International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 2194–2210

A Trust Based Dual Reinforcement Q-Routing protocol to detect misbehaving nodes in WMNs

*R.Thillaikarasi[1] and S.Mary Saira Bhanu[2]

[1]*Saranathan College of Engineering, Tiruchirappalli, TamilNadu, India*
[2]*National Institute of Technology, Tiruchirappalli, TamilNadu, India*

[1]*thillai-cse@saranathan.ac.in*, [2]*msb@nitt.edu*

Abstract

Nowadays, Wireless Mesh Networks (WMN) owing to its self adaptive nature are popular and used for diverse applications like community networks, broadband wireless access, E-healthcare etc.,. Secured routing in WMNs using cryptography, authentication is extensively addressed by researchers and the effects of misbehaving nodes that leads to the performance degradation of the network are explored mostly. The trustworthiness of every individual participating node while routing is essential to enhance the security of WMN. In this paper, a trust based dual reinforcement Q-Routing algorithm is proposed to detect the nodes performing packet dropping attacks in WMNs using cross layer approach. The Q-Routing algorithm uses reinforcement learning technique to detect the best path to reach the destination with minimal delivery time. But it is vulnerable to packet dropping attacks accomplished by misbehaving nodes in the network. The proposed model uses two trust components namely, behavioral trust and implied trust to compute the trust of the neighbor nodes by monitoring the behavior of nodes and links while forwarding the packets. The trust value of a link between the nodes computed at the data link layer and the trust value of the intermediate node computed at the network layer are the two cross layer metrics used by the proposed algorithm. These trust values are used to compute Q-Value of the nodes in addition to the minimal delivery time. Packet drops due to buffer overflow and energy depletion is also detected by this method which reduces false positive rate. The effect of packet dropping attacks on Q-Routing and the proposed Trust Based Dual Reinforcement Q-Routing (TBDQR) are analyzed using simulation. The experimental results confirm that TBDQR detects the packet dropping nodes and results in an increased packet delivery ratio compared with traditional Q-Routing.

Keywords: *Wireless Mesh Network, Reinforcement Learning, Q-Learning, Q-Routing, Blackhole attack, Greyhole attack, Cross Layer, Trust model*

1. Introduction

WMNs are multihop wireless networks capable of interconnecting heterogeneous type of networks together such as WiFi, WiMax, Wireless Sensor Networks (WSN) etc., together and hence supporting distinct type of network access [1]. WMNs are built with Mesh Routers (MR) that seamlessly routes internet traffic and Mesh Client (MC) such as laptop and mobile phones. WSN is yet another network which consists of sensors that are mainly used to collect data and send them to the internet via routers. WMNs can act as a gateway for WSNs to transfer the data over the internet where it is economically not feasible to do the same with older wireless technologies like WiFi and cellular communication. Similarly, due to the characteristics of WMNs like low cost and reliable nature nowadays they have become ideal for Internet of Things (IoT) applications like smart cities, health care, smart home, farming and industrial Internet [2]. WMNs are experiencing different types of security attacks like spoofing, eavesdropping, replay

* R.Thillaikarasi
ISSN: 2233-7857 IJFGCN

and content modification. Various techniques for detecting these attacks have been discussed at length. Cryptographic algorithms to ensure confidentiality and authentication to ensure the authenticity are used to overcome the above listed attacks. In spite of these techniques used for detecting these attacks, WMNs are still vulnerable to Denial of service attacks such as blackhole and greyhole. To overcome from these attacks trust should be ensured among all communicating nodes.

Routing in WMNs is a challenging task due to scalability and the unpredictable changes that happen in the network. The routing protocols of WMN are classified as adaptive or non-adaptive depending upon the means by which they respond instantaneously to the variations that occur in the network. Non-adaptive or static routing protocols, route the packets using a shortest path algorithm which exploits the information stored in the static table and routing decision is not based on dynamic changes that ensue in the network. During high load, heavy traffic might pass through some of the intermediate nodes participating in most of the transmissions. In such a situation, selecting an alternate path to reroute the packets certainly improves the network performance, but this is not feasible in the case of non-adaptive routing protocols [3,4].

Alternatively, an adaptive routing algorithm based on Reinforcement Learning (RL) technique [5] can be used to change their routing decision according to the topology and traffic changes of the network through exploration. Boyan and Littman have proposed Q-Routing [6] algorithm which uses RL technique to detect an optimal path to reach the destination dynamically. They have proved that Q-Routing is well suited for wireless environment due to its adaptive nature. It exploits forward learning technique where each receiver node sends a reply message with an estimated Q-Value to its sender node. Q-Value decides participation of the node in the routing path and these values are updated during every packet transmission.

During the learning process each participating node in communication is assumed to be trustworthy, but this not true in real world scenario. Nodes aiming to do blackhole attacks can participate in the route by sending a fake Q-Value with minimal delivery time to reach the destination during exploration and drop the data packets during exploitation. In case of greyhole attacks, some nodes after taking part in the route legitimately might start dropping the packets selectively or randomly. But Q-Routing lacks the provision to detect the malicious nodes and overcome these attacks. Secure communication is established only if all the communicating nodes are trusted. Comparing with security protocols based on cryptographic algorithms, the trust based security protocols are efficient and have minimal overhead in detecting malicious nodes.

Hence, this paper proposes a novel trust based dual reinforcement based Q-Routing algorithm to detect the blackhole nodes and reroute the packets through alternate routes. Though the trust can be characterized as reputation, opinion, probability or uncertainty [7], reputation is used as the trust in the proposed model. Since, only trusted nodes are allowed to participate in the route establishment the malicious packet dropping nodes are detected using the combination of two trust components namely, behavioral trust and implied trust. The behavioral trust is a cross layer metric which is the combination of the trust value of link and trust value of the nodes. Depending upon the forwarding nature of the link and node, their trust value is assigned either as 0 or 1. The implied trust is the Q-value computed by the next neighbor node in the route. Additionally, dual reinforcement technique is utilized in the proposed algorithm to improve the learning speed by learning in both forward and backward directions.
The rest of the paper is structured as follows: Related works are discussed in Section 2. In Section 3, Q-Learning, Q-Routing and packet dropping attacks are discussed. The proposed work is elaborated in Section 4. Section 5 describes the simulated environment and experimental results. Finally, Section 6 concludes the paper.

