posture-based routing

The posture-based routing is built on the body movements regularity assumption to analyze the network topology [13]. If the SN is able to predict its neighbors in a given time slot, efficient routing decision could be made to improve the data transmission rate and reduce the end-to-end latency [11]. In [14], the authors proposed a store-and-forward routing protocol for on-body SNs in Wireless Body Area Networks (WBAN). It is a distance vector routing protocol with a stochastic link cost. Although results showed an enhanced end-to-end delay, it is still a delay tolerant protocol that does not fit the critical application of WBAN. Moreover, the protocol is only proposed for on-body SNs. In [15], the authors proposed Network Management Cost Minimization for Dynamic Connectivity and Data Dissemination (NCMD) routing protocol to deal with high network dynamicity and reduce the network management cost. NCMD is proposed for on-body SNs to reduce the topology management overhead due to postural disconnections. However, the protocol complexity is high [11] and still regarded as a scope-specific protocol.

2.1.2. Thermal-aware routing

In thermal-aware routing, the nodes' temperatures are mainly used to evaluate the paths with a view to reducing nodes' temperature by avoiding high-temperature nodes. Authors in [16] proposed TARA, one of the early thermal aware routing protocols for implanted SNs to balance temperature rise caused by relaying activities. Another example is Reliability Enhanced-Adaptive Threshold based Thermal-unaware Energy-efficient Multi-hop ProTocol (RE-ATTEMPT) [17] where the authors propose a single-hop and multi-hop routing protocol to reduce the delay and energy consumption. RE-ATTEMPT is designed to address the main shortcomings of ATTEMPT routing protocol [18], such as unbalanced energy consumption and the inability to avoid dead nodes from the routing path. Although there are many routing proposals adopted this approach [19,20], the interest in this routing approach has decreased recently [11]. It is worth mentioning that both posture and thermal-based routing are regarded as scope-specific routing protocols.

2.1.3. Cluster-based routing

Cluster-based routing is another routing approach to tailor a routing protocol for WMSN. This method has been mainly proposed for WSN, where hundreds of nodes may exist in the network to reduce communication overhead. The Low Energy Adaptive Clustering Hierarchy (LEACH) [21] is the benchmark for this approach of routing with abundant proposed variants that have been surveyed in [22]. In this routing approach, the network is divided into clusters of nodes. Each cluster elects a cluster head to integrate and forward the information. For instance, the authors in [23] proposed a Clustering based Routing Protocol for wireless Body Area network (CRPBA) to enhance the energy consumption rate and prolong the network lifetime. CRPBA uses two methods to forward frames to the sink, direct forwarding and cluster based forwarding depending on the distance from the sink node and the frame data type. However, the energy of cluster head, which are close to the sink, is depleted fast causing network disruption. Although cluster-based routing approach is widely investigated in WSN for large networks, it may not fit WMSN where the maximum number of SNs is set to 64 [24].

2.1.4. Other routing approaches

In addition to the aforementioned approaches, there are some routing methods that do not fall within the previous categorization. For instance, in [25], the authors proposed independent multi-path routing protocol for WBAN. Another example is mobile sink routing protocols [11]. However, these approaches still do not meet the tough resource constraints and do not ensure reliable data delivery.

2.2. Learning-based routing protocols

Learning-based routing protocols mainly use Reinforcement Learning (RL) methods, especially Q-learning, to learn routing paths in WSN [26–29]; however, a few have targeted WMSN [30–32]. The reason could be attributed to the computational overhead incurred when adopting the traditional RL model to learn the network environment, which will be discussed further in Section 3 when proposing our RL model for routing applications. Researchers use different metrics to estimate routing decision cost, such as delivery latency, residual energy, and geographical distance [33]. However, this kind of metrics cannot deal with the free will of the other nodes. Relay nodes could get compromised or act selfishly and hence stop relaying packets for other nodes, which results in detrimental consequences. Therefore, there is a need to incorporate a security measure to avoid malicious paths. TMS provides an effective and robust measure to evaluate the trustworthiness of other nodes. To the best of our knowledge, only two schemes [29,32] are proposed in the literature that combine a TM scheme with a Q-learning routing model. Authors in [29] proposed ESRQ, a secure, lightweight routing scheme for WSN. However, it is unclear how the trust relationship is evaluated, which makes this scheme not reproducible due to missing details. Authors in [32] proposed QRT, a routing protocol designed for non-cooperative biomedical mobile wireless sensor networks. It has been proposed as an extension to RL-QRP [31] to deal with various kinds of misbehaving activities. The authors adopted the beta distribution trust scheme and integrated it with the Q-learning routing engine to produce a reliable routing protocol. However, proving its merit needs further investigation. Both ESRQ and QRT have not been thoroughly evaluated under different dropping attacks, especially on-off attacks. Moreover, all the proposed RL-based routing protocols in the literature use the same traditional RL model, which is a resource-consuming model and is not suitable for deployment on resource constrained SNs.

3. Protocol design

In this section, the proposed routing protocol for WMSN is discussed in detail. The design starts by presenting the network and threat models, which leads to specifying the protocol designing requirements. Our novel RL model to produce a lightweight routing protocol is then discussed. The delivery reliability is achieved by integrating our proposed TM scheme into the routing decision engine [34]. Moreover, two updating mechanisms have been proposed to accelerate the algorithm convergence as well as conserve resources.

3.1. Network model

WMSN consists of a set of bio-sensor nodes that could be placed on the body surface, inside the body, or off the body. These SNs have the ability to sense the body's physiological signals, such as body temperature, glucose levels, Electrocardiogram (ECG), and pulse rate. However, SNs have strict resource limitations that impose further constraints in adopting security countermeasures. For example, the lithium iodide cell battery of the pacemaker is meant to last for seven years before it gets replaced via surgery [35]. Therefore, lightweight countermeasures and protocols are essential to extend the battery life and avoid unnecessary surgical complications. All sensed information is forwarded to the sink node, which in turn forwards them to the remote medical server where physicians can monitor, analyze and even intervene when necessary.

Field hospitals are temporary hospitals set up due to civil emergencies, such as battlefields, disease outbreaks, and pandemics. For example, many field hospitals have been established in many parts of the world during the ongoing COVID-19 pandemic, especially in developing countries. In our experiments, the topology of a wireless medical sensor network of a field hospital ward is adopted. Fig. 1, shows the simulated ward in our experiments, which is $50\ \mathrm{m} \times 10\ \mathrm{m}$ where patient beds are distributed in an efficient way to save physical space and provide an adequate space to care at the same time. A maximum number of 64 SNs can be accommodated in this medical unit in compliance with IEEE 802.15.6 standard [24]. The network topology is a multi-hop star topology where SNs sense various bio-signals and forward them to the sink node. The communication range of the SNs is $5\ \mathrm{m}$; hence, SNs relay frames for other adjacent nodes. Therefore, an efficient, lightweight, and reliable routing protocol is required to forward the frames from the sensing units to the sink node, which in turn forwards them to the medical server.

3.2. Threat model

Due to the sensitive nature of the WMSN applications and the broadcast nature of the wireless communication, many potential threats may disrupt the network operation and endanger the patients' lives. Threats can be classified into internal and external. External threats could be defeated by deploying cryptographic security measures, such as authentication and encryption. Our proposed ecosystem assumes that secure mutual authentication is achieved and security keys are established. On the other hand, internal threats are difficult to defeat as they could be launched by legitimate nodes that have successfully got authenticated and may have a copy of the security keys. Therefore, this work aims to demonstrate the effectiveness of our 3R against packet dropping attacks, one of the devastating internal threats on WMSN.

Packet-dropping attacks are regarded as one of the most devastating internal attacks because of their consequences on the patient's life. For instance, a malicious node could drop a command sent by a physician to an insulin pump to release the insulin dose into the bloodstream. In addition, dropping could occur due to malicious activities like when a node got compromised, selfish behavior when a node acts selfishly with a view to saving resources, or when packets pass through overloaded nodes. Adversaries could launch different kinds of dropping attacks or may change the dropping patterns with a view to keeping themselves undetected. 3R protocol is evaluated for various kinds of dropping attacks with different parameter settings, such as blackhole, sinkhole, selective forwarding, and on-off attacks, which will be discussed in detail in Section 4.

3.3. 3R design requirements

Various objectives have been considered when designing 3R. These objectives include efficiency, lightweightness, scalability, and resiliency.

Efficiency is the first objective of designing a routing protocol. Ensuring a high packet delivery ratio is a must for any routing protocol. However, choosing the optimal path between the sender and receiver determines the routing protocol's efficiency, which is a crucial requirement for resource-constrained devices, such as SNs. The lowest cost path must always be chosen to ensure high efficient routing protocol. Radio Frequency (RF) activities, especially transmission (TX), constitute around 80% of the consumed energy [36]. In order to reduce the consumed energy, SNs must always choose the shortest path in order to reduce the number of transmissions. Therefore, 3R has been designed to always choose the shortest reliable path regardless of the network size, nodes deployment, or traffic rate.

Lightweightness is a key requirement to fit the strict resource constraints of SNs. All proposed Q-learning-based routing protocols in the literature consider transmitting one packet as a complete action, which calls for updating the Q-table for each sent or forwarded packet [9,31,32,37]. This method is a resource-consuming process, particularly when more packets are generated or forwarded. Therefore, in 3R, the RL model has been reformulated to consume less memory and processing resources.

Scalability is another requirement. In a multi-agent environment, each agent has to consider the actions of other agents, which causes a scalability problem when the number of agents increases as the action space grows exponentially [38]. Moreover, agents in a networked environment suffer from the partial observability problem as they do not have a full view of the network. Therefore, decentralized learning with a networked agent approach [39] was adopted in 3R to enable the learning agents to collaborate with their neighbors by sharing information. This approach is regarded as a solution to the poor scalability of fully centralized learning, and centralized training with decentralized execution approaches [38]. In addition, 3R has been evaluated for variable traffic rates and the maximum number of SNs in the network as defined in IEEE 802.15.6 [24].

Attack resilience is the most challenging task in designing a reliable routing protocol for WMSN. Dropping attacks could be catastrophic not only for the network operation but also for the patients. Authors in [40], investigated the performance of Routing Protocol for Low-Power and Lossy Networks (RPL), which is one of the candidate routing protocols for low-power and lossy networks, under blackhole attacks. The results indicated a significant data loss. Therefore, 3R has been designed to resist all kinds of known dropping attacks. Moreover, it is also resilient to route poisoning attacks.

