RRP: A Reliable Reinforcement Learning Based Routing Protocol for Wireless Medical Sensor Networks

Muhammad Shadi Hajar
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
m.hajar@rgu.ac.uk

Harsha Kalutarage
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
h.kalutarage@rgu.ac.uk

M. Omar Al-Kadri
School of Computing & Digital Tech.
Birmingham City University
Birmingham, United Kingdom
omar.alkadri@bcu.ac.uk

***Abstract*—Wireless medical sensor networks (WMSNs) offer innovative healthcare applications that improve patients' quality of life, provide timely monitoring tools for physicians, and support national healthcare systems. However, despite these benefits, widespread adoption of WMSN advancements is still hampered by security concerns and limitations of routing protocols. Routing in WMSNs is a challenging task due to the fact that some WMSN requirements are overlooked by existing routing proposals. To overcome these challenges, this paper proposes a reliable multi-agent reinforcement learning based routing protocol (RRP). RRP is a lightweight attacks-resistant routing protocol designed to meet the unique requirements of WMSN. It uses a novel Q-learning model to reduce resource consumption combined with an effective trust management system to defend against various packet-dropping attacks. Experimental results prove the lightweightness of RRP and its robustness against blackhole, selective forwarding, sinkhole and complicated on-off attacks.**

***Index Terms*—Routing, Reinforcement Learning, Trust Management, Blackhole, Selective Forwarding, Sinkhole, On-off.**

I. 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 [1,2]. Any security breach may disrupt the network operation and threaten the patient's life.

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, a SN could get compromised and start dropping packets with a view to disrupting the overall network operations. Another example is when a SN acts selfishly and stops relaying packets for others to save power or gain extra resources unfairly. 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].

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 [5], 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 [6–8]. 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. 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 RRP, a novel RL model is used to produce a lightweight, efficient routing protocol. Moreover, an effective Trust Management (TM) scheme is integrated with the RRP to ensure high delivery reliability. The reward function has been redefined as a punishment function based on the trustworthiness of potential routes.

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 six sections as follows. Related work is given in Section II. Section III overviews WMSN. The proposed routing protocol for WMSN is presented in Section IV, followed by evaluation and performance results in Section V. Finally, Section VI concludes the paper and highlights future work.

II. 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 [9]. 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 its unique characteristics and requirements, which makes inherited routing protocols from WSN do not necessarily fit WMSN. Therefore, there is still an imperative research gap to design a routing protocol that fits WMSN and meets its requirements.

Reinforcement learning has been widely used in the literature to find the optimal path with minimum overhead. Researchers use different metrics to achieve this goal, such as delivery latency, residual energy and geographical distance [10]. However, this kind of metrics can not deal with the free will of 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. Trust Management System (TMS) provides an effective and robust measure to evaluate the trustworthiness of other nodes. To the best of our knowledge, only two schemes [11,12] are proposed in the literature that combine a TM scheme with a Q-learning routing model. Authors in [11] provide 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 [12] proposed QRT, a routing protocol designed for non-cooperative biomedical mobile wireless sensor networks. It has been proposed as an extension to RL-QRP [13] 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.

III. Wireless Medical Sensor Network

A. Overview

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 [14]. Therefore, lightweight countermeasures and protocols are essential to extend the battery life and avoid unnecessary surgery 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.

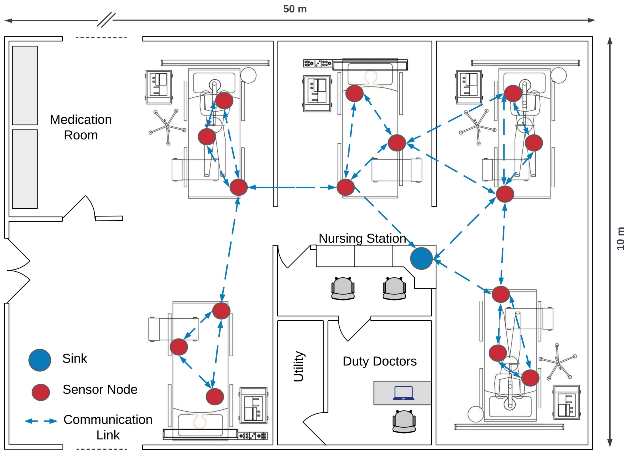


Fig. 1: Network Model

B. Network Model

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 a ward that is $50m \times 10m$ 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 [15]. 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 $5m$; 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 forward them to the medical server.

