Computing
DOI 10.1007/s00607-014-0438-1

Application of reinforcement learning to wireless sensor networks: models and algorithms

Kok-Lim Alvin Yau · Hock Guan Goh · David Chieng · Kae Hsiang Kwong

Received: 28 December 2012 / Accepted: 13 December 2014

Abstract Wireless sensor network (WSN) consists of a large number of sensors and sink nodes which are used to monitor events or environmental parameters, such as movement, temperature, humidity, etc. Reinforcement learning (RL) has been applied in a wide range of schemes in WSNs, such as cooperative communication, routing and rate control, so that the sensors and sink nodes are able to observe and carry out optimal actions on their respective operating environment for network and application performance enhancements. This article provides an extensive review on the application of RL to WSNs. This covers many components and features of RL, such as state, action and reward. This article presents how most schemes in WSNs have been approached using the traditional and enhanced RL models and algorithms. It also presents performance enhancements brought about by the RL algorithms, and open issues associated with the application of RL in WSNs. This article aims to establish a foundation in order to spark new research interests in this area. Our discussion has been presented

K.-L. A. Yau (✉)
Faculty of Science and Technology, Sunway University, No. 5 Jalan Universiti, Bandar Sunway, 46150 Petaling Jaya, Selangor, Malaysia
e-mail: koklimy@sunway.edu.my

H. G. Goh
Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia

D. Chieng
Wireless Communication Cluster, MIMOS Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia

K. H. Kwong
Recovision R&D, No. 1, Jalan Putra Mahkota 7/8D, Putra Heights, 47650 Subang Jaya, Selangor, Malaysia

Published online: 28 December 2014

in a tutorial manner so that it is comprehensive and applicable to readers outside the specialty of both RL and WSNs.

Keywords Wireless sensor networks · Reinforcement learning · Q-learning · Artificial intelligence · Context awareness

Mathematics Subject Classification 68T05

1 Introduction

Wireless sensor network (WSN) [1] is comprised of a large number of sensors and sink nodes to monitor events or environmental parameters, such as temperature and humidity, in a collaborative manner. The sensor nodes collect data and send it to the destination sink nodes in single or multiple hops; while the sink nodes process the data in order to provide meaningful information to end users. The WSN has seen numerous potential applications in medical field, disaster recovery and wildlife monitoring.

Generally speaking, each sensor node operates on battery power. Two main factors affect energy consumption. Firstly, the state of the transceiver in which energy consumption is high during transmission, reception, idle (or overhearing), and low during sleeping. Secondly, events other than successful packet transmission, including collision, retransmission and control packet transmission, incur energy consumption. Enhancing energy efficiency to prolong network lifetime without jeopardizing network performance has attracted a considerable research attention, and has been part of the objective of most schemes in WSNs because sensor nodes may be deployed at hard-to-reach areas.

In recent years, there has been an increasing interest in the application of an artificial intelligence approach called Reinforcement learning (RL) [2] to various schemes in WSNs in order to improve network performance. The RL approach adopts an unsupervised and online learning technique. Through unsupervised learning, external teacher or critic is not required to oversee the learning process; and so, a decision maker (or an agent) must make its own efforts to learn knowledge about the operating environment. Through online learning, an agent acquires knowledge on the fly while carrying out its normal operation; and so, empirical data or experimental results from the laboratory are not required. A wide range of schemes can be represented using RL models, and subsequently various network performances can be improved using RL algorithms.

Although extensive research has been carried out on a wide range of schemes in WSNs, no single study exists which adequately covers distinctive RL models and algorithms that have been applied, and so this is the focus of this article. The rest of this article is organized as follows. Sections 1.1 and 1.4 present an overview of RL and application schemes of WSNs, respectively. In the context of WSNs, Sect. 2 presents various components, features and enhancements of RL, while Sect. 3 presents various RL models and algorithms. Section 4 presents performance enhancements brought about by RL in various schemes. Section 5 presents open issues, and finally Sect. 6 presents conclusions.

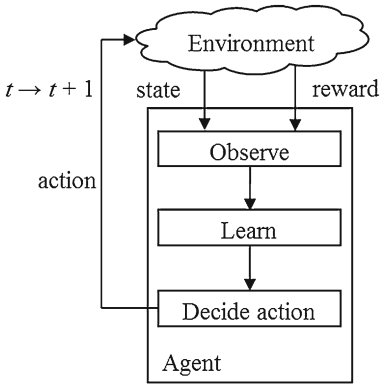

Fig. 1 A simplified RL model

1.1 Overview of reinforcement learning

This section presents an overview of the RL model and Q-learning.