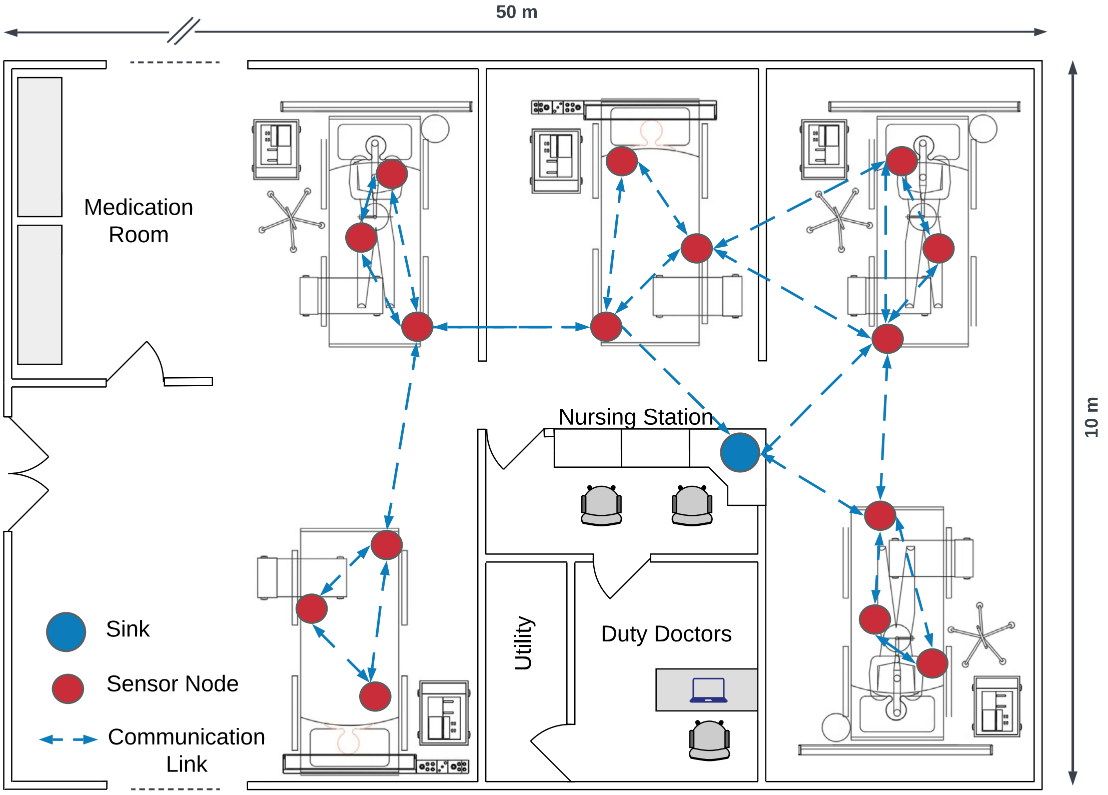


Fig. 1. Network model.

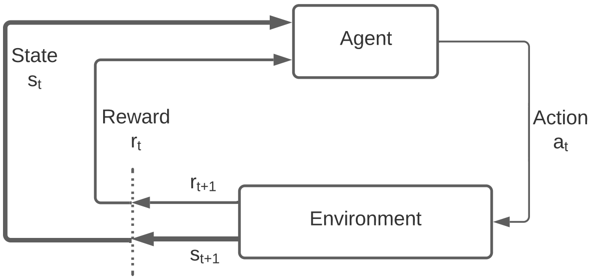


Fig. 2. Traditional RL model.

3.4. Multi agent reinforcement learning

Reinforcement Learning (RL) is an area of machine learning that focuses on how intelligent agents interact with an environment through a series of state–action pairs to maximize the cumulative rewards. In Multi-Agent Reinforcement Learning (MARL), many agents interact with a mutual environment and with each other to achieve a particular goal [38]. This interaction could be a collaboration to accomplish a common task, a competition to accomplish a self-goal, or a mix of both, like when members of two teams of a game collaborate with each other and at the same time compete with the other team.

In the traditional RL model, as illustrated in Fig. 2, at each time step t, the RL agent in an environment's state $s_t \in \mathbb{S}$ chooses an action $a_t \in \mathbb{A}$, which causes the environment to move to state $s_{t+1} \in \mathbb{S}$ and the agent to receive a reward $r_{t+1} \in \mathbb{R}$.

In routing applications, the agent learns a routing policy that chooses the optimal path to the destination by experimenting different actions and gathering evidence from the environment. The learning process in such a case must be online and continual due to the dynamicity of the network. The learned routing policy specifies the optimal adjacent node for each agent to forward its frames to. This routing policy is constantly updated to reflect any change in the network.

Q-learning is an off-policy, value-based, model-free reinforcement learning algorithm to evaluate the value of an action in a particular state [8]. Each agent maintains a Q-values table of $|\mathbb{S}| \times |\mathbb{A}|$ represents the expected long-term rewards if the agent takes the action a_t at the state s_t.

3.5. The proposed synchronous RL model

With the aforementioned design requirements in mind, 3R is built using the Q-learning algorithm, incorporating the proposed trust management scheme in Section 3.7 to ensure reliable data delivery. The learning agent is modeled as 3-tuple $(\mathbb{S}, \mathbb{A}, \mathbb{R})$. WMSN network represents the environment $\mathbb{E}$, which includes SNs that exchange messages where one of them acts as a sink S. Each state $s \in \mathbb{S}$ represents an SN. The action $a \in \mathbb{A}$ is defined as selecting the next forwarder to relay packets to a destination. The learning agent receives a reward $r_{t+1} \in \mathbb{R}$ for each action a_t.

3R defines $Q^i_{t+1}(s^i_t, a^i_t)$, which is the updated Q value of node i, given the state s^i_t and the action a^i_t, as the estimated future reward. Each learning agent maintains a Q-table, which gets updated once the agent performs an action a_t and observes the reward r_{t+1} as in Eq. (1).

$$Q^i_{t+1}(s^i_t, a^i_t) \leftarrow (1-\eta)Q^i_t(s^i_t, a^i_t) + \eta[r^i_{t+1}(s^i_{t+1}) + \gamma \max_{a \in A} Q^i_t(s^i_{t+1}, a^i_t)] \tag{1}$$

where $\eta \in [0,1]$ is the learning rate where small values of it cause long learning time and large values may cause oscillations, $\gamma \in [0,1]$ is the discount factor for the future rewards where small values of it make the agent myopic and cares more about the immediate rewards. In order to ensure reliable forwarding, trust is incorporated in estimating the reward. This makes the learning agent chooses the optimal reliable path. Moreover, the reward calculation is defined as a punishment to

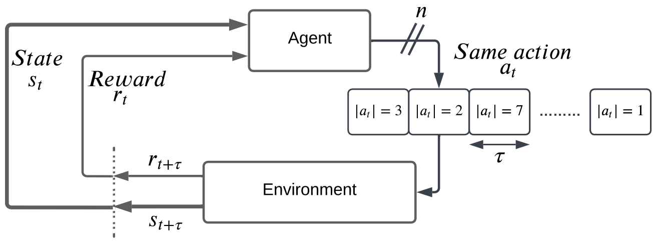


Fig. 3. Graphical representation of the proposed RL model.

force the learning agent to choose the shortest path to the destination, as shown in Eq. (2).

$$r^i_{t+1}(s^i_{t+1},j)=\begin{cases}-(1-T^{ij}_t).F^{(i)}_t & if \quad O^{ij}_t \neq \{\phi\}\\ -(1-T^{ij}_{t-\delta}).F^{(i)}_t & if \quad O^{ij}_t = \{\phi\} \wedge |O^{ij}| > \epsilon\\ 0 & Otherwise\end{cases} \tag{2}$$

where $r^i_{t+1}(s^i_{t+1},j)$ is the new reward received by node i which chose node j as a forwarder at the end of the time unit t, T^{ij}_t is the trust value maintained by node i for node j at time unit t, δ is a time lag used to get the last evaluated trust value, O^{ij}_t is the observations maintained by node i for node j at time unit t, ϵ is the threshold to specify the minimum required evidence and $F^{(i)}_t$ is the computed energy parameter as detailed in Section 3.8. The trust value T^{ij}_t is computed using Algorithm 4 as detailed in Section 3.7.

To the best of our knowledge, 3R is the first RL model using the time window technique to reduce the computational overhead of the traditional RL model, depicted in Fig. 2. 3R reformulated the RL model, assuming that the network will be static for a short period, which is an acceptable assumption as nodes could be regarded as stationary for a short interval. This assumption allows the learning agent to perform the same action multiple times during a short period of time before receiving the corresponding reward. Adopting this method significantly reduces the computational overhead by periodically updating the Q tables. The actions and rewards are re-defined in the proposed RL model in which the agent performs the same action a_t during the time unit t and gets its reward $r_{t+\tau}$ at the end of the time unit at $t+\tau$ as illustrated in Fig. 3. In the traditional RL model, the learning agent needs to observe the reward and update its Q-table for each packet, while 3R evaluates the reward and updates the Q-table after a defined time unit τ in order to reduce the computational overhead. This proposed method is referred to as synchronous updating. Moreover, asynchronous updating is also used in 3R to help the algorithm to converge swiftly, which will be elaborated further in Section 3.6.

The routing task must be achieved in a distributed manner as no agent has a full view of the network states. Therefore, 3R uses decentralized learning where the RL agents exchange their best Q values with their neighbors as detailed in Algorithm 1. The exchanged values are then used to update the Q-table and determine the best forwarder to the destination. Once the following action is taken, it changes the environment states, requiring periodic updates. Actions should not be greedily selected all the time for two reasons. First, routing is an online continual learning task. Second, exploiting the best action prevent the algorithm from converging to the global optimum. Therefore, ϵ−greedy strategy [41] is used to explore the environment with a probability of θ and exploit the best action with a probability of $(1-\theta)$. During the exploration phase, a random action $a^i_t \in \mathbb{A}$ is selected to search for possible alternative paths. At the beginning, 3R has no knowledge about the environment; hence the future rewards are initialized to zero for each neighbor $n^i \in N^i_t$, which is more realistic and requires no additional hardware or pre-configuration like those introduced in [31,32], where the authors used positioning information.

Algorithm 1: 3R protocol for making routing decisions

```
Input:
The reward: $r^i_{t+1}(s^i_{t+1}, j)$
The Q table: $Q_t$
The trust table: $T_t$
Output: The optimal next hop
initialization:
$Q^i_0(n^i \in N^i_t) = \begin{cases} 0 & if \quad n^i \neq S \\ 1 & if \quad n^i = S \end{cases}$
$T^i_0(n^i \in N^i_t) = 0.5$
    $a^i_1 = \begin{cases} S & if \quad S \in N^i \\ n^i & | \quad n^i \in N^i \end{cases}$
while TRUE do
    $Wait \quad \tau$
    $Broadcast \quad max(Q^i_t)$
    $\forall j \in N^i$ , $update(Q^{ij}_t)$ using Eq. (1)
    if $\epsilon - greedy > \theta$ then
        $a^i_{t+1} \leftarrow n^i_t \mid n^i_t \in N^i_t$
    else
        $a^i_{t+1} \leftarrow \underset{n^i_t \in N^i_t}{argmax}\, Q^i_t(s^i_t, a^i_t)$
    end
end
```

3.6. Updating methods

In the 3R routing protocol, two types of Q-table updating methods are employed to decrease resource consumption and improve algorithm convergence, as demonstrated in Algorithm 2. Synchronous updating is utilized to update the Q table at the end of each time unit with a view to reducing the processing overhead. As the action in our model consists of multiple sub-actions on a predefined time unit, the learning agent performs the same sub-action multiple times during a period τ, which means all packets will be forwarded to the same next hop. Meanwhile, the agent is observing the behavior of its next hop to evaluate its trustworthiness. By the end of the time unit, the agent is able to evaluate the trust value at time t and gets its reward $r^i_{t+1}(s^i_{t+1})$.