C. 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 [16]. 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.

Packet-dropping attacks are regarded as one of the most devastating internal attacks because of their consequences on the patients' lives. 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 behaviour 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. RRP protocol is evaluated for various kinds of dropping attacks with different parameter settings, such as blackhole, selective forwarding, and on-off attacks. Moreover, it will be evaluated for poisoning attacks, such as sinkhole attacks.

IV. RRP Protocol Design

In this section, the design requirements are specified and the proposed RRP is presented.

A. The 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 [17]. This interaction could be a collaboration to accomplish a common task, a competition to accomplish a self-goal, or a mix of both. At each time step t, the 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} 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 the 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 [7]. Each agent maintains a Q-values table of $|\mathbb{S}| \times |\mathbb{A}|$ represents the expected long-term rewards when the agent takes the action a_t at the state s_t.

B. The Proposed Synchronous Q-Routing Model

1) Design Requirements: Various objectives have been considered when designing RRP. 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 the receiver determines how efficient is the routing protocol, which is a crucial requirement for resource-constrained devices. The lowest cost path must always be chosen to ensure high efficient routing protocol. RF activities, especially transmission (TX), constitutes around 80% of the consumed energy [18]. In order to reduce the consumed energy, the SNs must always choose the shortest path in order to reduce the number of transmissions. Therefore, RRP 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 the 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 [8,12,13,19]. This method is a resource-consuming process, particularly when more packets are generated or forwarded. Therefore, a novel RL model is proposed to reduce the computational overhead.

Scalability is another requirement. In a multi-agent environment, each agent has to consider the actions of the others, which causes a scalability problem when the number of agents increases, as the action space grows exponentially [17]. Moreover, the agents in a networked environment suffer from a partial observability problem as they do not have a full view of the network. Therefore, decentralized learning with a networked agent approach [20] was adopted in RRP to enable the learning agents to collaborate with their neighbours by sharing information. This approach is a solution to the poor scalability of fully centralized learning and centralized training with decentralized execution approaches [17]. In addition, RRP has been evaluated for variable traffic rates and the maximum number of SNs as defined in IEEE 802.15.6 [15].

Resiliency to attacks 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. Therefore, RRP has been designed to resist all kinds of known dropping attacks. Moreover, it is also resilient to route poisoning attacks.

2) RRP Q-Routing Protocol: With the above design requirements in mind, RRP is built using the Q-learning algorithm in RL, incorporating an effective trust management algorithm to ensure an efficient, lightweight, and reliable routing protocol. The learning agent is modelled 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 a SN. The action $a \in \mathbb{A}$ is defined as selecting the next forwarder to relay packets to a destination. The agent receives a reward $r_{t+1} \in \mathbb{R}$ for each action a_t.

RRP 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 rewards. Each learning agents 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.

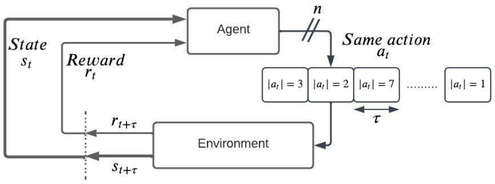


Fig. 2: RL Model

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

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

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

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. The trust value T^{ij}_t is computed using algorithm 4 as detailed in IV-C.

The traditional RL model is re-designed to produce a lightweight RL model that fits WMSN. Therefore, the actions and rewards of the RL model are re-defined. To the best of our knowledge, RRP is the first Q-learning model that uses the time window technique to reduce resource consumption. In RRP, 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. 2. Unlike other models [11,12,19] where the learning agent needs to observe the reward and update its Q-table for each packet, RRP evaluates the reward and updates the Q-table after a defined time unit τ. This proposed method is referred to as synchronous updating. Moreover, asynchronous updating is also used in RRP to help the algorithm to converge swiftly, which will be elaborated further in IV-B3.