ISSN: 2233-7857 IJFGCN

2. Related Work

WMNs are vulnerable to both active and passive types of attacks and MR may be compromised by internal or external entities leading to performance degradation. In WMN routing, functionality of the nodes is affected by the frequent topological changes due to link failure, node failure, mobility and misbehavior of malicious nodes which degrades the network performance. Traditional routing protocols like Adhoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and Optimized Link State Routing (OLSR) cannot be used successfully due to their non- adaptive nature. Nowadays, Machine Learning (ML) techniques are used by most of the researchers because of their adaptability and learning ability.

Anna Forster [7] elaborates different machine learning techniques and appropriate use of those techniques. Q-learning [8], a reinforcement machine learning technique is appropriate for solving the problems like routing and congestion control. In this technique, the software agents running on the nodes are used to learn about the environment and do actions to improve reward in a non-deterministic way. Q-Routing uses forward Q-learning techniques to learn about the environment using local information and find the optimal path to the destination. Further enhancements in Q-Routing are:

- Predictive Q-Routing algorithm [9]: In this algorithm memory is used to record the best experience so far learned which is helpful to increase the learning speed and ability to tweak optimal routing policy under low load.
- Dual reinforcement Q-Routing Algorithm [10]: In Q-Routing the packet forwarding nodes receive routing information from the neighbor nodes (forward exploration) while transmitting the data packets. Additionally, the neighbor node receives routing information from the forwarding nodes (backward exploration) which significantly improves learning speed of the nodes and DRQ-Routing is able to sustain at higher loads than Q-Routing.
- Confidence based Dual Reinforcement Q-Routing [11]: Key improvement of this algorithm over Q-Routing is the quantity and quality of exploration increased by using the reliability of each Q-Value measured by means of confidence value. As compared with a static learning rate of Q-Routing, CDRQ uses dynamic learning rate, which leads to fast, adaptive and optimal routing policies.

Sun et al. [12] proposed Q-Map, a Q-learning based algorithm for multicast routing which combines routing and resource reservation. The above mentioned algorithms are based on single-agent RL model (SARL) and are not efficient to provide global optimizations. In the paper [13] the author proposes a Multi-Agent RL model (MARL) where each node exchanges Q-Value, reward and actions with neighbor nodes. This feedback assists the nodes to select nodes that could participate in routing.

Most of the researchers have contributed their research on providing routing algorithms using RL technique to enhance the security of wireless networks. These routing algorithms are used to detect malicious nodes and to overcome security attacks like DoS (Denial of Service) attack, packet dropping attacks etc. In the paper [14] the authors have proposed a cooperative reinforcement learning algorithm with distributed multiple detectors to exchange information which significantly improves the communication among the detectors without much overhead. Anitha et al. [15] proposed a self-adaptive Q-learning based trust computation of a node over Associativity Based Routing (ABR). The connectivity association between the nodes over time and space is measured as associativity metric and it is used to compute the trust value of the route. Direct and indirect trust values are combined together to compute the trust value of the nodes. The

ISSN: 2233-7857 IJFGCN

weighted average trust value of the nodes in the route and associativity ticks are used to find the secure route.

A detailed survey on detection of packet dropping nodes using various machine learning techniques like Support Vector Machine (SVM), decision tree and Q-learning is presented by Patel et al.[16]. A probabilistic approach proposed by Jin et al. [17] is used to detect node failures by means of location estimation and localized monitoring. When a node is unable to hear a heartbeat message from its neighbor, it detects that the neighbor node is failed or moved out of its transmission range using its own information and the knowledge gathered from its other neighbors. In paper [18] the authors have suggested the usage of RL technique over AODV protocol to detect malicious nodes in MANET which significantly improves routing in a trusted way. The state of a neighbor node is computed using the ratio of various messages such as Route request, Route Reply, Route Error received from the neighbor node to the total messages received from other neighbor nodes. Each ratio is classified according to the thresholds and the behavior of the neighbor node is detected as malicious or benign.

Irissappane et al. [19] proposed a trust based method to route the packets in WSN that uses fitness factors of the sensor nodes like energy, distance from the sink node and routing behavior for selecting the next participating node in the route. Special type of alarm messages are used to find the fitness factors. For instance node i before forwarding the packet to neighbor node j, it has to send a query message to node j and node k where k is neighbor to both i and j. After receiving ACK messages from the neighbor nodes, i will decide whether to forward the packet to node j or not. But this process will increase the network traffic and energy consumption apparently.

A light weight and Efficient trust based Security Routing algorithm based on Q-Learning (ESRQ) is proposed by Liu et al.[20] to route the packets in WSN. The sensor nodes update their trust values by considering the security factors, routing factors and energy factor. Using watchdog technique each forwarding node detects whether the neighbor node forwards the packet with / without modification or not forwarding to calculate the trust value of the neighbor node. Similarly, by using distinct communication between base station to all other nodes and by using private messages between each and every node, all the sensor nodes acquire distance, energy, recommendation and indirect trust values. This protocol promotes excessive use of message transmission between the nodes and threshold energy is not mentioned clearly.

The foreseen methods discussed in the literature survey for detecting malicious nodes have the following pitfalls:

- Q-learning technique has been used in routing protocols like AODV, ABR to provide secure routing but security issues in Q-Routing have not been discussed.
- Capable of detecting blackhole attacks but the solution to improve the performance of the network has not been discussed.

Anna Forster, in her paper [8] elaborates application of ML techniques to Wireless Ad-Hoc Networks and discusses about the merits and demerits of those techniques. From the survey it is obvious that RL is appropriate for routing due to the following properties: i) Medium memory requirements ii) Medium computational requirements iii) High tolerance towards topological changes iv) High optimality of the results v) Medium initial cost vi) Low additional communication cost. Q-Routing which is completely distributive in nature uses RL technique to learn about the environment dynamically using the local information and routes the packets through the path having least latency. It was observed from the literature survey that very few researches concentrate on providing security using Q-learning. Hence, these limitations motivated us to develop a trusted Q-Routing protocol to detect and overcome blackhole and greyhole attacks in WMNs.