Each agent broadcasts its best estimation to adjacent nodes periodically. These broadcasted estimations are then used to update the Q table using the gained reward as in Eq. (1). However, as each agent only forward packets to one node during the time unit, it will not get rewards for other adjacent nodes, but it could receive an updated estimation from them. For instance, node i has $a^i_t = j$ at time t and receives updates from nodes j and k.

There are two cases for updating the Q-values of all adjacent nodes as shown in the synchronous part of Algorithm 2. The first case arises when $j == a^i_t$, indicating that node j was the selected node for forwarding the traffic during the previous time unit. Consequently, Q^{ij}_t will be updated using the received reward $r^i_{t+1}(s^i_{t+1})$ using Eq. (1). The second case involves updating the Q-values for other nodes, such as node k. In this case, the routing agent does not receive any reward from the network environment and relies solely on its most recently received reward to update the Q-values. In this case, 3R checks how certain it is about node k by checking the number of recent observations. If node i has adequate observations about node k, it will use the most recent reward $r^i_{t-\delta}(s^i_{t-\delta}, k)$ to update the Q^{ik}_t. Otherwise, it will ignore the received estimation and keep the Q value unchanged $Q^{ij}_{t+1} \leftarrow Q^{ij}_t$. This technique immunizes 3R from adopting fake second-hand information without being certain enough about the sender's trustworthiness. Moreover, it allows the protocol to respond quickly to network dynamicity.

Algorithm 2: Synchronous and asynchronous Q table updating

```
Input:
The Q table: $Q^i_t$
The reward: $r^i_{t+1}(s^i_{t+1}, j)$
The trust table: $T_t$
Output: Updated Q Table: $Q^i_{t+1}$
if Synchronous Update then
    foreach $j \in N^i_t$ do
        if $j == a^i_t$ then
            update $Q^{ij}_t$ using $r^i_{t+1}(s^i_{t+1}, j)$
        else
            if $|O^{ij}| > \epsilon$ then
                update $Q^{ij}_t$ using recent $r^i_{t-\delta}(s^i_{t-\delta}, j)$
            else
                $Q^{ij}_{t+1} \leftarrow Q^{ij}_t$
            end
        end
    end
end
if Asynchronous Update then
    if $\eta == 1$ then
        $r^i_{t+1}(s^i_{t+1}, j) = -e^{\eta}(1 - T^{ij}_t)$
    else
        $r^i_{t+1}(s^i_{t+1}, j) = -(1 - T^{ij}_t)$
    end
    if $RQ^i_{t-1}(s^i_{t-1}, j)$ then
        update $Q^{ij}_t$ using $r^i_{t+1}$ and $RQ^i_{t-1}(s^i_{t-1}, j)$
    else
        $Q^i_{t+1}(s^i_t, a^i_t = n_j) \leftarrow Q^{ij}_t - \zeta$
    end
    $a^i_t \leftarrow \underset{n^i_t \in N^i_t}{argmax}\ Q^i_t(s^i_t, a^i_t)$
end
```

On the other hand, although the proposed synchronous updating is very resource-efficient, as presented in the next section, it could be slow to converge and may need more learning time as the learning agent could keep forwarding packets to the wrong path for the whole time unit. This usually happens if loops occur when the learning agent is exploring the network. Unlike traditional learning model where the learning agent risks losing one packet for each exploring step, the synchronous updating model could lose more packets because it keeps forwarding packets to one next-hop during one time unit. Therefore, 3R introduces a loop detection and avoiding algorithm as shown in Algorithm 3. When a loop is detected, such as when node i receives its own transmitted packet $P^{id}_{t+\delta}$ after a time lag of δ, or when there is a potential for a loop to occur, such as when a node receives a packet P^{jd}_t to forward from its designated forwarder $a^i_t = j$ at time unit t, the asynchronous update is triggered, as illustrated in the asynchronous section of Algorithm 2. During this update, the corresponding Q-value is penalized either by updating Q^{ij}_t using r^i_{t+1} and $RQ^i_{t-1}(s^i_{t-1}, j)$ or by subtracting the loop penalizing parameters ζ from $Q^i_{t+1}(s^i_t, a^i_t = n_j)$. This adjustment allows the protocol to choose an alternative, promising next hop. Through this technique, 3R gains the ability to operate efficiently and achieve rapid convergence. For the definitions of all symbols used in this paper, please refer to Table 1.

Algorithm 3: Loop processing

```
Input: A packet to forward: $P^{sd}_t$
Output: Updated Routing
while TRUE do
    if $\forall\ i \in \mathbb{N}$ receives $P^{id}_{t+\delta}$ then
        Asynchronous Q table update as in Algorithm 2
        $a^i_t \leftarrow \underset{n^i_t \in N^i_t}{argmax}\ Q^i_t(s^i_t, a^i_t)$
        Update $P^{id}_t$
        Send $P^{id}_t$
    end
    if $\forall\ i \in \mathbb{N}$ receives $P^{jd}_t \wedge a^i_t = j$ then
        Asynchronous Q table update as in Algorithm 2
        $a^i_t \leftarrow \underset{n^i_t \in N^i_t}{argmax}\ Q^i_t(s^i_t, a^i_t)$
        Forward $P^{jd}_t$
    end
end
```

Table 1
Symbols used in this paper.

Symbol	Meaning
S	The sink node
$\mathbb{N}$	The set of all sensor nodes in the network
$\eta \in [0, 1]$	The learning rate parameter
$\gamma \in [0, 1]$	The discount factor
$Q^i_{t+1}(s^i_t, a^i_t)$	The updated Q values of node i, given the state s^i_t and the action a^i_t
$r^i_{t+1}(s^i_{t+1})$	The new reward received at the end of the time unit
T^{ij}_t	The trust value maintained by i of node j at time t
ϵ	A threshold to specify the minimum required evidence
τ	The time window [s]
N^i_t	The neighbors of node [i] at time t
$n^i \in N^i_t$	A neighbor of node i
$s^i_t \in \mathbb{S}$	The state of node i at time window t
$a^i_t \in \mathbb{A}$	The taken action by node i at the time window t
Q^{ij}_t	The Q value maintained by node i for node j at the time window t
θ	The exploration rate
O^{ij}	The observations maintained by node i for node j
$RQ^i_{t-1}(s^i_{t-1}, j)$	The last expected future reward received from node j
μ	The traffic rate
δ	A time lag
$\zeta \in]0, 1]$	A loop penalizing parameter
$\alpha,\ \beta \in\ [0, 1]$	The beta distribution levels
$b_t,\ d_t$	The slopes at time t
λ	The longevity factor

3.7. Trust evaluation

3R incorporates a trust management scheme as a security countermeasure to ensure reliable data transfer. Several TM schemes have been evaluated to choose the best candidate. LTMS [34] has been adopted for mainly two reasons. First, it has been developed to fit WMSN requirements. Second, it is an attack-resistant TM scheme. LTMS is a distributed trust evaluation scheme where each node has its trust evaluation engine as shown in algorithm 4. LTMS evaluates the forwarding service of adjacent nodes with a view to differentiate between trustworthy and untrustworthy ones. LTMS encompasses two algorithms aimed at ensuring a secure trust evaluation process. The initial algorithm, LTMS(1), is employed to assess the direct trust relationships among SNs. In contrast, the second algorithm, LTMS(2), offers an additional layer of defense against on-off attacks. Both of these algorithms are combined in Algorithm 4. In this algorithm, α and β stand for the levels of the beta probability distribution, while b_t and d_t represent the slopes at the specific time unit, t. Moreover, $Rep_{ij}(t)$ signifies the trustor i's upheld reputation value for trustee j, and $thr1$ is the designated threshold to distinguish between trustworthy and untrustworthy SNs. It is noteworthy that this threshold is commonly set at 0.5 in relevant literature [34]. Furthermore, $thr2$ denotes the minimum level of trustworthiness expected from SNs during regular operation, and it is established at 0.85 according to [34]. The notation $ShRep_{ij}(t)$ is the short-term reputation value at the time unit t, while the parameters $cycle$ and $malicious$ play a pivotal role in detecting instances of on-off attacks.

LTMS promptly detect any changes in forwarding behavior through integrating the slopes b_t and d_t with beta distribution levels. This technique allows α_t to decrease and may accumulate negative values during the attack. At the same time, β_t develops a positive value, giving more weight to any misbehavior and making it harder to forget. As trust management schemes are vulnerable to on-off attacks where smart adversaries change their behavior between good and bad with a view to keeping themselves undetected. LTMS uses an on-off protection module designed to detect on-off attacks after evaluating Rep^{ij}_t. The on-off module in LTMS is designed to detect repeated attack patterns. It incorporates the short-term reputation value and long-term trust values along with the novel updating mechanism to defeat on-off attacks. This on-ff protection module is only triggered when an on-off attack is detected.