1.2 RL model

Figure 1 presents a simplified version of a RL model. The purpose of RL is to estimate the long-term reward of each state-action pair through trial-and-error interactions with the operating environment. There are three main representations. Firstly, state represents the decision-making factors (or the operating environment) under consideration being observed by an agent. Examples are residual energy and the number of packets in the buffer queue. Secondly, action represents an optimal action being selected by the agent, which may change or affect the state and reward. Examples are selecting transmission power and selecting a next-hop node for packet transmission. Thirdly, reward represents the gains or losses in network performance for taking an action on a particular state in the previous time instant. Examples are throughput and energy consumption level.

At any time instant, an agent observes state and reward from its operating environment, learns the long-term reward of each state-action pair, decides and carries out an appropriate action on the environment so that the state and reward, which are the consequences of the action, improve in the next time instant. The agent interacts with the operating environment in a trial-and-error manner, and so given a particular state, an agent learns to carry out the optimal action as time progresses in order to improve the next state and reward.

The RL model in Fig. 1 can be embedded in each sensor node [3], or in the surrounding area of a sensor node [4]. For instance, each sensor node keeps track of the reward in regards to each neighboring sensor node in [5], and for each grid point in its surrounding operating environment in [4]. As an example on the application of RL in WSNs, it is used to learn the optimal route in routing (see Sect. 1.4). The state represents a destination (or sink) node, action represents the selection of a next-hop node to forward packets, and reward represents the progress in terms of the physical

distance towards the sink node. Maximizing reward reduces distance to the sink node, which enhances network performance.

There are two main advantages of RL. Firstly, it models network performance which covers most factors affecting the performance rather than each of the factors itself, and so this simplifies the design. Secondly, it learns on the fly during normal operation, and so it does not require prior knowledge of the operating environment. For instance, a sleep-wake scheduler aims to reduce energy consumption through sleeping for the right duration at the right time, and so traffic loads at neighboring nodes are pertinent to determine this duration although this information may not be known to the scheduler.

1.3 Q-learning

Q-learning [6] is a popular technique in RL. Denote decision epochs by $t \in T = \{1, 2, \ldots\}$, each agent i updates the Q-function $Q^i_{t+1}(s^i_t, a^i_t)$ of a particular state-action pair at time t as follows:

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

where $s^i_t \in S$ is state, $a^i_t \in A$ is action, $r^i_{t+1}(s^i_{t+1}) \in R$ is delayed reward, $0 \leq \gamma \leq 1$ is discount factor, and $0 \leq \alpha \leq 1$ is learning rate. Note that, the delayed reward $r^i_{t+1}(s^i_{t+1})$ for action selection at time t is dependent on the state at time $t+1$ and so it is received at time $t+1$. Also note that, higher γ value causes greater dependency on the discounted future reward $\gamma \max_{a \in A} Q^i_t(s^i_{t+1}, a)$ rather than the delayed reward $r^i_{t+1}(s^i_{t+1})$; while higher α value causes greater dependency on the delayed reward $r^i_{t+1}(s^i_{t+1})$ and the discounted future reward $\gamma \max_{a \in A} Q^i_t(s^i_{t+1}, a)$ rather than the Q-value $Q^i_t(s^i_t, a^i_t)$ at time t.

An agent i observes state s^i_t from the operating environment and chooses an action a^i_t at decision epoch t. The state s^i_t changes to s^i_{t+1} at decision epoch $t+1$. Subsequently, the agent receives delayed reward $r^i_{t+1}(s^i_{t+1})$ and updates Q-value $Q^i_{t+1}(s^i_t, a^i_t)$ using Eq. (1). The Q-value $Q^i_{t+1}(s^i_t, a^i_t)$ is updated using the maximum discounted future reward $\gamma \max_{a \in A} Q^i_t(s^i_{t+1}, a)$ as the agent takes the optimal action in any future states at time $t, t+1, \ldots$. As time progresses, the agent receives a sequence of rewards which contribute to the convergence of the Q-values to long-term rewards. The agent chooses an optimal action through maximizing value function $V^\pi(s^i_t)$ as shown below:

$$V^\pi(s^i_t) = \max_{a \in A}(Q^i_t(s^i_t, a)) \quad (2)$$

Hence, agent i's policy is as follows:

$$\pi_i(s^i_t) = \text{argmax}_{a \in A}(Q^i_t(s^i_t, a)) \quad (3)$$

In some cases, negative reward represents cost, which must be minimized, and so $V^\pi(s^i_t) = \min_{a \in A}(Q^i_t(s^i_t, a))$ and $\pi_i(s^i_t) = \text{argmin}_{a \in A}(Q^i_t(s^i_t, a))$. Note that, choosing the optimal action using Eq. (3) at all times does not update the Q-values of other

actions, which may cause the agent to converge to local optimal solutions. Hence, there are two methods for action selection. Exploitation chooses the best-known optimal action for performance enhancement; while exploration chooses the other actions once in a while to update the Q-values of other actions so that better actions may be discovered.

1.4 Application schemes of wireless sensor networks

Reinforcement learning is a versatile and universal solution to most problems and open issues associated with the dynamicity and uncertainty of the operating environment. RL has been applied in various schemes in WSNs as follows:

A.1 *Medium access control (MAC)* MAC protocols coordinate channel access among multiple nodes in a single-hop transmission to reduce collisions. Two main functions are sleep-wake scheduler [7–9] and transceiver selector [10] as follows:

A.1.1 Sleep-wake scheduler arranges the transmission, reception, idle and sleeping time durations. During the idle mode, sensor nodes listen for potential packet transmissions and the energy consumption is almost identical to that of receive mode. To reduce energy consumption, a sleep-wake scheduler schedules sleeping and waking (i.e. transmission, reception and idle) time durations. There are two main purposes in sleep-wake scheduling. Firstly, longer waking time duration (or higher duty cycle) increases bandwidth availability leading to higher throughput and lower packet latency; however, it increases energy consumption. The waking time duration may increase with network traffic load [8,9] or Quality of Service (QoS) requirements [11]. RL has been applied to minimize collisions and energy consumption in slot assignment [7], as well as to estimate traffic arrivals from neighboring nodes in order to adjust the sleeping and waking time durations [8,9]. Secondly, a mobile data collector node moves within an area to collect sensing outcomes from static sensor nodes [12]. RL has been applied in each sensor node to learn the waking time duration based on the arrival pattern of the mobile data collector node [12] in order to increase in-contact with the mobile data collector node while reducing energy consumption.

A.1.2 *Transceiver selector* selects either a long-range or short-range radio for data and control packet transmissions. Long-range (short-range) radio uses higher (lower) transmission power. To reduce energy consumption, a transceiver selector switches in between the transceivers based on physical range (e.g. whenever a mobile node moves from one effective transmission range to another) and channel conditions (e.g. fading, interference, shadowing, and multi-path effects) [10].

A.2 *Cooperative communications* select cooperative forward packets towards sink nodes in order to ameliorate the effects of deteriorating channel conditions and changes in network topology. For instance, in forwarding nodes to Fig. 2, the direct transmission $i \rightarrow j$ (or from node i to forwarding node j) is unsuccessful. Any packet retransmission through direct transmission $i \rightarrow j$ may still be unsuccessful if the channel experiences deep fading for a long period of time.

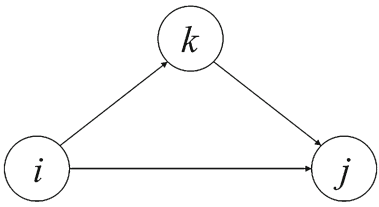

Fig. 2 Cooperative communications

Since node k overhears the packet, cooperative communication enables indirect transmission $i \rightarrow k \rightarrow j$. Since there may be a number of potential cooperative nodes $k \in K$, RL has been applied at node i to select a cooperative node [13], thereby providing spatial and time diversity gains.

A.3 *Routing* enables a sensor node to search for the best route to a sink node in clustered [14] and non-clustered networks [5]. Generally speaking, clustering segregates the entire network into groups with each consists of a clusterhead and member nodes. The clusterhead collects, processes and aggregates sensing outcomes received from member nodes, and subsequently send them to the sink node through single or multiple hops. RL has been applied in each sensor node to learn the best route to the sink node.