ISSN: 2233-7857 IJFGCN

3. Reinforcement Learning

ML is defined by Tom Mitchell [21] as "*A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E"*. ML techniques are categorized as supervised, semi supervised, unsupervised and reinforcement Learning. In reinforcement learning the system is able to gain the knowledge from its prior communications with the environment and efficiently decide on the actions to be performed in future. Comparing with other ML techniques, RL has been considered as an appropriate learning technique for performing routing in WMNs because RL does not require any dataset in prior, since it learns from the local observations directly. Using exploration (trial and error interaction with the environment) RL decides the appropriate actions, receives a reward and learns how to resolve the given problem. Generally, RL technique is portrayed as a Markov Decision Process (MDP) having an agent, a finite set of possible states S, a finite set of possible actions A, probability of transition from state S to state S' for an action as P(S,a) and a reward function R for that action [22]. RL can be applied to various types of applications as routing, managing resources and selecting channels dynamically in wireless networks [23].

3.1. Q-Learning

Q-Learning is the popularly used RL technique in Wireless networks where each host is represented as a learning agent. The agent records the current state, the event and the reward received from the environment in a lookup table called Q-Table. This table helps to decide the subsequent action to be executed. The agent initially selects and executes an action with highest Q-Value greedily. After receiving the reward for that action, Q-Value is updated as shown in Equation 1.

$$Q^{new}(S_t, A_t) = (1-\alpha) \times Q^{old}(S_t, A_t) + \alpha \times [R_t + \gamma \times max_a Q(S_{t+1}, a)] \quad (1)$$

where R_t is the reward gained when moving from state S_t to state S_{t+1}, α is the learning constant used to decide learning speed varies from 0 to 1. γ is the discounting factor varies from 0 to 1. Since Q-Learning is an iterative algorithm, the Q-Table should be initialized before exploration.

3.2. Q-Routing

Boyan et al. [5] have proposed Q-Routing protocol as an application of Q-Learning algorithm to find an optimal path from source to destination. Prior knowledge about the network topology or traffic pattern is not essential for Q-Routing as it is adaptive in nature. A sample Mesh network is represented in Figure1 where MCs can be directly connected to MRs or through Access Points (AP). All MRs maintain a two dimensional Q-Table of size m x n where m is the total number destination nodes and n be the number of neighbor nodes in the network.

ISSN: 2233-7857 IJFGCN

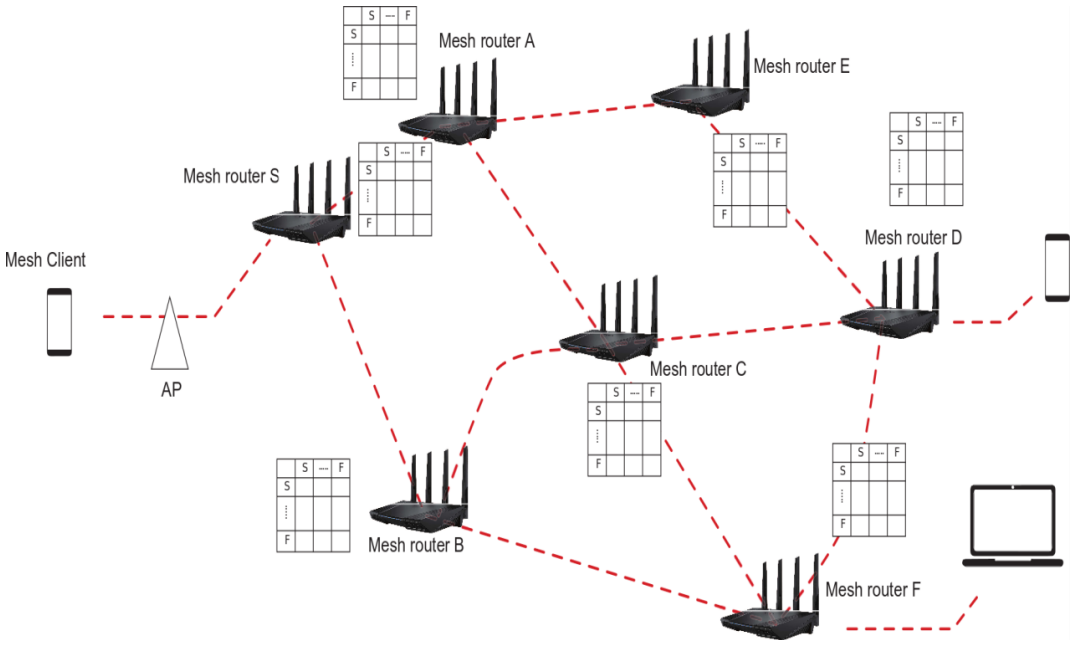

Figure1. Illustration of Q-routing

Let W be the set of all nodes of the WMN represented in Figure 1. The structure of the Q-Table maintained by a node x ϵ W is shown in Figure 2. Each cell of the Q-Table of node x is used to store a Q-Value which is represented as $Q_x(D,y)$ where D is the destination node and y is the next neighbor node of x. The Q-Table is initialized with zero and the source node S learns the route to transfer the packets to destination node D by using Q-Values which is the minimal delivery time to reach D.

Neighbor nodes

Destination nodes

	1	2	3	…	…	….	n
1	0	0	0	…	…	….	0
2	20	20	53	…	…	….	32
3	30	39	0	…	…	….	33
:	:	:	:	…	…	….	32
:	:	:	:	…	…	….	0
:	:	:	:	…	…	….	33
m	0	0	32	…	…	….	39

Figure2. Q-Table

With reference to Figure 1, let S, D are the two MRs that wish to accomplish communication and A, B, C, E and F the remaining MRs. There exist multiple paths to reach D from S and A, B are its neighbors. Initially, S randomly transmits the packet through B and receives the acknowledgement packet containing an estimate of the time required to reach D through B. The notation $Q_B(D,x)$ represents Q-Value required to transfer a packet from node B to node D through node x which is neighbor to node B. Node S receives B's estimate as T which is the remaining time left on the route to reach node D as shown in Equation 2.

$$T = \min_{x \in neihbors\ of\ B} Q_B(D,x) \quad (2)$$

Subsequently, node S updates its Q-Value $Q_S(D,B)$ using Equation 3

$$Q_S^{new}(D,B) = (1-\alpha) \times Q_S^{old}(D,B) + \alpha \times [T_q(S) + T + T_{tr}(S,B)] \quad (3)$$

ISSN: 2233-7857 IJFGCN

Adhering to Q-learning technique and equation (1), Q-routing is modeled as a Margov process where a state is a node, an action is the neighbor node selection based on the minimal time to reach the destination, the reward function is sum of $T_q(S)$ - the waiting time on node S's queue and $T_{tr}(S,B)$ is the transmission delay between node S and B and discount factor γ is set to 1 to strive for a high future reward. Then the MR B forwards the packet either to C or F depending upon node's minimum estimation to reach D and update the corresponding Q-Value. This process continues until D is reached. After few episodes all the nodes update their Q-Values to reach all other nodes. Any modification in the topology or traffic will be immediately reflected on the Q-Value during the exploitation phase and hence the best route is always selected for transmission.