Algorithm 4: Secure Trust Evaluation

```
Input: Observations & beta shape parameters
Output: Trust value
initialization;
while TRUE do
    if $b_{t-1} \leq 0$ && $d_{t-1} > 0$ then
        $\alpha_t = \lambda(\alpha_{t-1} + b_{t-1}) + s_t$;
        $\beta_t = \lambda(\beta_{t-1} + d_{t-1}) + u_t$;
        $b_t = \alpha_t - \alpha_{t-1}$;
        $d_t = \beta_t - \beta_{t-1}$;
    else
        $\alpha_t = \lambda.\alpha_{t-1} + s_t$;
        $\beta_t = \lambda.\beta_{t-1} + u_t$;
        $b_t = \alpha_t - \alpha_{t-1}$;
        $d_t = \beta_t - \beta_{t-1}$;
    end
    if $\alpha_t \leq 0$ then
        $Rep^{ij}_t = 0$;
    else
        $Rep^{ij}_t = \frac{\alpha_t}{\alpha_t + \beta_t}$;
    end
    if $T^{ij}_{t-1} \geq thr_1$ && $Rep^{ij}_t < thr_1$ then
        if $malicious > 0$ then
            $cycle = t - malicious$;
            $malicious = 0$;
        else
            $malicious = t$;
        end
    end
    if $cycle > 0$ && $Trust(t-1) < thr_2$ then
        $ShRep^{ij}_t = mean(T^{ij}_{t-cycle:t})$;
        $T^{ij}_t = min(ShRep^{ij}_t, Rep^{ij}_t)$;
    else
        $T^{ij}_t = Rep^{ij}_t$;
        $cycle = 0$;
    end
end
```

3.8. Energy model

Optimizing the network lifetime is still a challenging concern in WSN and WMSN in particular. Due to the critical applications of WMSN, dead nodes may have catastrophic consequences. Moreover, in some cases, replacing the battery may need surgical intervention. Considering the residual energy of the adjacent nodes is widely used to maximize the overall network lifetime [42,43]. However, exchanging energy information between adjacent nodes is neither energy nor computational efficient. In contrast, 3R only uses local energy information with a view to reducing the computational overhead and avoiding filtering out false second-hand information. Moreover, it uses two sources of energy information with a view to load balancing energy consumption across the network. When the residual energy percentage is greater than a threshold ϑ, this parameter does not contribute in evaluating the consumed energy ratio $E^{(i)}_t \in [0,1]$ as shown in Eq. (3). In that case, SNs choose the most reliable shortest path, which in turn makes some nodes overloaded due to their trustworthiness and positions. Therefore, 3R defines the energy consumption ratio $C^{(i)}_t$ to evaluate the extra burden incurred by nodes due to relaying activities, as shown in Eq. (4). The weighted average of $E^{(i)}_t$ and $C^{(i)}_t$ is calculated in Eq. (5). As integrating the energy into the reward function may influence the nodes routing decision to choose a malicious path, the energy factor is bounded by $\lambda \in [0,1]$ as shown in Eq. (6).

$$E^{(i)}_t = \begin{cases} 0 & if \ \ \frac{e_{res}(t)}{e_{init}} > \vartheta \\ 1 - \frac{e_{res}(t)}{e_{init}} & Otherwise \end{cases} \tag{3}$$

$$C^{(i)}_t = 1 - \frac{c_n(t)}{c_a(t)} \tag{4}$$

$$\psi^{(i)}_t = \omega E^{(i)}_t + (1-\omega) C^{(i)}_t \tag{5}$$

$$F^{(i)}_t = e^{\lambda \psi^{(i)}_t} \tag{6}$$

where $e_{res}(t)$ is the remaining energy at time t, e_{init} is the initial energy, ϑ is the residual energy threshold, $c_n(t)$ is the node normal energy consumption rate, $c_a(t)$ is the overall energy consumption rate, ω is the average weight, λ is the bound parameter where $\lambda = 0$ is used to disable the energy module.

4. Evaluation and performance results

This section simulates and analyzes the 3R routing protocol. Various simulation scenarios have been considered using different parameters setting and under different dropping attacks.

4.1. Experimental setup

A WMSN of 64 SNs has been adopted to comply with IEEE 802.15.6 [24]. The SNs have been distributed randomly in an area of 50 m × 10 m mimicking a ward in a field hospital as shown in Fig. 1. One SN acts as a sink while other nodes have the ability to relay frames for other SNs. The traffic is generated using the exponential probability density function as shown in Eq. (7).

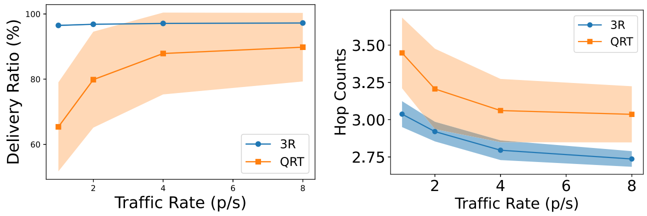
$$p(x;\mu) = \begin{cases} \mu e^{-\mu x} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{7}$$

where μ is the rate parameter and x is the time gap between two consecutive packets.

3R has been benchmarked with QRT [32], which is an extension to RL-QRP routing protocol [31] where the authors integrated a reputation and trust scheme to deal with non-cooperative and misbehaving nodes in biomedical sensor networks. QRT was the only available RL-based routing protocol in the literature designed for WMSN that incorporates the TM scheme to achieve reliable data delivery. In order to ensure a fair comparison between the two protocols, the reported parameters setting of QRT have been adopted. Table 2 shows the simulation parameters setting. The learning rate η and the discount factor γ have been set to 0.5. The experiments were carried out using a discrete event simulator based on Simpy [44]. The simulation time is 500 s, where the first 50 s is regarded as a training period unless otherwise indicated. This training period has been specified to allow QRT to converge, followed by a relatively long simulation time to study the stability of routing decisions of both protocols. During the simulation, the agents adopt the ε−greedy strategy to balance between exploration and exploitation where ε is set to 0.1 as in QRT. Each experiment has been repeated 30 times, and then the results have been averaged out and reported with one standard deviation. It is worth mentioning that when the sample size is 30, the sampling distribution approximates the Gaussian distribution [45].

Table 2
3R simulation parameters.

Parameter	Value
Application	Poisson random traffic
Exponential transmission interval μ	1, 2, 4, 8
Radio range	5 m
Propagation loss model	Range propagation loss
Number of SN	64
Time unit	1 s
Simulation time	500 s
Learning period	50 s
Learning rate η	0.5
Discount factor	0.5
ε−greedy	0.1



(a) (b)

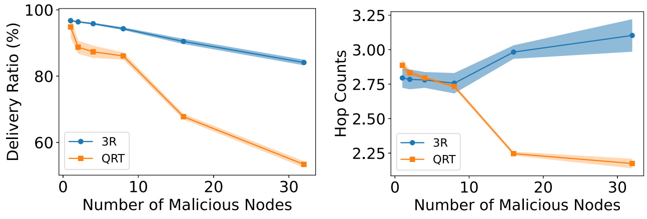
Fig. 4. The average delivery ratio and hop counts during normal operation for various traffic rates.

4.2. Normal operation

In this experiment, the performance of 3R has been evaluated, assuming that there are no malicious activities inside the network. Benign nodes randomly drop around 1% of the received packets to relay as WMSN is intolerant to higher rates of packet loss. This experiment aims to ensure that 3R chooses the optimal path to the destination with the highest delivery ratio. Some SNs generate low traffic rates around 1 packet/s, such as heart rate sensors [46]. Therefore, the experiment has been run for four different traffic rates starting from $\mu = 1$ p/s and doubling the traffic rate each time. Figs. 4(a) and 4(b) show the average delivery ratio, and the average hop counts with one standard deviation, respectively. The results show that 3R achieves the highest delivery ratio with minimal variability, while QRT did not work well for the lowest traffic rate with a delivery ratio of 75%. QRT's performance shows a slight improvement for traffic rates starting at $\mu = 2$ p/s to achieve around 90%; however, the high variability of the delivery ratio confirms that QRT struggles to converge. On the other hand, Fig. 4(b) reveals that 3R chooses the shortest path to the destination compared to QRT. This proves that although 3R updates the routing decision periodically, it performs efficiently thanks to its asynchronous updating methods to avoid bad routing decisions. It is worth noting that the performance is slightly enhanced for higher traffic rates because the learning agents can get more evidence from the environment to enhance their routing decisions.

4.3. Blackhole attacks

The Blackhole attack is a well-known attack in WMSN where compromised nodes drop all the received frames instead of forwarding them to the destination. This causes severe detrimental consequences, especially for medical applications [4,47]. In this experiment, the delivery ratio and the hop counts are evaluated under different blackhole attacks. The number of malicious nodes was doubled each time, starting from one and up to 50% of the total number of the SNs. The experiment was run for 30 times for each parameters setting, and then the results are averaged out and reported with one standard deviation as shown in Figs. 5(a) and 5(b). The results reveal a superior performance for 3R in contrast with QRT. Although QRT performed well when there is only one malicious node, the delivery ratio sharply dropped by introducing more malicious SNs to the network due to the inability to detect the malicious paths. In contrast, 3R showed a steady superior performance even when 50% of the SNs are malicious. It is worth mentioning that the slight decrease in the delivery ratio of 3R when increasing the number of malicious SNs is due to ε-greedy strategy where 10% of the actions are made randomly with a view to exploring the environment. On the other hand, the hop count results explain how each protocol responds to the hostile environment. Fig. 5(b) shows that 3R performs better when there are up to 8 malicious SNs. When the number of malicious nodes increases, 3R needs more hops to reach the destination to avoid malicious SNs. However, in QRT, the number of hops needed to get to the destination is decreased unexpectedly by increasing the number of malicious nodes, which explains the poor delivery ratio. These results indicate that QRT failed to build reliable paths that avoid malicious nodes and confirm that 3R chooses the most reliable shortest paths.



(a) (b)

Fig. 5. The average delivery ratio and hop counts evaluated under blackhole attacks, considering a range of varying percentages of malicious nodes.

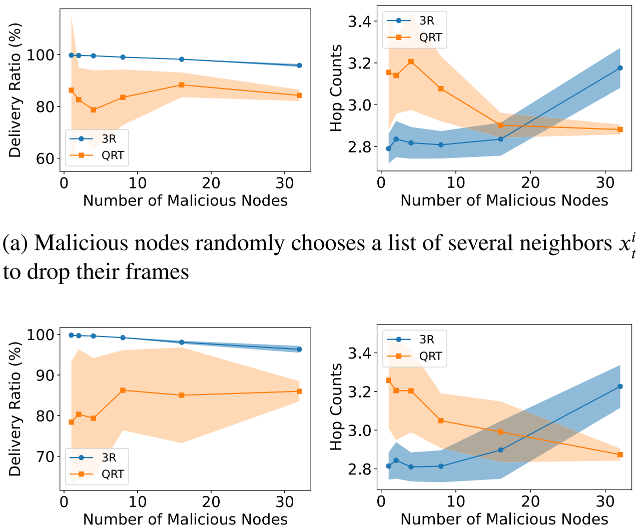
4.4. Selective forwarding attacks

In the selective forwarding attack, the malicious nodes forward some frames and drop others selectively [48,49]. This behavior is hard to detect as the same malicious node could be trustworthy for some nodes and untrustworthy for others. In this experiment, 3R has been evaluated under selective forwarding attack, where malicious nodes randomly choose a list of neighbors not to relay their frames. Two scenarios have been considered. In the first, the malicious node randomly chooses a list of several neighbors x_t^i to drop their frames as in Eq. (8):

$$|x_t^i| = \begin{cases} \frac{|N_t^i|}{2}, & if \quad |N_t^i| = 2k \\ \frac{|N_t^i|}{2} + 1, & if \quad |N_t^i| = 2k+1 \end{cases} \qquad where \quad k \in \mathbb{W} \tag{8}$$

This means that 50% to 67% of the received frames to relay will be dropped. In the second scenario, malicious nodes run volatile selective forwarding attacks by randomly changing x_t^i per 20% of the simulation time. Figs. 6(a) and 6(b) show the delivery ratio and the hop counts under both attack scenarios. Our proposed 3R protocol outperforms QRT in both scenarios and provides a reliable delivery with minimal variability. At the same time, QRT shows a high variability when the number of malicious nodes is less than 25% of the total number of SNs, indicating difficulty in converging. By increasing the number of malicious nodes, the delivery ratio of QRT decreases significantly.