Algorithm 1: RRP Protocol

```
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:
$Q^i_0(n^i \in N^i_t) = \begin{cases} 0 & if \quad n^i \neq S \\ 1 & if \quad n^i = S \end{cases}$
$T^i_0(n^i \in N^i_t) = 0.5$
$a^i_1 = \begin{cases} S & if \quad S \in N^i \\ n^i & | \quad n^i \in N^i \end{cases}$    (3)

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 $\varepsilon - greedy > \theta$ 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
```

The routing task must be achieved in a distributed manner as no agent has a full view of the network states. Therefore, RRP uses the 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. Once the next action is taken, it changes the environment, making periodic updates required. 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 prevents the algorithm from converging to the global optimum. Therefore, $\varepsilon-$greedy strategy [21] 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 is selected to search for possible alternative paths. At the beginning, RRP 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 [12,13], where the authors used positioning information.

3) Q-Table Updating Methods: In RRP routing protocol, two kinds of Q table updating methods are used to reduce resource consumption, as shown in algorithm 2. The synchronous updating is used to update the Q table at the end of each time unit with a view to reducing the processing overhead. As the action in the proposed RL model consists of multiple sub-actions on a predefined time unit, the learning agent performs the same sub-action multiple times during the period τ, which means all packets will be forwarded to the same next hop. Meanwhile, the agent is observing the behaviour 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. In this case, RRP updates the Q value of node j using Eq. 1 and checks how certain it is about node k by checking the number of recent observations. If

Algorithm 2: Synchronous and Asynchronous Q Table Updating

```
Input:
The Q table: $Q_t^i$
The reward: $r_{t+1}^i(s_{t+1}^i, j)$
The trust table: $T_t$
Output: Updated Q Table: $Q_{t+1}^i$
if Synchronous Update then
    foreach $j \in N_t^i$ do
        if $j == a_t^i$ then
            update $Q_t^{ij}$ using $r_{t+1}^i(s_{t+1}^i, j)$
        else
            if $|O^{ij}| > \epsilon$ then
                update $Q_t^{ij}$ using recent $r_{t-\delta}^i(s_{t-\delta}^i, j)$
            else
                $Q_{t+1}^{ij} \leftarrow Q_t^{ij}$
            end
        end
    end
end
if Asynchronous Update then
    if $\eta == 1$ then
        $r_{t+1}^i(s_{t+1}^i, j) = -e^{-\mu}(1 - T_t^{ij})$
    else
        $r_{t+1}^i(s_{t+1}^i, j) = -(1 - T_t^{ij})$
    end
    if $RQ_{t-1}^i(s_{t-1}^i, j)$ then          // $RQ_{t-1}^i(s_{t-1}^i, j)$ is last
     expected future reward received from $j$
        update $Q_t^{ij}$ using $r_{t+1}^i$ and $RQ_{t-1}^i(s_{t-1}^i, j)$
    else          // $\zeta$ is the loop penalising parameter
        $Q_{t+1}^i(s_t^i, a_t^i = n_j) \leftarrow Q_t^{ij} - \zeta$
    end
    $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
end
```

Algorithm 3: Loop Processing

```
Input: A packet to forward: $P_t^{sd}$
Output: Updated Routing
while TRUE do
    if $\forall\ i \in \mathbb{N}$ receives $P_{t+\delta}^{id}$ then     // $P_{t+\delta}^{id}$ is a packet from $i$
     to $d$ after time lag $\delta$
        Asynchronous Q table update as in algorithm 2
        $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
        Update $P_t^{id}$
        Send $P_t^{id}$
    end
    if $\forall\ i \in \mathbb{N}$ receives $P_t^{jd} \wedge a_t^i = j$ then
        Asynchronous Q table update as in algorithm 2
        $a_t^i \leftarrow \underset{n_t^i \in N_t^i}{argmax}\ Q_t^i(s_t^i, a_t^i)$
        Forward $P_t^{jd}$
    end
end
```

node i has adequate observations about node k, it will use the most recent reward $r_{t-\delta}^i(s_{t-\delta}^i, k)$ to update the Q_t^{ik}. Otherwise, it will ignore the received estimation and keep the Q value unchanged. This technique immunizes RRP 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.