4. Proposed Trust based Dual Reinforcement Q-Routing

Q-Routing algorithms discussed in section 3 are vulnerable to packet dropping attacks. There is no explicit route request or reply messages used by Q-Routing algorithms to find out the path between any two nodes. During data transmission each Forwarding Node (FN) should send data packet and time to reach the source node through it (backward learning). Every node learns the time to reach the Next Neighbor (NN) nodes using periodic transmission of heartbeat messages. Upon the reception of a data packet, NN should forward it to its NN and send an ACK packet to FN containing the estimated remaining time to reach the corresponding destination node. Every FN should update their Q-Value and nodes with minimal Q-Value which are used to transmit the next packet. Each and every participating node in the network is assumed to be trustworthy and hence all the nodes rely on the Q-Value they have received from their neighbors. But this is not true in a real life scenario where the malicious nodes intended to drop packets may send a fake minimal Q-Value to its FN. It may acquire participation in route and finally discard all the packets it receives.

This paper proposes a trust based DRQ algorithm to detect and isolate the nodes dropping the packets in the network. The criterion for evaluating the trust value is based on the behavior of the nodes and links which is transformed into a discrete quantity. The trust relationship between the nodes is investigated using the packet forwarding capability of the nodes. TBDQR utilizes a distributed design to build, maintain and update the trust values. Behavioral and implied trusts are the two key components used by the FN node while updating the Q-value and this is depicted in Figure 3.

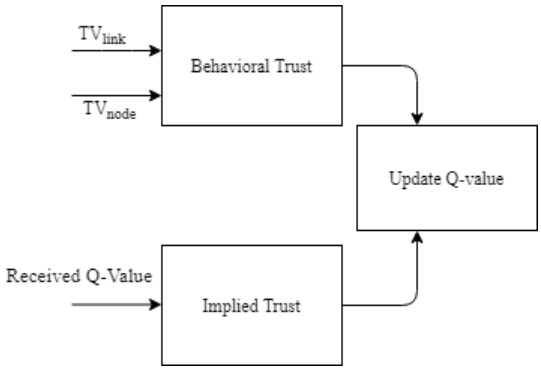

Figure 3. Computation of Trust value

FN computes the behavioral trust of the NN by considering the behavioral activities of NN and the communication link between FN and NN. Generally, the error free links forward the packet successfully to NN node and send back the corresponding Link level Acknowledgement (LL_ACK) to FN. Consequently, the trust value of the link is set as one and on the contrary, the link error reported sets the trust value of the link as zero. Similarly, the packet forwarding capability of the node decides the trust value of the node. Both the values are used to compute behavioral trust of NN by FN. The Q-value sent back in the upstream to the FN by the NN after

ISSN: 2233-7857 IJFGCN

receiving the packet is labeled as implied trust value. The FN node should update the Q-value using these trust values.

The working principle of the proposed algorithm uses DRQ algorithm with an additional security parameter S_p which is used to compute Q-Values. It is calculated by every forwarding node using the link and next node's forwarding capability. All the nodes are assumed to be operated in promiscuous mode to check the forwarding capabilities of their neighbor node. In promiscuous mode all the nodes are able to overhear the communication in its transmission proximity.

S_p is the combination of two cross layer metrics which are the trust value of the link (TV_{link}) and the trust value of the node (TV_{node}). The value for TV_{link} is set as one if FN receives LL_ACK for the forwarded data packet; otherwise TV_{link} is set as zero as shown in Equation 4.

$$TV_{link} = \begin{cases} 1 & if\ node\ receives\ ACK\ at\ DLL \\ 0 & otherwise \end{cases} \quad (4)$$

Forward Packet Buffer (FPB) is the data structure which is used to store the id of every packet transmitted by FN. If the FN overhears the packet with same id, then its NN is non-malicious and hence TV_{node} is set as one, otherwise TV_{node} is set as zero as shown in Equation 5. TV_{link} is a link level metric where as TV_{node} is a network layer metric and both are used to decide the route for packet at the network layer level.

$$TV_{node} = \begin{cases} 1 & if\ FN\ overhears \\ 0 & otherwise \end{cases} \quad (5)$$

FN computes the security parameter using Equation 6.

$$S_p = TV_{link} \times TV_{node} \quad (6)$$

where the TV_{link} is the trust value of the link between FN and Receiving Node (RN), TV_{node} is the trust value of RN. The value of S_p is one if both trust values are equal to one. Equation 7 is used to compute Q-Value of FN to the destination node D through the neighbor node (i.e RN).

$$Q_{FN}^{new}(D,RN) = S_p((1-\alpha) \times Q_{FN}^{old}(D,RN) + \alpha \times (T_q(FN) + \min_{z \in neighbors\ of\ RN} Q_{RN}(D,z) + T_{tr}(FN,RN)) \quad (7)$$

where T_q(FN) is the waiting time in FN's queue and T_{tr}(FN,RN) is the packet transmission time between FN and RN.

4.1. Working of TBDRQ algorithm

The advantage of Q-Routing compared with any other static routing strategy lies in is its ability to adapt to changes in its important system parameters during communication. Each node maintains a Q-Table to store Q-Values for all other nodes, which plays a vital role in making routing decisions. Besides, the Q-Values are updated every time when a packet is forwarded. Before communication, Q-Table should be initialized with random maximum values and these values are updated during packet forwarding. Since Q-Routing is RL based, there is no separation between exploration and exploitation phase.

Initially, when a node decides to transmit the packet, it selects one of its neighbors with the least Q-Value and sends the packet. Upon receiving the reply packet from RN containing the estimate to reach the destination, the corresponding Q-Value is updated. This process continues until the packet is received by the destination node. The behavior of the nodes during packet transmission is illustrated in Send_packet and Receive_packet algorithms. All the nodes initialize FPB as an

ISSN: 2233-7857 IJFGCN

empty buffer, set the flag variables TV_{link}, TV_{node} as one and declare an Audit Table (AT) to store packet id, source address and destination address of the overheard packets. The terminologies used in algorithms 1 and 2 are summarized in Table 1.