On the other hand, the hop counts results shown in Figs. 6(a) and 6(b) reveal how each protocol responds to the hostile environment. 3R performs better when the number of malicious nodes is less than 25%. Moreover, when the number of malicious nodes reaches 50%, the hop count gradually increases to avoid any path through malicious nodes. It worth noting that increasing the number of malicious nodes



(a) Malicious nodes randomly chooses a list of several neighbors x_t^i to drop their frames

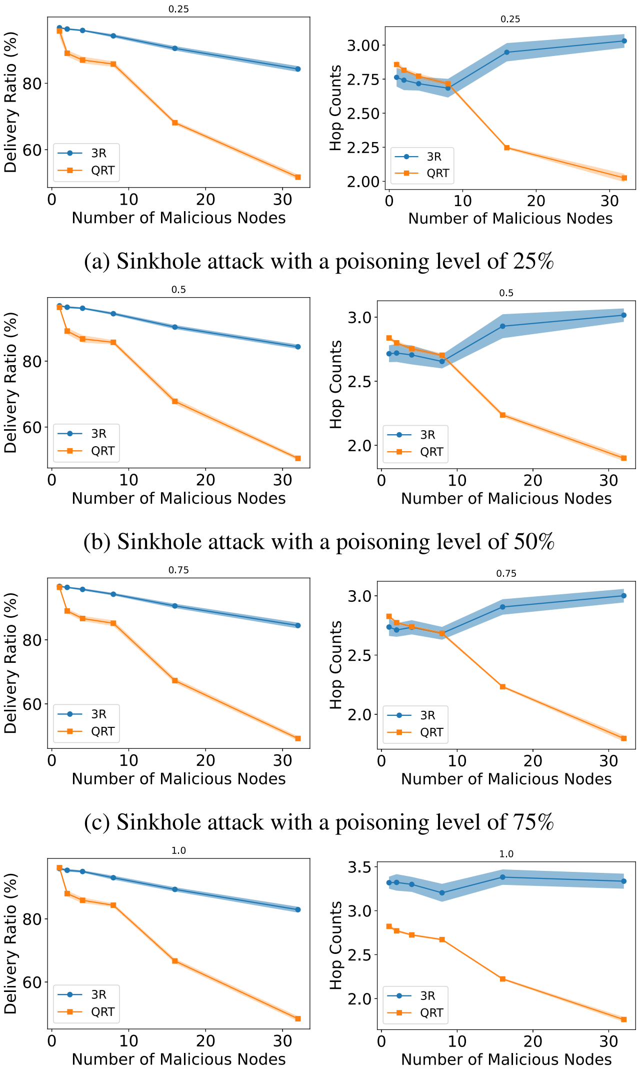
(b) Malicious nodes run volatile selective forwarding attacks by randomly changing x_t^i per 20% of the simulation time

Fig. 6. The average delivery ratio and hop counts under two selective forwarding attack scenarios.

resulted in a slight variability increase. This variability increase in the hop counts indicates the ability to find redundant, reliable paths to the destination, which could be seen in the stable delivery ratio. In contrast, QRT needs more hop counts for the limited number of malicious nodes. Furthermore, it fails to find reliable paths inferred from its low delivery ratio and hop counts.

4.5. Sinkhole attacks

The sinkhole attack is one of the most destructive attacks on routing protocols in which the malicious node attracts the network traffic by advertising false routing information [5]. This route poisoning attack is an easy to launch and extremely hazardous attack. In RL-based routing protocols, the learning agents exchange routing information to update the Q table and re-evaluate the optimal paths. When the adversary advertises false overestimated information to a specific destination, it can poison the Q tables of other nodes and attract all the traffic in order to drop it. In this experiment, the robustness of 3R is evaluated under different poisoning levels. Four scenarios have been considered in this experiment. The malicious nodes advertise the actual Q values increased by 25%, 50%, 75% and 100%. In the last scenario, when the Q values are increased by 100%, the malicious nodes will advertise the value zero to the network, which is the highest Q value that could be achieved as the reward function is designed to penalize dropping activities to ensure that the learning agents will always choose the most reliable shortest path. Figs. 7(a)–7(d) show the delivery ratio and the hop counts for the four scenarios. What stands out in these figures is the stable delivery ratio of 3R for different route poisoning levels, which reveals a high resiliency to sinkhole attacks. Moreover, they reveal how 3R finds the optimal paths through a hostile environment. 3R shows the same behavior as previous experiments when the number of malicious SNs increases. It avoids malicious nodes by choosing the most reliable path with minimal achievable hop counts. It is worth noting that when the malicious nodes advertise zeros as their best estimation, 3R shows a slight increase in hop counts even for a low number of malicious nodes, but with a high delivery ratio. The reason behind this behavior is that advertising this level of fake information affects the Q tables of the surrounding nodes, making the learning agent even tries to avoid the neighbors of malicious nodes.



(a) Sinkhole attack with a poisoning level of 25%

(b) Sinkhole attack with a poisoning level of 50%

(c) Sinkhole attack with a poisoning level of 75%

(d) Sinkhole attack with a poisoning level of 100%

Fig. 7. The average delivery ratio and hop counts under various poisoning levels of sinkhole attacks.

On the other hand, QRT shows a good delivery ratio when only one malicious node exists. However, by increasing the number of malicious nodes, the delivery ratio drops significantly to levels below 50% for 32 malicious nodes. This failure in avoiding malicious nodes can be clearly seen in the hop counts results, where QRT experiences a steep drop in contrast to what is expected, which explains the meager delivery ratio as packets would be ended up in a sinkhole.

4.6. On-off attacks

Although trust management schemes detect malicious activities, they are vulnerable to on-off attacks, where smart adversaries can change their behavior alternately to cheat the TMS and keep themselves undetected [50]. The failure to detect on-off attacks negatively impacts the performance of trust-based routing protocols by making them to make wrong routing decisions. The on-off attack cycle consists of one on and one off periods. During the on period, the adversary drops packets intentionally, while during the off period, it behaves well to rebuild its trust score and keep itself undetected. In this experiment, three simulation scenarios have been considered, variable traffic rates, variable on-off cycles, and non-identical periods.

4.6.1. Variable traffic rates

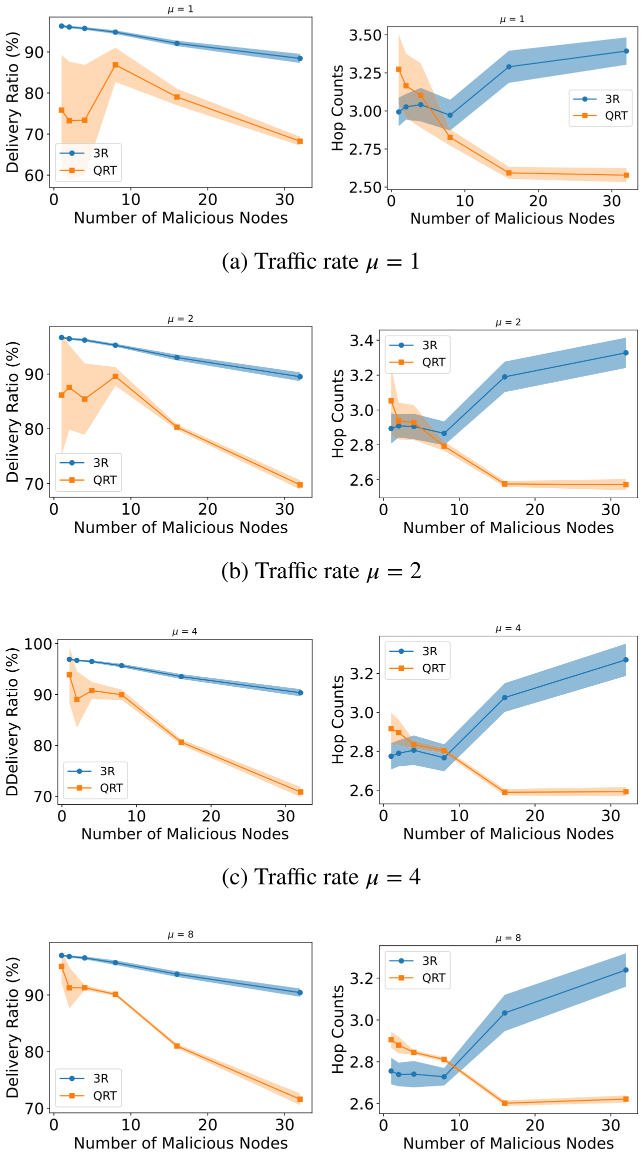
Experiments in our previous work [34] shows that some TM schemes failed to operate properly under low traffic rates. Thus, our first simulation scenario is designed to evaluate the routing performance under on-off attacks for different traffic rates starting from a low traffic rate $\mu = 1$ and doubling it each time. Figs. 8(a)–8(d) show the delivery ratio and the hop counts for the traffic rates $\mu = 1$, $\mu = 2$, $\mu = 4$ and $\mu = 8$, respectively. The on-off attack cycle is set to 40 s. The results show that QRT does not work properly for low traffic rates. Moreover, by increasing the number of malicious nodes, the delivery ratio decreased significantly, which could be attributed to the inability to avoid malicious nodes from the routing path. On the other hand, 3R shows superior performance for all traffic rates. It achieved a delivery ratio between around 90% to 97% for all malicious nodes ratios. The hop counts results show that 3R can find alternative paths to avoid malicious nodes, which obviously appears when having 25% – 50% of nodes behaving maliciously. It is worth noting that by increasing the traffic rate, 3R can find more optimal paths, which could be attributed to having more evidence from the environment.

4.6.2. Variable on-off cycles

This experiment evaluates the performance under different on-off attack cycles. The on-off attack's cycle varies from 10 s to 40 s. Figs. 9(a)–9(d) show the delivery ratio and hop counts results for various on-off attacks' cycles. 3R shows superior and stable performance for all on-off cycles. It achieved the same delivery ratio of the previous scenario, coupled with the same behavior of selecting paths to destinations, which again indicates its robustness against different on-off attacks. On the other hand, QRT shows lower delivery ratios with high variability for a low number of malicious nodes. By increasing the number of malicious nodes, the delivery ratio decreased significantly, which explains the rationale behind decreasing hop counts when having more malicious nodes.

4.6.3. Non-identical periods

Smart adversaries can execute more intricate on-off attacks by shortening the on period compared to the off period. This strategy aims to deceive the TMS and adds complexity to the detection of the attack. In this experiment, non-identical on-off attacks are launched by making the on period less than the off period. Four scenarios have been considered by varying the on period from 25% of the off period and up to 100%. The on-off cycle is set to 40 s and the traffic rate is set to 4 p/s. Figs. 10(a)–10(d) show the delivery ratio and hop counts for various on period's ratios. 3R shows a stable, superior performance, indicating its ability to detect attacks and isolate the malicious nodes. On the other hand, QRT shows low delivery ratios when increasing the number of malicious nodes with high variability for the low number of malicious nodes, which indicates difficulty in converging to the global optimum.