A.4 *Rate control* adjusts the packet transmission rate of a source node, and hence the congestion level of intermediate nodes, along a route [15,16].

A.5 *Sensing coverage* is a WSN application that maximizes the physical sensing coverage of an area so that any event of interest is accurately detected by at least a single sensor node. Sensing coverage can be applied in surveillance and monitoring tasks (e.g., intruder and fire detections). To reduce energy consumption, RL has been applied in each sensor node to minimize the overlapping of sensing coverages [17].

A.6 *Task scheduling* schedules and carries out the right task at different time instant. For instance, in [18], RL has been applied in each sensor node to learn the usefulness of each task (i.e. sensing, transmitting, receiving, aggregating data, and sleeping) at different time instant in order to reduce energy consumption.

2 Reinforcement learning: components, features and enhancements

This section presents the traditional and enhanced components and features of RL in the context of WSNs. For each component and feature, we show the traditional approach and subsequently the alternative or enhanced approaches.

2.1 State

Traditionally, each state is comprised of a single type of information. For instance, each state $s_t^i \in S = \{1, 2, \ldots, K\}$ represents the number of packets in the buffer queue [8]. The state representation can be enhanced in two ways. Firstly, the state may not be represented because there is a single state only, and this is called stateless

[7]. Secondly, each substate may be comprised of distinctive substates. For instance, state $\mathbf{s_t^i} = (s_{x,t}^i, s_{y,t}^i) \in S$, where $s_{x,t}^i \in S_x$ and $s_{y,t}^i \in S_y$ represent a set of potential neighboring nodes and data flows, respectively) [32].

The state representation can be further enhanced through minimization of Hamming distance for state space. In general, larger state space increases memory requirement, and reduces the convergence rate to optimal actions since an agent must explore more state-action pairs. The number of states can be reduced based on Hamming distance [12]. An agent calculates the weighted Hamming distance between two states, specifically $H(s_1 - s_2) = W_1 \cdot |V_1(s_1) - V_1(s_2)| + W_2 \cdot |V_2(s_1) - V_2(s_2)| + \cdots + W_N \cdot |V_N(s_1) - V_N(s_2)|$, where the weight W_n represents the significance of the corresponding variable $|V_n(s_1) - V_n(s_2)|$ in differentiating the two states. Both states s_1 and s_2 share a single entry in the Q-table if their Hamming distance is less than a threshold or $H(s_1 - s_2) < H_T$.

2.2 Action

Traditionally, each action represents a single action out of a set of possible actions. For instance, in a routing scheme A(3) [5,14], each action $a_t^i \in A = \{1, 2, \ldots, K\}$ represents a next-hop node for packet transmission, while A represents a set of all neighbor nodes. The action representation can be enhanced in two ways. Firstly, each action may be represented by subactions. For instance, in [7,13], action $\mathbf{a_t^i} = (a_{1,t}^i, a_{2,t}^i, \ldots, a_{K,t}^i) \in A_1 \times A_2 \times \cdots \times A_K$, where $a_{k,t}^i \in A_k = \{0, 1\}$. Secondly, each subaction may further be comprised of distinctive subactions. For instance, in [32], action $\mathbf{a_t^i} = (a_{x,t}^i, a_{y,t}^i) \in A$, where $a_{x,t}^i \in A_x = \{0, 1\}$ and $a_{y,t}^i \in A_y = \{a_{y,1,t}^i, a_{y,2,t}^i, \ldots, a_{y,K,t}^i\}$.

2.3 Delayed reward

Traditionally, each delayed reward represents the performance enhancement achieved by a state-action pair. A single reward computation approach is applicable to all state-action pairs. For example, in a sleep-wake scheduler A(1.1) [8], the delayed reward $r_{t+1}^i(a_{t+1}^i)$ is a ratio of the effective transmission and reception time durations to the waking time duration. Additionally, the delayed reward can be a constant value, such as $r_{t+1}^i(a_{t+1}^i) = 1$ or -1 to indicate successful and unsuccessful transmissions [7]. The delayed reward can be further enhanced in the context of WSNs as described next.

2.3.1 Distinctive reward functions

Different reward functions can be used to compute rewards under distinctive network conditions [5,10].