Table 1. Table of Notation

Notation	Description
S	Source Node
D	Destination Node
z, w	Neighbor Nodes of RN, FN respectively
AT	Audit Table
$Remain_{buffer}$	Remaining buffer capacity of RN
$Remain_{power}$	Remaining power of RN
$THmin_{buffer}$	Minimum Threshold buffer capacity to required to store packets = 2
$THmin_{energy}$	Minimum Threshold power required to transmit or receive packets = 0.002J
$Q_{FN}(S,w)$	Estimated Q-Value to transfer a packet from FN to S through w
$Q_{FN}(D,RN)$	Estimated Q-Value to transfer a packet from FN to D through RN
$Q_{RN}(D,z)$	Estimated Q-Value to transfer a packet from RN to D through z

Algorithm 1 explains the actions to be taken place while transmitting the packet from FN to RN. When a packet is forwarded from FN to RN forward exploration is accomplished at FN where as backward exploration is accomplished at RN.

Each forwarding node operated in promiscuous mode, pushes the packet identifier on to FPB before forwarding the packet to RN node. After receiving LL_ACK, it listens whether the RN node is forwarding the packet to its NN or not. Eventually, a non-malicious node forwards the packet to its NN node legitimately and hence FN overhears the packet, removing the packet id from FPB and set TV_{node} as one. Alternatively, a malicious node drops the packet without forwarding, but may send a fake estimate with minimal value to FN to participate in the route continuously. But FN has not overheard the packet, and hence set the value of TV_{node} as zero. The trust value of the node and link are computed and after receiving the estimate to reach the destination from RN, FN updates the corresponding Q-Value.

Algorithm 1: Send Packet(D,FN)
// Sending packet from FN to D through RN
1: FN selects neighbor node RN having minimal Q-Value > 0
if (RN found) then
    Push packet id on to FPB
    Send packet P to RN and $Q_{FN}(S,w)$ - minimum time to reach S from FN through w (neighbor of FN)
    if (not receive LL_ACK from RN) then
        Compute $Remain_{buffer}$ & $Remain_{power}$ of RN using AT
        if ($Remain_{buffer}$ < $THmin_{buffer}$ or $Remain_{power}$ < $THmin_{energy}$) then
            Set $Q_{FN}(D,RN)$ as maximum value // RN failed node
            Goto 1
        endif
    else

ISSN: 2233-7857 IJFGCN

```
                Receives an estimate Q_RN(D,z) from RN
                if (overheared packet's id is in FPB) then
                   TV_node = 1
                   Remove packet id from FPB
                else
                   TV_node = 0
                end if
                S_P = TV_link * TV_node
                Update Q_FN(D,RN) using Equation 7
        end if
else // if all neighbors are malicious
        FN increases its transmission frequency and uses two hop routing
end if
```

If FN fails to receive LL_ACK from RN which indicates that the packet may be dropped by RN due to the following reasons i) buffer overflow triggered by congestion ii) nodes may shutdown due to power failure and iii) malicious behavior of the nodes. Nodes performing packet dropping due to various reasons except malicious activity are also penalized as malicious nodes and introduce false positive rate if there is no appropriate mechanism to identify the actual cause for the packet drops. In order to evade this, the proposed work uses an AT.

FN computes the remaining buffer capacity and the remaining battery power of RN approximately from the number of packets transmitted and received by RN recorded in its AT. If the estimated values are lower than the threshold minimal power and threshold buffer capacity then it is obvious that the reason for not forwarding the packet by RN is not due to the malicious behavior. Hence, such nodes are identified as failed nodes rather than malicious nodes. If RN is identified as a failed node then the corresponding Q-Value in FN is set with maximum value so that RN is not isolated as malicious and moreover it will not be a participating node for a while until it recovers from power failure or buffer overflow problem. If RN is malicious node, it will drop the packets forwarded through it and the corresponding Q-Value in FN is set to zero. Algorithm 2 explains the actions carried out by RN while receiving the packet from FN.

Algorithm 2: Receive Packet(S,RN)

```
// RN receives the packet from the source node S through FN
RN receives the packet P and estimated Q_FN(S,w)
Update Q_RN(S,FN) using the new Q_FN(S,w)
if (RN is Destination) then
        Consumes the packet
else
        Return Q_RN(D,z) to FN
        if (Malicious) then
                Drop the packet
        else
                Forward the packet to node z
        end if
end if
```

Each RN consumes the packet if it is destined for it or else forwards it to its next neighbor. In addition to the packet to be forwarded to the destination, RN also receives the estimated minimal time to reach the source node through FN. The corresponding Q-Value is updated (backward exploration) in the Q-table and RN returns back its estimated Q-Value to reach the destination to FN. If a node is a compromising node and if it aims for blackhole attack then it will drop all the

ISSN: 2233-7857 IJFGCN

packets forwarded through it. Consequently, the neighbor which forwards the packet to malicious node will set the corresponding Q-value as zero. The main difference between blackhole attacks and greyhole attacks is that all Q(D,RN) entries of FNs will be zero, which isolates RN from the network in case of blackhole attacks but only a few entries of Q(D,RN) will be zero for greyhole attacks.

If every participating MR follows the TBDQR method for packet transmission then the network performance in terms of Packet Delivery Ratio (PDR) and the throughput will be 100%. On the contrary, in the worst case scenario where all the neighbor nodes are malicious, FN may expand its transmission range by increasing the transmission frequency and two hop routing technique may be adopted. Furthermore, using this technique it is possible achieve better network performance with the overhead of network delay.

5. Simulation & Results

The proposed TBDRQ protocol is simulated using network simulator NS3.20 which is an event simulator. Either C++ or Python program is used to simulate any type of networks and routing protocols. The proposed TBDQR algorithm is simulated using C++ programs. The assumptions considered in this work are i) Packet forwarding operation is taken place at the backbone of the WMN network, which consists of MRs. ii) MRs are static and they do not have any constraint on power. So mobility and node failure due to power failure are not considered in this work. iii) All nodes operate in promiscuous mode. iv) Some of the nodes are selected randomly and they are programmed to act as malicious nodes by dropping the data packets that passes through them. v) Neither source nor destination acts as a malicious node.

5.1. Simulation Environment

A total of 16 nodes are arranged into a 4 x 4 mesh topology in the space of 1600m x 1600m and they are equipped with one 802.11b interface. The nodes can transmit the CBR packets of size 512bytes with the data rate of 1Mbps. The transmission range of a node is set as 500m and the total simulation time is varied according to the number of packets and number of transactions carried out during the simulation. The following table 2 summarizes the simulation parameters.