(a) Traffic rate $\mu = 1$

(b) Traffic rate $\mu = 2$

(c) Traffic rate $\mu = 4$

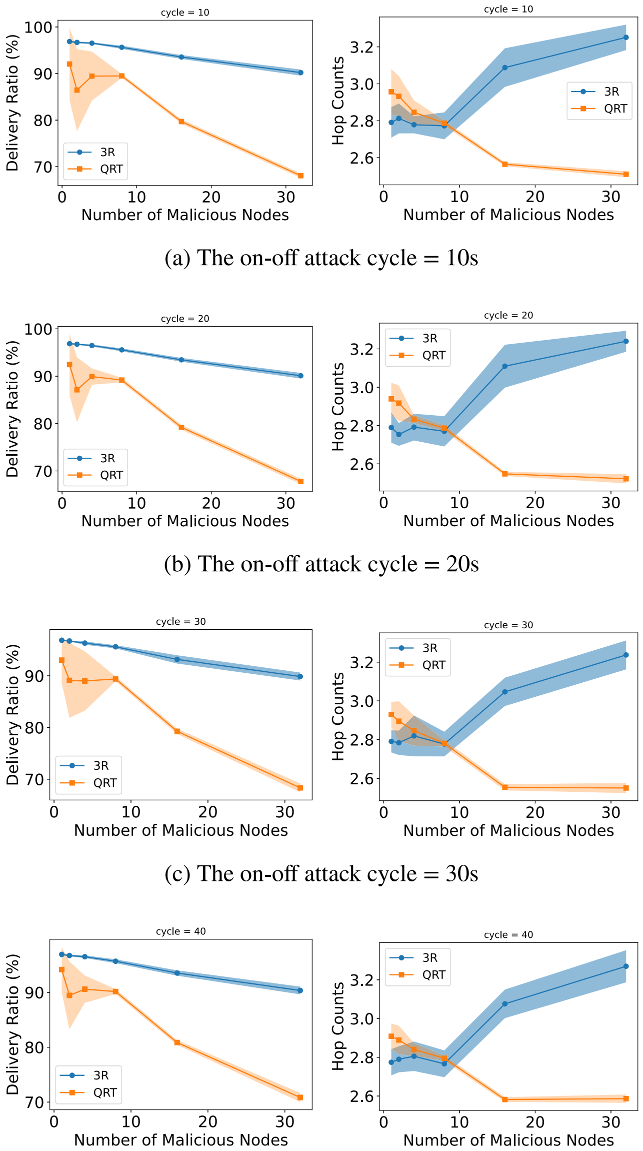
(d) Traffic rate $\mu = 8$

Fig. 8. The average delivery ratio and hop counts under On-Off attacks across varying traffic rate.

4.7. Network dynamicity and convergence

The convergence time is a crucial factor in routing applications as slow convergence results in more packets to lose, which could endanger the patient's life. Moreover, nodes' mobility could change the environment and require the algorithm to re-converge again. In this experiment, the convergence has been studied for the stationary and non-stationary environment under blackhole attacks where 50% of the nodes are malicious. First, stationary SNs have been considered to compare the convergence time of both protocols. Fig. 11 shows the convergence time of both protocols. 3R is able to converge with less than 20 s thanks to its asynchronous updating method in which the agent updates its routing decision engine once evidence obtained from the environment. This method makes 3R an adaptive protocol that can reflect any environment change. In contrast, QRT needs around double this time to converge. It is worth noting that QRT shows a bit better performance at the start because it uses positional information to make routing decisions, whereas 3R only depends on its trial/error process to learn the optimal routing decisions.

In the second scenario, mobility has been introduced to study how algorithms re-converge in a dynamic environment. For example, patients can change their locations within the hospital ward. Therefore, in this experiment, two different patients will change their locations at time 50 s and 100 s. The patient could have up to 3 SNs. Thus, three simulations have been run for 1, 2, and 3 randomly chosen SNs for each patient. The results show a fast re-convergence in all cases for 3R, as shown in Fig. 12(a). Once the environment change happens, 3R updates its routing engine asynchronously to reflect the new environment. This could be seen as a slight decrease in the delivery ratio at the time of movements, followed by fast re-convergence. On the other hand, QRT experienced a noticeable decrease with difficulty in re-converging, especially after the second movement, as shown in Fig. 12(b). The reason behind this poor performance could be attributed to considering positional information, which influences the routing decision.



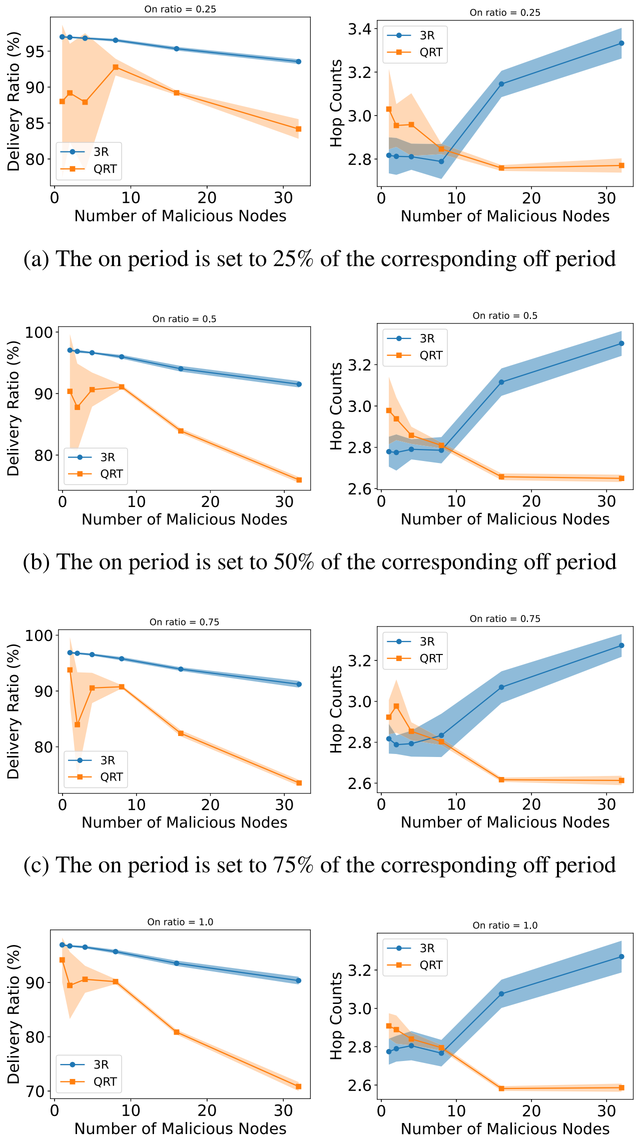
(a) The on-off attack cycle = 10s

(b) The on-off attack cycle = 20s

(c) The on-off attack cycle = 30s

(d) The on-off attack cycle = 40s

Fig. 9. The average delivery ratio and hop counts across various cycles of On-Off attacks.



(a) The on period is set to 25% of the corresponding off period

(b) The on period is set to 50% of the corresponding off period

(c) The on period is set to 75% of the corresponding off period

(d) The on and off periods are identical

Fig. 10. The average delivery ratio and hop counts under non-identical periods of on-off attacks.

4.8. *Energy efficiency*

The energy efficiency has been evaluated in two experiments, by modeling the energy consumption as explained in Section 3.8 in our simulations. In the first, the network lifetime has been compared between both protocols. The second scenario shows the average consumed energy by a node for different traffic rates. Network lifetime could be defined as the running time until a node dies [33]. Both simulation scenarios have been carried out under normal operation without introducing any attack. Fig. 13(a) shows the percentage of alive nodes during the simulation. QRT has a very short network lifetime compared to 3R. The first node dies after around 12 s on average. This deficiency could be attributed to two reasons. First, QRT does not take any energy-related factors into account to choose the optimal path, and most importantly, the excessive information exchanging increases the RF activities significantly, which is responsible for 80% of the consumed energy. On the other hand, 3R shows superior performance because of its resource-conservative design, which is clearly reflected in consuming less energy for all traffic rates, as obviously seen in Fig. 13(b).

4.9. *Computational overhead*

In this subsection, we compare the average processing time and memory consumption of both protocols, 3R and QRT. The experiment was carried out on an Intel Core i5-8500T processor at 2.1 GHz and 8 GB RAM. The simulation has been run for 30 times, and then the results have been averaged out and reported with one standard deviation. The network is in normal operation, and no attacks are launched during the simulation. The traffic rate is set to $\mu = 4$ p/s as QRT does not perform properly for lower traffic rates.

Fig. 14(a) shows the average processing time of 3R and QRT. The results show that QRT consumes more processing time than 3R. Moreover, the results show high variability of around 25%. This variability indicates that the algorithm sometimes takes longer to converge; hence, more packets will loop inside the network before reaching their destination. On the other hand, 3R consumes less processing time and saves

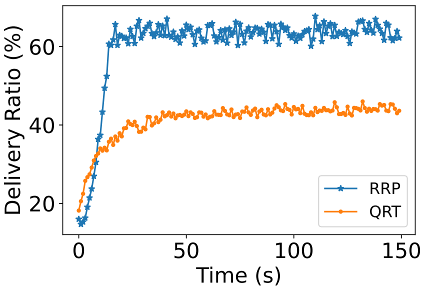
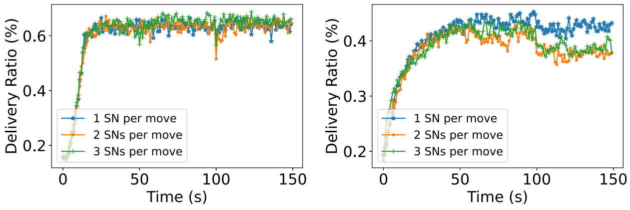


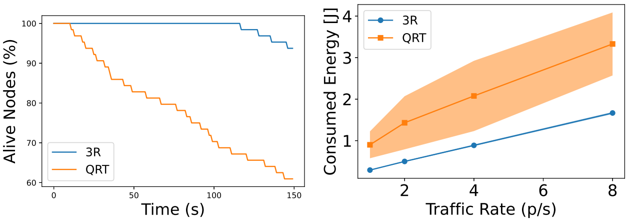
Fig. 11. The average convergence time for static SNs.



(a) 3R

(b) QRT

Fig. 12. The average delivery ratio under various mobility scenarios for 3R and QRT, respectively.



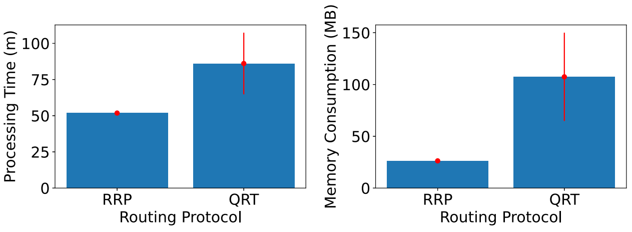
(a) The percentage of alive nodes during the simulation

(b) The consumed energy for various traffic rates

Fig. 13. The energy efficiency results.

around 40% of the processing time of QRT. Moreover, 3R shows almost no variability, indicating the stability of performance and the ability to converge at approximately the same time for different simulation runs. This lightweight processing overhead is attributed to the proposed resource-efficient RL model, where the learning agent receives one reward for multiple actions and hence updates the routing engine less than the traditional RL model.