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 models where the learning agent risks losing one packet for each exploring step, the synchronous updating model could lose more packets because it keeps forwarding to one next-hop during one time unit. Therefore, RRP introduces a loop detection and avoiding algorithm as shown in algorithm 3. Once a loop is detected, or there is a possibility for a loop to occur, the asynchronous update is called as shown in algorithm 2. The updating process penalizes the corresponding Q value which allows to choose another promising next hop. This technique enables RRP to perform efficiently and converge swiftly.

C. Trust Evaluation

RRP incorporates a trust management scheme as a security measure to ensure reliable data transfer. Several TM schemes have been evaluated to choose the best candidate. LTMS [14] 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. The trust scheme comprises two parts. The first is a novel updating algorithm to promptly detect any changes in forwarding behaviour. It integrates the slopes b_t and d_t with beta distribution levels. This technique allows α_t to decrease and may accumulate a negative value during the attack. At the same time, β_t develops a positive value, giving more weight to any misbehaviour and making it harder to forget. The second part is an on-off protection module designed to detect on-off attacks. Trust management schemes are vulnerable to on-off attacks where smart adversaries change their behaviour between good and bad with a view to keeping themselves undetected. The on-off module in LTMS is designed to detect repeated attack patterns. It incorporates the short 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.

V. Evaluation and Performance Results

This section simulates and analyzes the RRP routing protocol. Various simulation scenarios have been considered under different dropping attacks.

A. Experimental Setup

A WMSN of 64 SNs has been adopted to comply with IEEE 802.15.6 [15]. The SNs have been distributed randomly in an area of $50m \times 10m$ 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. RRP has been benchmarked with QRT [12], which is an extension to RL-QRP routing protocol [13] where the authors integrated reputation and trust scheme to deal with non-cooperative and misbehaving nodes in biomedical sensor networks. In order