Table 2. Simulation Parameters

Simulation parameters	Values
Simulation Area	1600m x 1600m
No. of Nodes	16
Malicious nodes	0 - 6
No. of Transmissions	3 - 10
Traffic Type	CBR traffic
Packet Size	512 bytes
Pause time	1sec
Mobility model	Constant mobility model
Data link type	802.11b
Learning Rate	0.8
Simulation time	100sec – 200sec

5.2. Experimental Results and Discussion

The proposed TBDQR is able to identify the packet dropping nodes, whereas DQR is unable to identify them leading to packet loss and degradation of network performance. This has been proven by conducting the following experiments.

ISSN: 2233-7857 IJFGCN

Experiment 1:

The performance of TBDQR is compared with DQR using the metric Packet Loss Ratio (PLR) which is the ratio of the number of packets dropped to the total number of packets transmitted. CBR flow has been initiated by eight different sources and out of sixteen nodes ten nodes have been set as source and destination. Remaining six nodes are set to be malicious in a random fashion in an increasing number at each step of simulation run and the difference in PLR between the proposed algorithm and DQR algorithm is depicted in figure 4.

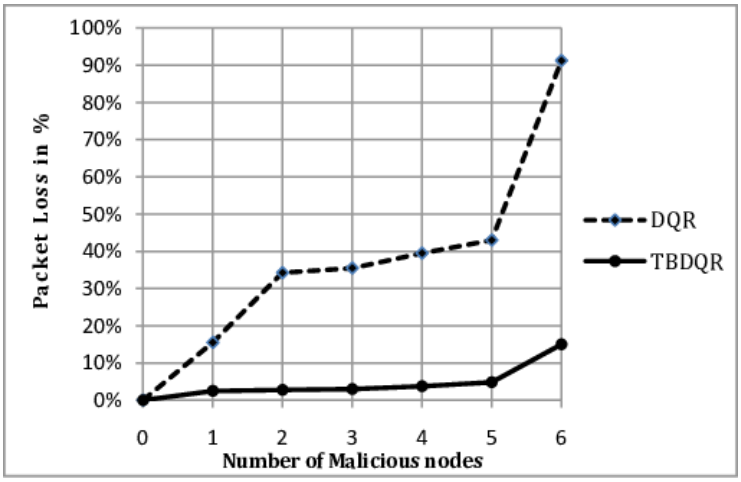

Figure 4. Comparison of Packet Loss Ratio

The packet loss ratio increases in both the cases when the number of malicious nodes increased from 0 to 6. Since DQR fails to detect the malicious nodes, packet loss will be more when the number of blackhole nodes increases gradually. Eventually, TBDQR after transmitting a packet will check whether the next node forwards the packet or not. If the next node is malicious or deteriorated due to buffer overflow or energy depletion then the packet will be dropped and so there is a negligible raise in PLR. After detecting the malicious nodes, the corresponding Q-Values in FN's Q-Table will be set as zero which prevents the forthcoming packets from being not to be forwarded through that malicious node. The remaining packets are forwarded to the destination through an alternate path, which improves PDR as shown in Figure 5.

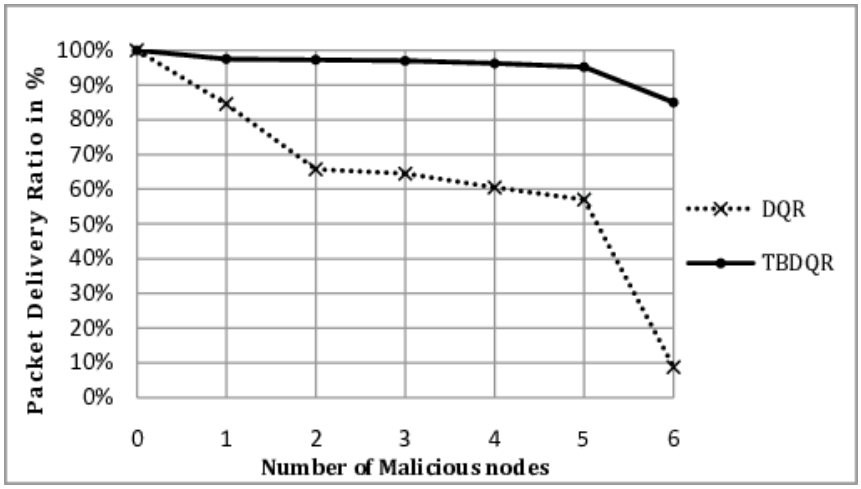

Figure 5. Comparison of Packet Delivery Ratio

ISSN: 2233-7857 IJFGCN

PDR is the ratio between the number of packets successfully received by the destination nodes and the total number of packets transmitted by the source nodes. In each simulation run 400 packets of size 512 bytes were transmitted through eight different CBR traffic and PDR of DQR is compared with proposed algorithm. As the number of malicious nodes gradually increases the total number of packets dropped by the misbehaving nodes also increases causing reduction in PDR of DQR. Except the source and destination when the remaining six nodes are turned to be malicious, PDR of DQR dropped below 10% whereas in TBDQR it is above 85%.

In TBDQR once the malicious node is detected the remaining packets are forwarded through the alternate paths identified using Q-Table. The percentage of packets rerouted through the alternate paths while increasing the number of malicious nodes is shown in Figure 6. Since there is no malicious node detection technique in DQR, the packets are dropped by the malicious nodes and hence percentage of packets rerouted is zero.

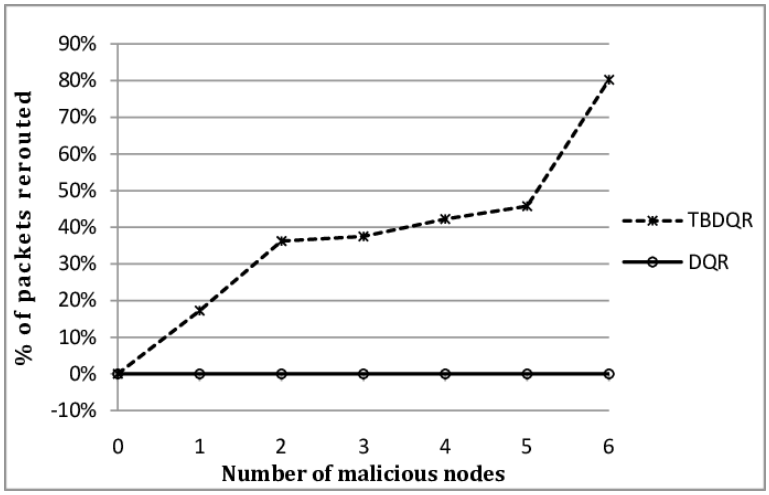

Figure 6. Packets rerouted through alternate paths

Percentage of packets rerouted purely depends on the number of malicious nodes present in the path. According to the scenario selected for the simulation, node 6 is the most frequently participating node in almost all CBR traffic and hence when it is turned to be malicious along with other five nodes approximately 90% of the packets are rerouted.