The second important performance metric is memory consumption. Average memory consumption was calculated and reported with one standard deviation in Fig. 14(b). The memory allocation has been traced during the simulation using tracemalloc [51], a trace memory allocation module. The results show that QRT consumes a considerable amount of memory, around 107MB, with a high variability of around 39%. This high memory consumption is attributed to having more packets looping in the network, while the high variability indicate difficulty in converging to the global optimum. For each simulation run, QRT converges to a local optimum, which causes inconsistent memory consumption between different simulation runs. On the other hand, 3R is a memory-conservative protocol. It consumes a decent amount of memory, around 26MB, which saves around 75% of the memory consumed by QRT. Moreover, 3R shows almost no variability, indicating that 3R did not experience any converging difficulties thanks to its novel updating mechanisms.



(a) Average processing time

(b) Average consumed memory

Fig. 14. The average processing time and memory consumption.

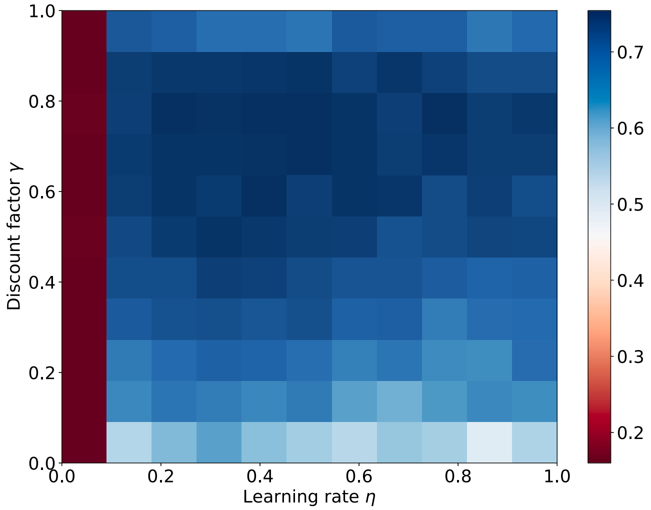


Fig. 15. Hyperparameters optimization heatmap.

4.10. Hyperparameters tuning

Optimizing the used hyperparameters lead to faster convergence and overall better performance. In previous experiments, the benchmark parameters setting has been adopted to ensure fair compression between both protocols. In this experiment, the learning rate η and the discount factor γ are tuned using the grid search approach. As both parameters are continuous in the domain $[0, 1]$, a step of 0.1 has been used for each parameter. This involves a combination of 121 simulation parameter settings. Each one of them has been run for 30 times. The simulation has been run for blackhole attacks where 50% of the nodes are malicious and the traffic rate is set to 1 p/s. Fig. 15 shows the heatmap of the average delivery ratio of these simulations. Closer inspection of the figure shows poor delivery ratio for $\eta = 0$ and $\gamma = 0$. When $\eta = 0$, the algorithm only uses its current observation and does not learn from previous experience, which causes poor performance. The values between $[0.2, 0.4]$ show the highest performance, although the heatmap also shows good performance for $\eta = 0.8$; however, consulting the hop counts results shows that the algorithms take slightly more hops to reach the destination, indicating that the algorithm did not converge to the global optimum.

On the other hand, the discount factor plays a significant role also. What stands out in the figure is myopic learning agent performs poorly. For instance, when the discount factor $\gamma = 0$, the learning agent only considers its direct observations from the observable environment to choose the optimal path, which likely leads to losing the packet due to existing a malicious node in the in-observable path. Interestingly, the maximum value of γ also shows poor performance as the learning agent weighs current and expected future rewards equally, which influences the routing decision negatively based on in-observable future rewards. Thus, the optimal value for the discount factor is around 0.8, which could efficiently balance the current and future expected rewards.

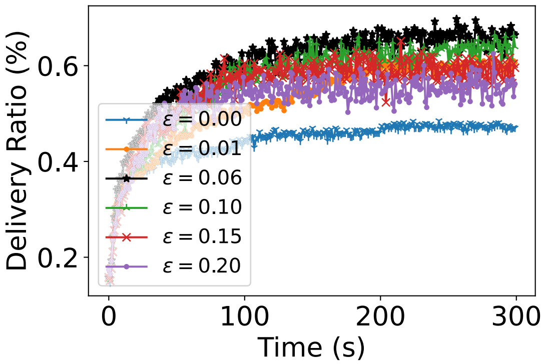


Fig. 16. Optimizing ε-greedy exploration algorithm.

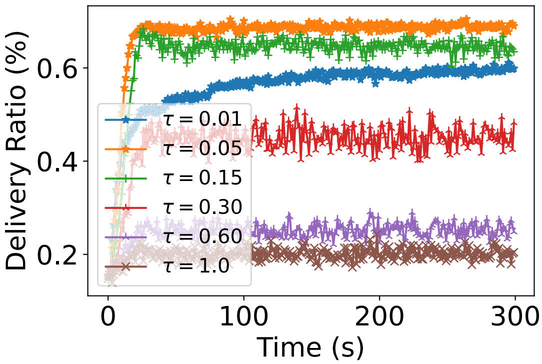


Fig. 17. Optimizing softmax exploration algorithm.

4.11. Exploration exploitation optimization

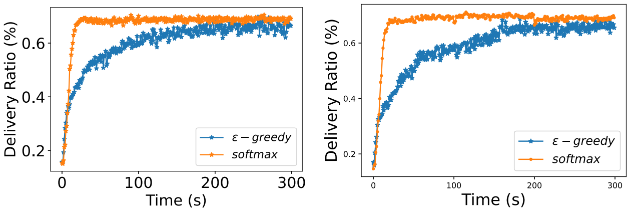
Exploration exploitation trade-off is a critical component in RL model. The learning agent needs to explore the stochastic environment in order to maximize the long-term reward. During the exploration phase, the learning agent tries to discover the most rewardable actions. However, taking an action at a state s may affect the immediate reward as well as the subsequent rewards, while insufficient exploration may lead to converging to a sub-optimal solution. ε-greedy and Softmax exploration methods are the most used algorithms in the literature to balance exploration-exploitation. ε-greedy has been used in previous experiments. However, the reported values in our benchmark have been adopted to ensure a fair comparison. In this experiment, we optimize the value ε under blackhole attacks where 50% of the nodes are malicious. ε is a continuous value in the range [0, 1]. Therefore, a step of 0.01 has been chosen. The simulation has been run for values in the range $[0, 0.2]$ as higher values over-explore the environment and shows poor performance. Fig. 16 shows the results of only some ε values for clarity. The value $\varepsilon = 0.06$ achieves the highest reward and is able to converge faster than other values. This means that the learning agent randomly explores the environment with a probability of 6%.

Softmax exploration algorithm is a value-based approach to exploring the environment in which the learning agents make informed routing decisions based on its Q table. Softmax algorithm is modeled using Gibbs distribution as shown in Eq. (9)

$$\pi(a|s) = Pr\{a_t = a|s_t = s\} = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_1^n e^{\frac{Q(s,a)}{\tau}}} \tag{9}$$

where τ is called the temperature parameter, which is used to control the probability of choosing the greedy action. Decreasing the value of τ, increases the probability of choosing the greedy action. Moreover, ε-greedy could be derived from Softmax algorithm when $\tau \to \infty$ as all possible actions will have the same probability. Optimizing the temperature value of the Gibbs distribution is not straightforward [52] as any change in reward function can influence the probabilities of available actions. In this experiment, the temperature parameter will be optimized under the same conditions. As $\tau \in \mathbb{R}^+$, we start at $\tau = 0.01$ and then increase the value up to $\tau = 1$. Fig. 17 shows the results of some temperature values for clarity. The performance starts poorly at $\tau = 0.01$ and enhanced gradually to reach the peak at $\tau = 0.05$, and further increase beyond 0.05 decreases the delivery ratio.

In the third experiment, we compare both algorithms using optimized values for stationary and non-stationary environments. Fig. 18(a) shows the results for a stationary environment. Softmax shows superior performance. The algorithm converges fast to the global optimum, while ε-greedy needs more time to converge. This could be attributed to the mechanism of making routing decisions. ε-greedy exploration could be regarded as blind exploration as actions are chosen randomly during the exploration, while the Softmax algorithm takes informative actions based on the current estimations. On the other hand, the results of the non-stationary environment in Fig. 18(b) show that both algorithms are able to re-converge fast after movement detection.



(a) Stationary Network (b) Non-stationary Network

Fig. 18. Comparing ε-greedy and softmax exploration algorithms for stationary and non-stationary environment.

5. Conclusion

There is still a persistent need for a lightweight and secure routing protocol for WMSN. Although RL is regarded as a promising approach to building a routing protocol for WMSN, the widely used RL model is a resource-consuming model. Moreover, reliable data delivery cannot be achieved using only routing metrics as this information cannot deal with the free will of other relay nodes inside the network. Realizing the aforementioned problems open the way to re-design a lightweight RL model and integrate a security tool with the routing decision engine to ensure lightweight and reliable data transfer for WMSN. The proposed RL model does not necessitate updating the Q table after each sent/forwarded packet, but rather it updates it periodically after receiving a reward for a set of actions within one time unit. The performance results show that the proposed RL model can significantly reduce the computational overhead. Furthermore, integrating TMS with the routing engine enables reliable data delivery and avoids malicious paths that cannot be achieved using traditional routing metrics. Finally, the experimental results prove the robustness of our proposed method in defeating well-known dropping attacks. In the future, hyperparameters optimization will be further investigated to determine if there is a relation between the hyperparameters and other environment parameters. Moreover, exploration-exploitation trade-off dilemma will be further studied to design a well-tailored exploration strategy for routing applications as poor exploration strategy may affect the overall network performance negatively.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] X. Li, B. Tao, H.-N. Dai, M. Imran, D. Wan, D. Li, Is blockchain for internet of medical things a panacea for COVID-19 pandemic? Pervasive Mob. Comput. 75 (2021) 101434.

[2] M.S. Hajar, M.O. Al-Kadri, H.K. Kalutarage, A survey on wireless body area networks: Architecture, security challenges and research opportunities, Comput. Secur. 104 (2021) 102211.

[3] H. Fu, Y. Liu, Z. Dong, Y. Wu, A data clustering algorithm for detecting selective forwarding attack in cluster-based wireless sensor networks, Sensors 20 (1) (2020) 23.

[4] N. Khanna, M. Sachdeva, A comprehensive taxonomy of schemes to detect and mitigate blackhole attack and its variants in MANETs, Comp. Sci. Rev. 32 (2019) 24–44.