As an example, in a transceiver selector A(1.2) [10], the action is to select a transceiver and its transmission power level for packet transmissions. The cost (or negative reward) for each packet transmission depends on the amount of energy consumption, and it is a function of the number of retransmissions, MAC delays (i.e. channel sensing and backoff), transmission and reception power levels, as well as packet size. Higher

cost indicates higher energy consumption. However, there is a condition in which the reward computation is different. Specifically, if transmissions are unsuccessful even though the highest transmission power has been used, a zero reward value is assigned in order to avoid the agent from exploring other actions with lower transmission power levels until the transmissions are successful at the highest transmission power level.

2.3.2 Average delayed reward function

Traditionally, delayed reward is an instant value. The application of an average delayed reward has been shown to improve the overall system performance [19], and it has been applied in [16,20].

As an example, in [16], the average delayed reward is as follows:

$$r^i_{a,t+1}(s^i_{t+1}) \leftarrow r^i_{a,t}(s^i_{t+1}) + \alpha_r \Big[r^i_t(s^i_{t+1}) - r^i_{a,t}(s^i_{t+1}) + \max_{a \in A} Q^i_t(s^i_{t+1}, a) - \max_{a \in A} Q^i_t(s^i_t, a) \Big] \tag{4}$$

where $r^i_{a,t}(s^i_{t+1})$ represents the average delayed reward, and α_r represents the learning rate of the average delayed reward computation. The Q-function (1) is rewritten to incorporate the average delayed reward as follows:

$$Q^i_{t+1}(s^i_t, a^i_t) \leftarrow (1-\alpha) Q^i_t(s^i_t, a^i_t) + \alpha \left[r^i_{t+1}(s^i_{t+1}) - r^i_{a,t}(s^i_{t+1}) + \gamma \max_{a \in A} Q^i_t(s^i_{t+1}, a) \right] \tag{5}$$

In [16], the average delayed reward approach is applied in congestion avoidance A(4) to adjust the packet transmission rate of a source node in order to adjust the congestion level. The state represents the number of packets in the buffer queue; the action selects a next-hop node and a packet transmission rate; and the delayed reward is a function of energy efficiency and packet loss rate.

As another example, in [20], the average delayed reward is as follows:

$$r^i_{a,t+1}(s^i_{t+1}) \leftarrow (1-\alpha_r) \cdot r^i_{a,t}(s^i_{t+1}) + \alpha_r \cdot r^i_t(s^i_{t+1}) \tag{6}$$

The Q-function (1) is rewritten to incorporate the average delayed reward as follows:

$$Q^i_{t+1}(s^i_t, a^i_t) \leftarrow (1-\alpha) Q^i_t(s^i_t, a^i_t) + \alpha \left[r^i_{t+1}(s^i_{t+1}) - \gamma (r^i_{a,t}(s^i_{t+1}) + \max_{a \in A} Q^i_t(s^i_{t+1}, a)) \right] \tag{7}$$

In [20], the average delayed reward approach is applied in a sleep wake scheduler A(1.1) to adjust the waking time duration (or duty cycle), transmission power and modulation levels in order to reduce energy consumption. The state represents channel gain and the number of packets in the buffer queue; the action selects the waking time duration, as well as transmission power and modulation levels; and the reward is a

ratio of the number of received and transmitted packets to energy consumption and the processing cost in the buffer queue.

2.4 Discounted reward

Traditionally, the discounted reward has been applied to indicate the dependency of Q-function on future rewards. As an example, in a routing scheme A(3) [5], the delayed reward represents the link cost from a node to a next-hop node; while the discounted reward $\gamma \max_{a \in A} Q_t^i(s_{t+1}^i, a)$ represents the route cost from the next-hop node to a sink node, which may be multiple hops away. The discounted reward may be omitted with $\gamma = 0$ to show the lack of dependency on future rewards, and this approach is generally called the myopic approach which enables an agent to adapt to instantaneous changes in the operating environment [21]; and further discussion on this approach is presented in Sect. 3.1. The discounted reward can be further enhanced using average discounted reward function. Generally speaking, the future reward may be uncertain in some cases, and so an agent may be uncertain about its action selection. In [22], the average discounted Q-value is computed for all possible actions; and hence $\max_{a \in A} Q_t^i(s_{t+1}^i, a)$ in Eq. (1) is replaced by $\sum_{a \in A}[P(a) \times Q_t^i(s_{t+1}^i, a)] / \sum_{a \in A} P(a)$. Note that, if all possible actions are taken into account, then $\sum_{a \in A} P(a) = 1$.