Experiment 2:

The effect of increasing the number of CBR traffic from 1 to 8 with transmissions of 50 packets of size 512bytes each and the effect of six malicious nodes on PLR is recorded for both algorithms and the same is depicted in Figure 7.

ISSN: 2233-7857 IJFGCN

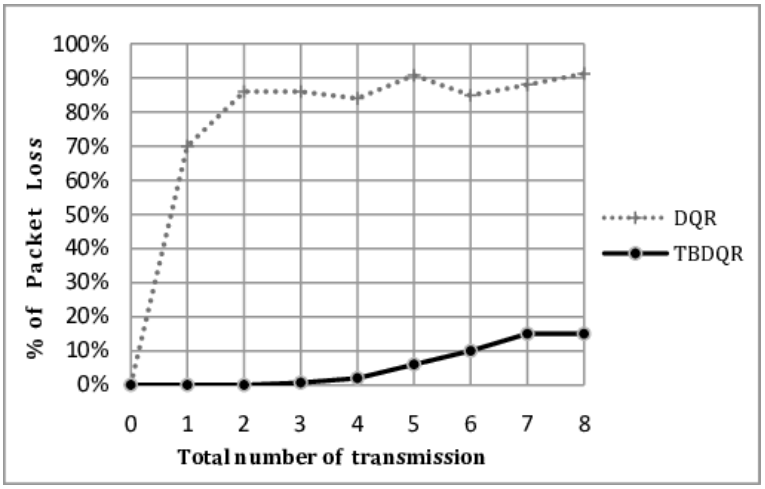

Figure 7. PLR versus Number of CBR traffic

While increasing the total number of packets transmitted, due to the delayed delivery time experienced by some of the nodes an alternate path through a malicious node may be selected in DQR which increases PLR. Approximately PLR is above 80% in the case of DQR where as it is below 20% in the case of TBDQR since there is a mechanism to evade malicious or failed nodes from the path. Nevertheless, dropping of first packet at malicious nodes is the reason for negligible packet loss in TBDQR. This evidence obviously shows that minimal PLR is observed by the proposed algorithm as compared with DQR.

Experiment 3:

The impact of increasing the traffic on DQR and TBDQR is compared by conducting the simulation of ten CBR traffic flows with increasing number of packets transmitted as 10,20,...100 by each source node with 6 malicious nodes. Figure 8 describes the performance of DQR compared with TBDQR using PLR computed by varying the number of packets transmitted by each source node.

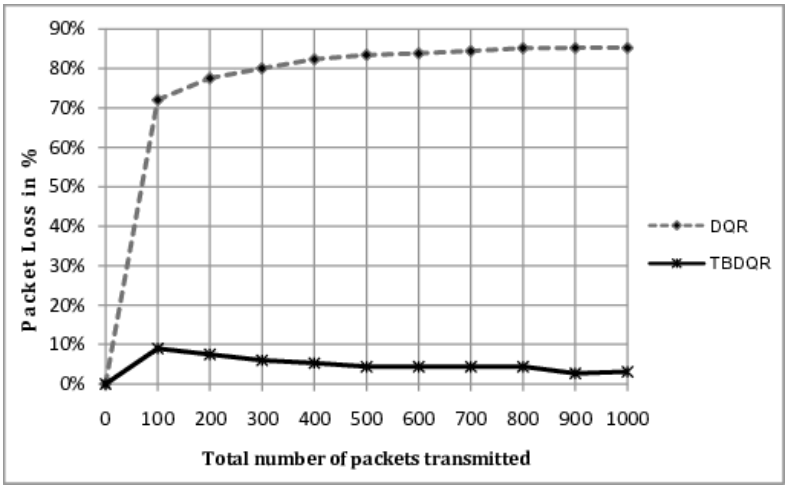

Figure 8. PLR versus Total number of packets transmitted

Since the proposed algorithm learns about the whole network with minimal packet loss, packet delivery to the destination nodes are high compared with DQR in all the cases. TBDQR experiences PLR below 10% whereas PLR of DQR is above 70% when traffic increases.

ISSN: 2233-7857 IJFGCN

Moreover, network delay is yet another metric which is used to analyze the performance of any routing algorithm and it is the delay experienced by the packet to travel from source to destination. Network delay for the proposed algorithm is analyzed to check whether there will be a phenomenal delay experienced by this method while checking for packet drops. The experimental setup of having eight CBR traffic with three malicious nodes is considered. Simulation is accomplished by varying the number of packets transmitted through individual CBR traffic and the time taken by DQR and TBDQR algorithms to transmit the packets are recorded. The effect of detection of malicious nodes on network delay is plotted in Figure 9.

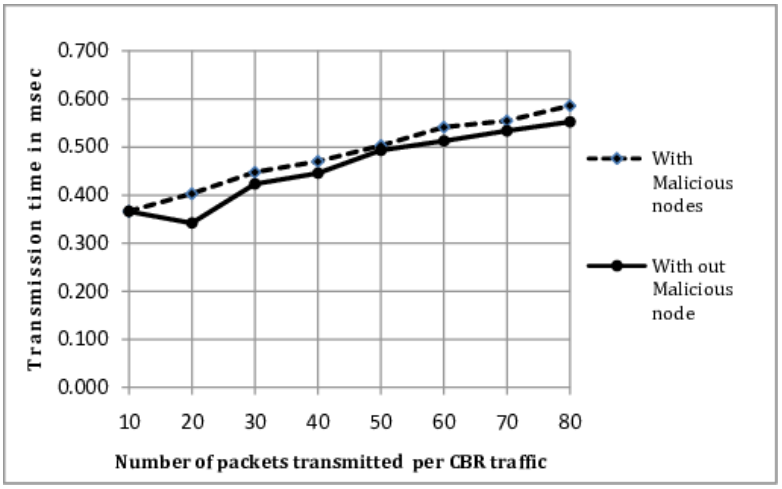

Figure 9. Transmission Time versus Number of malicious nodes

From the simulation it is observed that the time difference between transferring the packets in WMN with or without malicious nodes is absolutely minimal and it is not more than 0.03msec.