[5] K. Prathapchandran, T. Janani, A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST, Comput. Netw. 198 (2021) 108413.

[6] S.M. Altowaijri, Efficient next-hop selection in multi-hop routing for IoT enabled wireless sensor networks, Future Internet 14 (2) (2022) 35.

[7] W.-K. Yun, S.-J. Yoo, Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks, IEEE Access 9 (2021) 10737–10750.

[8] R. Maivizhi, P. Yogesh, Q-learning based routing for in-network aggregation in wireless sensor networks, Wirel. Netw. 27 (3) (2021) 2231–2250.

[9] G. Künzel, L.S. Indrusiak, C.E. Pereira, Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing, IEEE Trans. Ind. Inform. 16 (8) (2020) 5617–5625, http://dx.doi.org/10.1109/TII.2019.2941771.

[10] Z. Ullah, I. Ahmed, F.A. Khan, M. Asif, M. Nawaz, T. Ali, M. Khalid, F. Niaz, Energy-efficient harvested-aware clustering and cooperative routing protocol for WBAN (E-HARP), IEEE Access 7 (2019) 100036–100050.

[11] Y. Qu, G. Zheng, H. Ma, X. Wang, B. Ji, H. Wu, A survey of routing protocols in WBAN for healthcare applications, Sensors 19 (7) (2019) 1638.

[12] R. Cavallari, F. Martelli, R. Rosini, C. Buratti, R. Verdone, A survey on wireless body area networks: Technologies and design challenges, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1635–1657.

[13] K. Karmakar, S. Biswas, S. Neogy, MHRP: A novel mobility handling routing protocol in Wireless Body Area network, in: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), IEEE, 2017, pp. 1939–1945.

[14] M. Quwaider, S. Biswas, DTN routing in body sensor networks with dynamic postural partitioning, Ad Hoc Netw. 8 (8) (2010) 824–841.

[15] A. Samanta, S. Misra, Energy-efficient and distributed network management cost minimization in opportunistic Wireless Body Area networks, IEEE Trans. Mob. Comput. 17 (2) (2018) 376–389, http://dx.doi.org/10.1109/TMC.2017.2708713.

[16] Q. Tang, N. Tummala, S.K. Gupta, L. Schwiebert, TARA: thermal-aware routing algorithm for implanted sensor networks, in: International Conference on Distributed Computing in Sensor Systems, Springer, 2005, pp. 206–217.

[17] A. Ahmad, N. Javaid, U. Qasim, M. Ishfaq, Z.A. Khan, T.A. Alghamdi, RE-ATTEMPT: a new energy-efficient routing protocol for wireless body area sensor networks, Int. J. Distrib. Sens. Netw. 10 (4) (2014) 464010.

[18] N. Javaid, Z. Abbas, M. Fareed, Z.A. Khan, N. Alrajeh, M-ATTEMPT: A new energy-efficient routing protocol for wireless body area sensor networks, Procedia Comput. Sci. 19 (2013) 224–231.

[19] O. Rafatkhah, M.Z. Lighvan, M2e2: A novel multi-hop routing protocol for wireless body sensor networks, Int. J. Comput. Netw. Commun. Secur. 2 (8) (2014) 260–267.

[20] M.M. Monowar, M. Mehedi Hassan, F. Bajaber, M.A. Hamid, A. Alamri, Thermal-aware multiconstrained intrabody QoS routing for wireless body area networks, Int. J. Distrib. Sens. Netw. 10 (3) (2014) 676312.

[21] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.

[22] M. Al-Shalabi, M. Anbar, T.-C. Wan, A. Khasawneh, Variants of the low-energy adaptive clustering hierarchy protocol: Survey, issues and challenges, Electronics 7 (8) (2018) 136.

[23] B. Abidi, A. Jilbab, E.H. Mohamed, An energy efficiency routing protocol for wireless body area networks, J. Med. Eng. Technol. 42 (4) (2018) 290–297.

[24] IEEE, IEEE standard for local and metropolitan area networks - part 15.6: Wireless body area networks, IEEE Std 802.15.6-2012 (2012) 1–271, http://dx.doi.org/10.1109/IEEESTD.2012.6161600.

[25] Y. Peng, S. Zhang, A power optimization routing algorithm for Wireless Body Area network, Electron. Sci. Technol. 31 (2018) 38–41.

[26] W.-K. Yun, S.-J. Yoo, Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks, IEEE Access 9 (2021) 10737–10750.

[27] G. Künzel, L.S. Indrusiak, C.E. Pereira, Latency and lifetime enhancements in industrial wireless sensor networks: A Q-learning approach for graph routing, IEEE Trans. Ind. Inform. 16 (8) (2019) 5617–5625.

[28] R. Maivizhi, P. Yogesh, Q-learning based routing for in-network aggregation in wireless sensor networks, Wirel. Netw. 27 (3) (2021) 2231–2250.

[29] G. Liu, X. Wang, X. Li, J. Hao, Z. Feng, ESRQ: An efficient secure routing method in wireless sensor networks based on Q-learning, in: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2018, pp. 149–155.

[30] K. Upreti, N. Kumar, M.S. Alam, A. Verma, M. Nandan, A.K. Gupta, Machine learning-based congestion control routing strategy for healthcare IoT enabled wireless sensor networks, in: 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), IEEE, 2021, pp. 1–6.

[31] X. Liang, I. Balasingham, S.-S. Byun, A reinforcement learning based routing protocol with QoS support for biomedical sensor networks, in: 2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies, IEEE, 2008, pp. 1–5.

[32] Y. Naputta, W. Usaha, RL-based routing in biomedical mobile wireless sensor networks using trust and reputation, in: 2012 International Symposium on Wireless Communication Systems (ISWCS), IEEE, 2012, pp. 521–525.

[33] W. Guo, C. Yan, T. Lu, Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing, Int. J. Distrib. Sens. Netw. 15 (2) (2019).

[34] M.S. Hajar, M.O. Al-Kadri, H. Kalutarage, Ltms: A lightweight trust management system for wireless medical sensor networks, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 1783–1790.

[35] F. Morady, Electrophysiologic interventional procedures and surgery, in: Goldman's Cecil Medicine, Elsevier, 2012, pp. 369–373.

[36] N. Azdad, M. Elboukhari, Wireless Body Area networks for healthcare: Application trends and mac technologies, Int. J. Bus. Data Commun. Netw. (IJBDCN) 17 (2) (2021) 1–20.

[37] F. Yuan, J. Wu, H. Zhou, L. Liu, A double Q-learning routing in delay tolerant networks, in: ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019, pp. 1–6.

[38] T. Li, K. Zhu, N.C. Luong, D. Niyato, Q. Wu, Y. Zhang, B. Chen, Applications of multi-agent reinforcement learning in future internet: A comprehensive survey, IEEE Commun. Surv. Tutor. (2022).

[39] K. Zhang, Z. Yang, H. Liu, T. Zhang, T. Basar, Fully decentralized multi-agent reinforcement learning with networked agents, in: International Conference on Machine Learning, PMLR, 2018, pp. 5872–5881.

[40] A. Kumar, R. Matam, S. Shukla, Impact of packet dropping attacks on RPL, in: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2016, pp. 694–698, http://dx.doi.org/10.1109/PDGC.2016.7913211.

[41] M. Tokic, G. Palm, Value-difference based exploration: adaptive control between epsilon-greedy and softmax, in: Annual Conference on Artificial Intelligence, Springer, 2011, pp. 335–346.

[42] J. Jiang, X. Zhu, G. Han, M. Guizani, L. Shu, A dynamic trust evaluation and update mechanism based on C4. 5 decision tree in underwater wireless sensor networks, IEEE Trans. Veh. Technol. 69 (8) (2020) 9031–9040.

[43] V. Krishnaswamy, S.S. Manvi, Trusted node selection in clusters for underwater wireless acoustic sensor networks using fuzzy logic, Phys. Commun. 47 (2021) 101388.

[44] N. Matloff, Introduction to discrete-event simulation and the simpy language, vol. 2, Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August, 2008, pp. 1–33.

[45] H.-J. Chang, C.-H. Wu, J.-F. Ho, P.-y. Chen, On sample size in using central limit theorem for gamma distribution, Inf. Manage. Sci. 19 (1) (2008) 153–174.

[46] M.N. Islam, M.R. Yuce, Review of medical implant communication system (MICS) band and network, Ict Express 2 (4) (2016) 188–194.

[47] M.S. Hajar, H. Kalutarage, M.O. Al-Kadri, DQR: A double q learning multi agent routing protocol for wireless medical sensor network, in: 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm), Springer, 2022.

[48] S.S. Javadi, M. Razzaque, Security and privacy in wireless body area networks for health care applications, in: Wireless Networks and Security, Springer, 2013, pp. 165–187.

[49] M.S. Hajar, H. Kalutarage, M.O. Al-Kadri, RRP: A reliable reinforcement learning based routing protocol for wireless medical sensor networks, in: 2023 IEEE 20th Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2023.

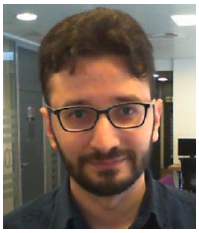
[50] R.R. Sahoo, S. Sarkar, S. Ray, Defense against on-off attack in trust establishment scheme for wireless sensor network, in: 2019 2nd International Conference on Signal Processing and Communication (ICSPC), IEEE, 2019, pp. 153–160.

[51] TRACEMALLOC, TRACEMALLOC - Trace memory allocations - Python 3.10.2 documentation, URL https://docs.python.org/3/library/tracemalloc.html. Accessed: 2022-02-08.

[52] S. Amin, M. Gomrokchi, H. Satija, H. van Hoof, D. Precup, A survey of exploration methods in reinforcement learning, 2021, arXiv preprint arXiv:2109.00157.

Muhammad Shadi Hajar is a lecturer in Networking and Cyber Security at Robert Gordon University in the UK. He received his B.Eng. in Computer Engineering and Automation in 2008 and M.Sc. in Computer Engineering and Networking in 2013 from Damascus University, Damascus, Syria. He holds a Ph.D. in Cyber Security from Robert Gordon University in the UK. His current research interests are AI for security applications, reliable routing, Wireless Medical Sensor Networks, trust management, and lightweight authentication.

Harsha Kumara Kalutarage is a senior lecturer in Cyber Security at Robert Gordon University in the UK. His research combines AI & Security with a particular focus on using AI techniques for security applications (e.g. IoT) and building security for AI-embedded systems via analyzing the security vulnerabilities of AI algorithms. He has 10+ years of research experience in this area and has produced 60+ publications, patents and technology transfers to industry.



M. Omar Al-Kadri is a senior lecturer in networking and cyber security at Birmingham City University. His current research interests include security of wireless communications with application to healthcare, security of vehicular networks, full-duplex communications, and 5G security. He holds a PhD in telecommunication engineering from King's College London, and MSc in Networking and data communications from Kingston University.