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
        $Rep_t^{ij} = 0$;
    else
        $Rep_t^{ij} = \frac{\alpha_t}{\alpha_t + \beta_t}$;
    end
    if $T_{t-1}^{ij} \geq thr_1$ && $Rep_t^{ij} < thr_1$ then
        if $malicious > 0$ then
            $cycle = t - malicious$;
            $malicious = 0$;
        else
            $malicious = t$;
        end
    end
    if $cycle > 0$ && $Trust(t-1) < thr_2$ then
        $ShRep_t^{ij} = mean(T_{t-cycle:t}^{ij})$;
        $T_t^{ij} = min(ShRep_t^{ij}, Rep_t^{ij})$;
    else
        $T_t^{ij} = Rep_t^{ij}$;
        $cycle = 0$;
    end
end
```

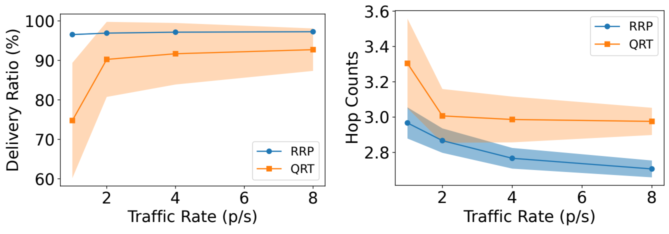
TABLE I: Simulation Parameters

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

to ensure a fair comparison between the two protocols, the reported parameters setting of QRT have been adopted. Table I shows the setting of simulation parameters. 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 [22]. The simulation time is $500s$ where the first $50s$ is regarded as a training period. During the simulation, the agents adopt the $\varepsilon-$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. The results are averaged out and reported with one standard deviation.

B. Normal Operation

In this experiment, the performance of RRP 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. This experiment aims to ensure that RRP 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 [23]. Therefore, the experiment has been run for four different traffic rates starting at $\mu = 1p/s$ and doubling it each time. Fig. 3a and Fig. 3b show the average delivery ratio and the average hop counts with one standard deviation, respectively. Results show that RRP 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 $\mu = 2p/s$ to achieve around 90%; however, the high variability of the delivery ratio confirms that QRT struggles to converge. On the other hand, Fig. 3b reveals that RRP always chooses the shortest path to the destination. Moreover, the performance is slightly enhanced by generating more traffic because the learning agents get more evidence from the environment to enhance their routing decisions.

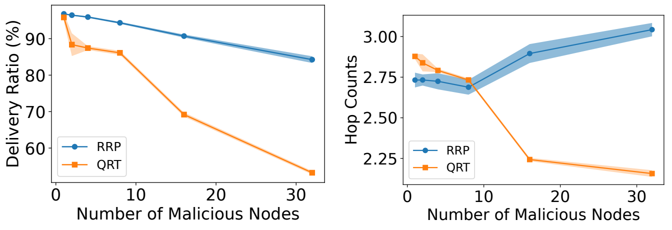


(a) (b)

Fig. 3: The average delivery ratio and hop counts under normal operation

C. Blackhole Attacks

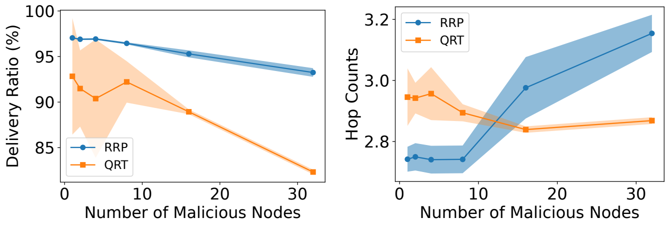
Blackhole attack is a well-known attack in WSN where compromised nodes drop all the received frames instead of forwarding them to the destination, which causes severe detrimental consequences, especially for medical applications [4]. In this experiment, the delivery ratio and the hop counts are evaluated 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 Fig. 4a and Fig. 4b. The results reveal an outstanding performance for RRP 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. In contrast with QRT, RRP 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 RRP 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. 4b shows that RRP performs better when there are up to 8 malicious



(a)

(b)

Fig. 4: The average delivery ratio and hop counts under blackhole attacks



(a)

(b)

Fig. 5: The delivery ratio and hop counts under selective forwarding attacks

SNs. When the number of malicious nodes increases, RRP 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 RRP chooses the most reliable shortest paths.

D. Selective Forwarding Attack

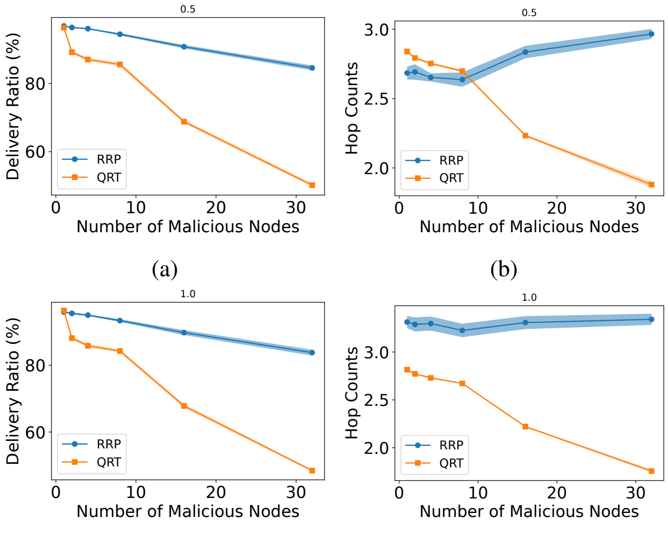