2.5 Q-function

The traditional Q-function (see Eq. (1)) can be further enhanced.

2.5.1 Q-value Initialization

Generally speaking, the Q-values are initialized to a certain value (e.g. a zero value) so that all possible actions are given a fair chance during exploration. However, it can be initialized with different values to speed up rate. For instance, in a cooperative communication scheme A(2) [13], the Q-values are initialized based on the distance between a node i and its next-hop node j in which higher Q-values indicate more favorable nodes in making the progress in terms of the physical distance towards a sink node in order to reduce end-to-end delay.

2.5.2 Reward equivalent Q-function

In [23,24], the learning rate and discount factor are set to $\alpha = 1$ and $\gamma = 0$, so the Q-function equals delayed reward $Q_{t+1}^i(a_t^i) = r_{t+1}^i(a_t^i)$, and it is applied to speed up the learning process in a routing scheme A(3). In [23], a node i selects its next-hop node a_t^i and updates its Q-value $Q_{t+1}^i(a_t^i) = r_{t+1}^i(a_t^i) = c_{a_t^i} + \min_a Q_t^{a_t^i}(a)$, where $c_{a_t^i}$ represents the link cost between node i and its next-hop node a_t^i, and $\min_a Q_t^{a_t^i}(a)$ indicates that node a_t^i chooses its next-hop node with the minimum Q-value. Note that, nodes must exchange Q-values, which indicates the route cost to a destination sink node, among themselves.

2.6 Exploration and exploitation

Traditionally, there are two popular approaches to achieve a balanced trade-off between exploration and exploitation, namely softmax and ε-greedy [2] which have been applied in [4,5,8,14,23,25], respectively. For instance, in [8], an agent chooses exploration actions with a small probability ε and exploitation actions with a probability $1 - \varepsilon$. In [21], during initial exploration, an agent explores all the available actions in a round-robin manner in order to discover the Q-values of all actions [21]. The exploration and exploitation mechanism can be further enhanced through adjusting the exploration probability.

The exploration probability may be adjusted based on the uncertain and dynamic levels of the operating environment due to nodal mobility and varying channel conditions. As an example, in [26], using the ε-greedy approach, node i adjusts its exploration probability $\varepsilon_t^i = n_{a+d,T}^i / n_T^i$, where $n_{a+d,T}^i$ represents the number of nodes that appear and disappear in node i's transmission range within a time window T, and n_T^i represents the number of node i's neighboring nodes. As another example, in [12], node i adjusts its exploration probability $\varepsilon_t^i = \varepsilon_{min} + \max[0, (\varepsilon_{max} - \varepsilon_{min}) \times (e_{max} - e)/e_{max}]$, where e represents the number of events of interest with lower e value increases the exploration probability ε_t^i.

The exploration probability may also be adjusted based on action selection. In [24], using the ε-greedy approach, node i adjusts its exploration probability as follows:

$$\varepsilon_{t+1}^i = \begin{cases} \varepsilon_t^i + \varepsilon_{step}, & if\ a_t^i \neq a_{t-1}^i \\ \varepsilon_t^i - \varepsilon_{step}, & \text{otherwise} \end{cases} \tag{8}$$

Note that, $\varepsilon_{t+1}^i = \varepsilon_t^i + \varepsilon_{step}$ helps to discover the optimal actions when the operating environment becomes unstable (i.e. when the consecutive actions change, or $a_t^i \neq a_{t-1}^i$), while $\varepsilon_t^i = \varepsilon_t^i - \varepsilon_{step}$ helps to achieve the optimal actions when the operating environment becomes stable.

3 Reinforcement learning: algorithms

The traditional RL approach (see Sect. 1.1) has been applies in various schemes to provide performance enhancement in WSNs as shown in Table 1.

A major contribution of this section is the discussion on a number of new additions and enhancements to the traditional RL algorithms, which have been applied to various schemes in WSNs. A summary of the new RL models and algorithms is shown in Table 2. The following subsections describe the model and algorithm, including the purpose(s) of the scheme(s), followed by its associated RL model (i.e. state, action and reward representations), and finally the algorithm.