6. Conclusion

The Q-Routing algorithm uses reinforcement learning technique to learn about the network topology using local information and it is efficient to find out the routes adaptively in WMN. The packets are routed through the fastest routes in Q-Routing instead of shortest routes used by well known WMN routing protocols such as DSR, AODV etc. Unlike the traditional Q-Routing based algorithms, the proposed TBDQR is designed in such a way to thwart packet dropping attacks using cross layer approach. The Q-Value of a node is updated with the trust of a node (estimated at network layer), the trust of a link (estimated at data link layer) and minimal time to reach the destination. Finally, the simulation and results proved that the proposed algorithm is capable of detecting blackhole nodes and reroute the packets through alternate paths. The major benefit of TBDQR algorithm is its robustness to node misbehavior and its adaptive nature which results in improved packet delivery ratio than any other Q-Routing algorithms. However detection of packet modification attacks, co-operative blackhole attacks, Sybil attacks can be carried out as future work.

References

[1] Ian F Akyildiz, Xudong Wang, and Weilin Wang, “Wireless mesh networks: a survey”, Computer networks, 47(4), **(2005)**, pp. 445 – 487.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito, “The internet of things: A survey”, Computer networks, 54(15), **(2010)**, pp.2787 -2805.

[3] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. "RFC3561: Ad hoc on-demand distance vector (AODV) routing." **(2003)**.

ISSN: 2233-7857 IJFGCN

[4] Clausen, Thomas, and Philippe Jacquet, eds. "RFC3626: Optimized link state routing protocol (OLSR)." **(2003)**.

[5] Richard S Sutton and Andrew G Barto, "Reinforcement learning: An introduction", MIT press, **(2018)**.

[6] Justin A Boyan and Michael L Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach", In Advances in neural information processing systems, **(1994)** pp. 671-678.

[7] Shantanu Konwar, Amrita Bose Paul, Sukumar Nandi, and Santosh Biswas, "MCDM based trust model for secure routing in wireless mesh networks", In 2011 World Congress on Information and Communication Technologies, IEEE, **(2011)**, pages 910-915.

[8] Anna Forster, "Machine learning techniques applied to wireless ad-hoc networks: Guide and survey", 3rd international conference on intelligent sensors, sensor networks and information, IEEE, **(2007)**, pp. 365-370.

[9] Samuel PM Choi and Dit-Yan Yeung, "Predictive q-routing: A memory-based reinforcement learning approach to adaptive traffic control", In Advances in Neural Information Processing Systems, **(1996)**, pp. 945-951.

[10] Shailesh Kumar and Risto Miikkulainen, "Dual reinforcement q-routing: An on-line adaptive routing algorithm", Proceedings of the artificial neural networks in engineering Conference, **(1997)**, pp. 231-238.

[11] Shailesh Kumar and Risto Miikkulainen, "Confidence based dual reinforcement q-routing: An adaptive online network routing algorithm", In IJCAI, volume 99, Citeseer, **(1999)**, pp.758-763.

[12] Ruoying Sun, Shoji Tatsumi, and Gang Zhao, "Q-map: A novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning", Proceedings 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM'02, vol 1, IEEE, **(2002)**, pp. 667-670.

[13] Xuedong Liang, Ilangko Balasingham, and Sang-Seon Byun, "A multi-agent reinforcement learning based routing protocol for wireless sensor networks", IEEE International Symposium on Wireless Communication Systems, **(2008)**, pp. 552-557.

[14] Xin Xu, Yongqiang Sun, and Zunguo Huang, "Defending DDoS attacks using hidden Markov models and cooperative reinforcement learning", In Pacific-Asia Workshop on Intelligence and Security Informatics, Springer, **(2007)**, pp. 196-207.

[15] Anitha Vijaya Kumar and Akilandeswari Jeyapal, "Self-adaptive trust based ABR protocol for MANETs using q-learning", The Scientific World Journal, **(2014)**, pp. 1-9.

[16] Nirav J Patel and Rutvij H Jhaveri, "Detecting packet dropping nodes using machine learning techniques in mobile ad-hoc network: A survey", In 2015 International Conference on Signal Processing and Communication Engineering Systems, IEEE, **(2015)**, pp. 468-472.

[17] Ruofan Jin, Bing Wang, Wei Wei, Xiaolan Zhang, Xian Chen, Yaakov Bar-Shalom, and Peter Willett, "Detecting node failures in mobile wireless networks: a probabilistic approach", IEEE Transactions on Mobile Computing, 15(7), **(2015)**,1647-1660.

[18] Hansi Mayadunna, Shanen Leen De Silva, Iesha Wedage, Sasanka Pabasara, Lakmal Rupasinghe, Chethena Liyanapathirana, Krishnadeva Kesavan, Chamira Nawarathna, and Kalpa Kalhara Sampath, "Improving trusted routing by identifying malicious nodes in a MANET using reinforcement learning", In 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, **(2017)**, pp.1-8.

ISSN: 2233-7857 IJFGCN

[19] Athirai A Irissappane, Jie Zhang, Frans A Oliehoek, and Partha S Dutta, "Secure routing in wireless sensor networks via POMDPS, In Twenty-Fourth International Joint Conference on Artificial Intelligence, **(2015)**.

[20] Gaosheng Liu, Xin Wang, Xiaohong Li, Jianye Hao, and Zhiyong 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/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, **(2018)**, pp. 149-155.

[21] what-is-machine-learning. Available at https://machinelearningmastery.com/what-is-machine-learning/, Last Updated on April 16, **(2018)**.

[22] Carlos Henrique Costa Ribeiro, "A tutorial on reinforcement learning techniques", In Supervised Learning Track Tutorials of the 1999 International Joint Conference on Neuronal Networks, **(1999)**.

[23] Kok-Lim Alvin Yau, Peter Komisarczuk, and Paul D Teal, "Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues", Journal of Network and Computer Applications, 35(1), **(2012)**, 253-267.

ISSN: 2233-7857 IJFGCN