In selective forwarding attack, the malicious nodes forward some frames and drop others selectively [16]. This behaviour is hard to detect as the same malicious node could be trustworthy for some nodes and untrustworthy for others. In this experiment, RRP has been evaluated under selective forwarding attack, where malicious nodes randomly choose a list of neighbors not to relay their frames. The malicious node randomly chooses a list of several neighbors x_t^i to drop their frames at the beginning of the simulation. Fig. 5a shows the delivery ratio under selective forwarding attack. RRP outperforms QRT and provides a reliable delivery with minimal variability, while QRT shows a high variability when the number of malicious nodes is less than 25% of the total number of SNs, which indicates a converging difficulty. By increasing the number of malicious nodes, the delivery ratio of QRT decreases significantly. On the other hand, the hop counts results shown in Fig. 5b reveals how each protocol responds to the hostile environment. RRP performs better when the number of malicious nodes is less than 25%. Moreover, when the number of malicious nodes goes up to 50%, the hop counts gradually increase to avoid any path through malicious nodes with a slight increase in the variability, which indicates the alternative paths found out by RRP. In contrast, QRT needs more hop counts for the limited number of malicious nodes. Furthermore, it fails to find reliable paths as inferred from its low delivery ratio and hop counts.

E. Sinkhole Attacks

Sinkhole attack is one of the most destructive attacks on routing protocols. The malicious node attracts the network traffic by advertising false routing information [24]. 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 evaluate the optimal paths as described in algorithm 1. 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 RRP is evaluated under different sinkhole attacks. Four scenarios have been considered in this experiment. The malicious SNs 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 SNs 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. Fig. 6a and 6c show the delivery ratio for only two scenarios 50% and 100% due to space constraints. What stands out in these figures is the stable delivery ratio of RRP for different route poisoning levels, which reveals a high resiliency to sinkhole attacks. Moreover, Fig. 6b and 6d reveal how RRP finds the optimal paths through a hostile environment. RRP shows the same behaviour as previous experiments when the number of malicious SNs increases. It avoids malicious SNs by choosing the most reliable path with the minimal achievable hop counts. It is worth noting that in Fig. 6d when the malicious SNs advertise the value zero as their best estimation, RRP shows a slight increase in hop counts even for a low number of malicious SNs, but with a high delivery ratio. The reason behind this behaviour is that advertising this level of fake information affects the Q tables of the surrounding nodes, which makes the learning agent even tries to avoid the surrounding neighbors of malicious SNs.

F. On-Off Attacks

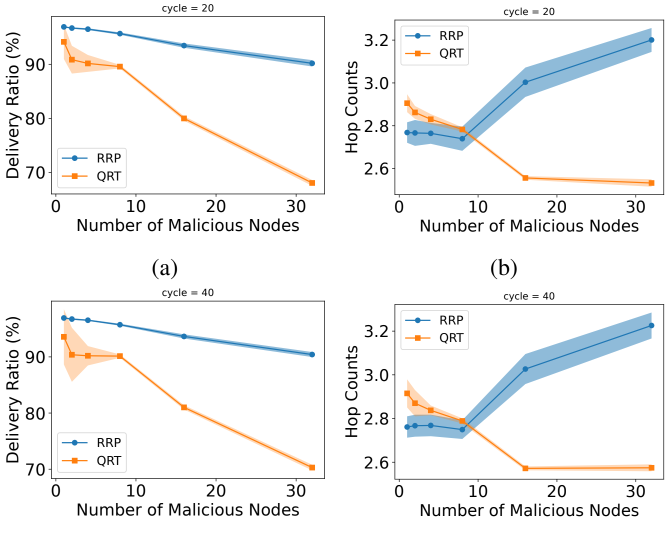
Although trust management schemes are used to detect malicious activities, they are vulnerable to on-off attacks, where smart adversaries can change their behaviour alternately with a view to cheating the TMS and keep themselves undetected [25]. The failure to detect on-off attacks negatively impacts the performance of trust-based routing protocols by making them take 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, we evaluate the performance



(a) (b)

(c) (d)

Fig. 6: The average delivery ratio and hop counts under sinkhole attacks



(a) (b)

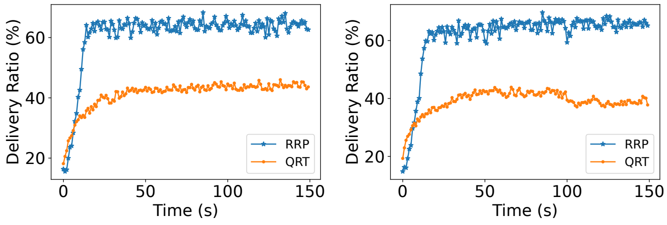
(c) (d)

Fig. 7: The delivery ratio and hop counts for different On-Off attacks' cycles

under different on-off attacks' cycles. The on-off attack's cycle varies from $10s$ to $40s$. Fig. 7a and 7c show the delivery ratio for only two cycles $20s$ and $40s$ due to space constraints, while Fig. 7b and 7d show the hop counts' results. RRP shows superior and stable performance for all on-off cycles. It achieved an average delivery ratio between around 90% and 97% for a variable number of malicious nodes. The average hop counts ranged between 2.75 and 3.2 with a tendency to use longer and more reliable paths when increasing the number of malicious nodes.