3.1 Algorithm 1: myopic RL model with $\gamma = 0$

The myopic RL model sets the discount factor to zero value or $\gamma = 0$, so that there is lack of dependency on future rewards, and it has been applied in MAC protocols A(1)

Table 1 RL models with direct application of the traditional RL approach for various schemes in WSNs

References	Purpose	State	Action	Reward/cost
A.1 MAC				
Liu and Elhanany [8], A(1.1)	Each receiver (agent) selects the waking time duration (or duty cycle) as a function of the transmitter and its own traffic load. The purpose is to achieve a balanced trade-off between throughput P(1) and energy consumption P(3)	Number of packets for transmission	Selecting a duty cycle	Reward is a ratio of the effective transmission and reception time durations to the waking time duration
Hsu et al. [11], A(1.1)	Each sensor node (agent) selects the waking time duration (or duty cycle) as a function of QoS requirements and residual energy levels. Note that, energy harvesting is possible. The purpose is to fulfill the QoS requirements and to reduce energy consumption P(3)	• QoS requirements • Harvested energy • Residual energy • Difference between the harvested and consumed energy levels	Selecting a duty cycle	Reward is a positive constant value if • QoS requirements are fulfilled, or • Harvested energy level is higher than energy consumption level otherwise, the reward is a negative constant value
Shah et al. [12], A(1.1)	Each sensor node (agent) selects the waking time duration (or duty cycle). The purpose is to wake up at the right time whenever it is in contact with a mobile data collector node so that its sensing data can be sent to the collector node in order to increase in-contact time P(6) and to reduce energy consumption P(3)	• In-contact time duration • Boolean value indicating the presence or absence of a mobile data • Specific time when the state is evaluated	Selecting a duty cycle	Reward is a function of • Number of the occurrence of in contact with a mobile data collector node • Energy consumption level

Table 1 continued

References	Purpose	State	Action	Reward/cost
Mao et al. [20], A(1.1)	Each receiver (agent) selects the waking time duration (or duty cycle) as a function of the transmitter traffic load. The purpose is to achieve a balanced trade-off between packet latency P(2) and energy consumption P(3)	Number of packets for transmission at the transmitter of the agent	Selecting a duty cycle	Cost (or negative reward) is a function of • Penalty factor of energy consumption • Penalty factor of packet latency • Number of packets in the buffer queue
Gummeson et al. [10], A(1.2)	Each receiver (agent) selects either a long-range or short-range transceiver for data and control packet transmissions as a function of the number of packet retransmissions. The purpose is to reduce energy consumption P(3)	Transmission power level for each transceiver: • Current power • Higher power • Lower power	Selecting a transceiver and its transmission power level for data and control packet transmissions	Reward is dependent on the amount of energy consumption associated with each packet transmission. For each packet transmission, the reward is a function of • Number of packet retransmissions • MAC delays (i.e. channel sensing, backoff) • Transmission and reception power levels • Packet size
A.3 routing				
Dong et al. [5]	Each sender (agent) selects a next-hop node. The purpose is to make the most progress in terms of the physical distance towards a destination sink node in order to increase throughput P(1), as well as to reduce end-to-end delay P(2), and energy consumption (or to increase network lifetime) P(3)	Destination sink node in the network	Selecting a next-hop node	Reward is a function of • Progress in terms of the physical distance towards the destination sink node • Residual energy of the next-hop node

Table 1 continued

References	Purpose	State	Action	Reward/cost
Liang et al. [27]	Each sender (agent) selects a next-hop node. The purpose is to make the most progress in terms of the physical distance towards a destination sink node in order to increase throughput P(1), as well as to reduce end-to-end delay P(2)	Destination sink node in the network	Selecting a next-hop node	Reward is a ratio of the progress in terms of the physical distance towards the destination sink node to packet latency
Naputta and Usaha [28]	Each sender (agent) selects a next-hop node in the presence of malicious nodes that may drop forwarding packets in mobile networks. The purpose is to transmit packets towards a destination sink node in a reliable and efficient manner in order to increase route discovery rate P(5), as well as to reduce end-to-end delay P(2)	Destination sink node in the network	Selecting a next-hop node	Reward is a function of • A ratio of the progress in terms of the physical distance towards the destination sink node to packet latency • Trust value, which is a ratio of successful to total packet transmissions
Villaverde et al. [24]	Each sender (agent) selects a route. The purpose is to select a route with high route reliability and residual energy level in order to increase throughput P(1), as well as to reduce end-to-end delay P(2), and energy consumption (or network lifetime) P(3)		Selecting a route	Reward is a function of • Route reliability, which is dependent on packet error rate • Residual energy