G. Network Dynamicity and Convergence

The convergence time is crucial 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. 8a shows the convergence time of both protocols. RRP is able to converge with less than $20s$ thanks to its asynchronous updating algorithm. However, QRT needs around double this time to converge. In the second scenario, the mobility has been introduced to study how both protocols re-converge in a dynamic environment. The patients can change their locations within the hospital ward. Therefore, in this experiment, two different patients change their locations at times $50s$ and $100s$. The patient could have up to 3 SNs. Thus, three simulations have been run for 1, 2, and 3 randomly chosen SNs. The results show a fast re-convergence in all cases. Fig. 8b shows the results for 3 SNs randomly chosen to change their locations at $50s$ and $100s$. RRP shows a slight decrease in delivery ratio during the movements. However, it recovers fast and re-converges again. On the other hand, QRT experienced a noticeable decrease with difficulty in re-converging, especially after the second movement.



(a) (b)

Fig. 8: The average convergence time

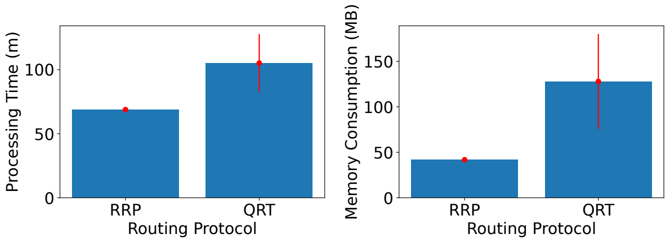
H. Computational Overhead

In this subsection, we compare the average processing time and memory consumption of both protocols RRP and QRT. The experiment was carried out on an Intel Core i5-8500T processor at 2.1GHz and 8GB RAM. The simulation has been run for 30 times, and then the results are 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 $\mu = 4p/s$ as QRT does not perform properly for lower traffic rates.

Fig. 9a shows the average processing time of RRP and QRT. The results show that QRT consumes more processing time than RRP. Moreover, results show high variability of around 23%. 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, RRP consumes less processing time and saves around

35% of the processing time of QRT. Moreover, RRP shows almost no variability, indicating the stability of performance and the ability to converge at approximately the same time for different simulation runs.

The second important performance metric is memory consumption. Average memory consumption was calculated and reported with one standard deviation in Fig. 9b. The memory allocation has been traced during the simulation using tracemalloc [26], a trace memory allocation module. Results show that QRT consumes a considerable amount of memory, around 128MB, with a high variability of around 52%. On the other hand, RRP is a memory conservative protocol. It consumes a decent amount of memory, around 42MB, which saves around 67% of the memory consumed by QRT. Moreover, RRP shows almost no variability, indicating that RRP did not experience any converging difficulties thanks to its novel updating mechanisms.



(a)

(b)

Fig. 9: The average processing time and memory consumption

VI. Conclusion and Future Work

There is still a persistent need for a lightweight and secure routing protocol for WMSN. Although many routing protocols have been proposed for WSN, they are not necessarily suitable for practically deploying on WMSN due to their different network operation conditions, network topology, resource constraints and sensitivity of applications. In this paper, we proposed a novel hybrid routing protocol that combines a new RL model design with an effective trust management scheme. Simulation results show a superior performance even under complicated attacks coupled with minimal resource footprint, making it a suitable candidate for WMSNs deployment. RRP will be further developed to consider more network operation parameters in the future, such as energy consumption. Moreover, the learning parameters used in this paper will be deeply investigated to find the optimal parameters setting.

References

[1] X. Li, B. Tao, H.-N. Dai, M. Imran, D. Wan, and D. Li, "Is blockchain for internet of medical things a panacea for covid-19 pandemic?" *Pervasive and Mobile Computing*, vol. 75, p. 101434, 2021.

[2] M. S. Hajar, H. Kalutarage, and 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.

[3] H. Fu, Y. Liu, Z. Dong, and Y. Wu, "A data clustering algorithm for detecting selective forwarding attack in cluster-based wireless sensor networks," *Sensors*, vol. 20, no. 1, p. 23, 2020.

[4] N. Khanna and M. Sachdeva, "A comprehensive taxonomy of schemes to detect and mitigate blackhole attack and its variants in manets," *Computer Science Review*, vol. 32, pp. 24–44, 2019.

[5] S. M. Altowaijri, "Efficient next-hop selection in multi-hop routing for iot enabled wireless sensor networks," *Future Internet*, vol. 14, no. 2, p. 35, 2022.

[6] W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10 737–10 750, 2021.

[7] R. Maivizhi and P. Yogesh, "Q-learning based routing for in-network aggregation in wireless sensor networks," *Wireless Networks*, vol. 27, no. 3, pp. 2231–2250, 2021.

[8] G. Künzel, L. S. Indrusiak, and C. E. Pereira, "Latency and lifetime enhancements in industrial wireless sensor networks: A q-learning approach for graph routing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5617–5625, 2020.

[9] Z. Ullah, I. Ahmed, F. A. Khan, M. Asif, M. Nawaz, T. Ali, M. Khalid, and F. Niaz, "Energy-efficient harvested-aware clustering and cooperative routing protocol for wban (e-harp)," *IEEE Access*, vol. 7, pp. 100 036–100 050, 2019.

[10] W. Guo, C. Yan, and T. Lu, "Optimizing the lifetime of wireless sensor networks via reinforcement-learning-based routing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, 2019.

[11] G. Liu, X. Wang, X. Li, J. Hao, and 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.

[12] Y. Naputta and 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.

[13] X. Liang, I. Balasingham, and 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.

[14] M. S. Hajar, M. O. AlKadri, and 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.

[15] IEEE, "Ieee standard for local and metropolitan area networks - part 15.6: Wireless body area networks," *IEEE Std 802.15.6-2012*, Feb 2012.

[16] M. S. Hajar, M. O. Al-Kadri, and H. K. Kalutarage, "A survey on wireless body area networks: architecture, security challenges and research opportunities," *Computers & Security*, p. 102211, 2021.

[17] T. Li, K. Zhu, N. C. Luong, D. Niyato, Q. Wu, Y. Zhang, and B. Chen, "Applications of multi-agent reinforcement learning in future internet: A comprehensive survey," *IEEE Comm.HI Surveys & Tutorials*, 2022.

[18] N. Azdad and M. Elboukhari, "Wireless body area networks for healthcare: Application trends and mac technologies," *International Journal of Business Data Communications and Networking (IJBDCN)*, vol. 17, no. 2, pp. 1–20, 2021.

[19] F. Yuan, J. Wu, H. Zhou, and 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.

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

[21] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," in *Annual conference on artificial intelligence*. Springer, 2011, pp. 335–346.

[22] N. Matloff, "Introduction to discrete-event simulation and the simpy language," *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, vol. 2, no. 2009, pp. 1–33, 2008.

[23] M. N. Islam and M. R. Yuce, "Review of medical implant communication system (mics) band and network," *Ict Express*, pp. 188–194, 2016.

[24] K. Prathapchandran and T. Janani, "A trust aware security mechanism to detect sinkhole attack in rpl-based iot environment using random forest–rftrust," *Computer Networks*, vol. 198, p. 108413, 2021.

[25] R. R. Sahoo, S. Sarkar, and 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.

[26] "Tracemalloc - trace memory allocations - python 3.10.2 documentation," accessed: 2022-02-08. [Online]. Available: https://docs.python.org/3/library/tracemalloc.html