Table 2 Summary of RL models and algorithms for various schemes in WSNs

Model	Purpose	References
Myopic RL model with $\gamma = 0$	This model sets the discount factor to zero value or γ=0, so that there is lack of dependency on future rewards	Chu et al. [7,29] Mihaylov et al. [30] Forster and Murphy [14]
RL model with continuous space representation	This model represents an action set with a continuous action space in order to address the curse of dimensionality	Niu and Deng [31]
RL model with directed exploration	This model enables an agent to explore actions in a guided manner using domain-specific knowledge (e.g. rewards) in order to improve the convergence rate to the optimal action	Alberola and Pesch [21]
Cooperative RL model	This model enables agents to observe the local operating environment, and subsequently make their respective local action selections as part of the optimal joint action for network-wide performance enhancement	Liang et al. [13,26,32,33] Maalej et al. [34] Tham and Renaud [17] Seah et al. [4] Renaud and Tham [35]
Model-based RL model	This model estimates the state transition probability matrix **T**, which forms the model and represents the operating environment, and subsequently updates the Q-values using **T** in order to increase the convergence speed	Hu and Fei [36]
Hierarchical RL model	This model segregates the entire system into upper and lower levels, and applies two separate RL approaches in each level to achieve global and local optimal actions, respectively	Hu and Fei [25]

Table 3 Myopic RL algorithm with discount factor $\gamma = 0$

Repeat

(a) Choose action $a_t^i \in A$ based on state s_t^i

(b) Observe state s_{t+1}^i and reward $r_{t+1}^i(s_{t+1}^i)$

(c) Update Q-value:

$$Q_{t+1}^i(s_t^i, a_t^i) \leftarrow (1-\alpha) Q_t^i(s_t^i, a_t^i) + \alpha \cdot r_{t+1}^i(s_{t+1}^i) \tag{9}$$

[7,29,30] and clustering A(3) [14]. Table 3 presents the RL algorithm for the myopic RL model.

3.1.1 Chu's slot assignment scheme for MAC protocol

Chu et al. [7] propose a slot assignment scheme for MAC protocol A(1) using the myopic RL model (see Table 3), and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2) and energy consumption P(3). The purpose

Table 4 RL model for Chu's slot assignment scheme [7]

Action	$\mathbf{a_t^i} = (a^i_{1,t}, a^i_{2,t}, \ldots, a^i_{K,t}) \in A_1 \times A_2 \times \cdots \times A_K$, each subaction $a^i_{k,t} \in A_k = \{0, 1\}$ represents the selection of time slot k. Specifically, $a^i_{k,t} = 1$ if time slot k is selected, and vice-versa. K represents the number of time slots in a time window
Reward	$r^i_{t+1}(a^i_{k,t}) = \begin{cases} 1, & \text{if successful transmission} \\ -1, & \text{if unsuccessful transmission} \end{cases}$

Table 5 RL model for Forster's intra-cluster routing scheme [14]

Action	$a^i_t \in A = \{1, 2, \ldots, J\}$, each action a^i_t represents a next-hop neighbor node j. J represents the number of node i's neighbor nodes
Reward	$r^i_{t+1}(a^i_t) = 1$, where $r^i_{t+1}(a^i_t)$ represents the link cost to the next-hop neighbor node

is to select a time slot within a time frame for data transmission in order to minimize collisions in a time-slotted MAC protocol.

Table 4 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the possibility of successful data transmission in each time slot using Q-value $Q^i_{t+1}(a^i_{k,t})$. The state is not represented. The action $\mathbf{a_t^i}$ is to select time slot(s) for data transmission. The reward $r^i_{t+1}(a^i_{k,t})$ indicates successful and unsuccessful transmissions in time slot k, respectively. The Q-function (1) is rewritten as $Q^i_{t+1}(a^i_{k,t}) \leftarrow (1-\alpha) Q^i_t(a^i_{k,t}) + \alpha \cdot r^i_{t+1}(a^i_{k,t})$ since the state is not represented; and time slots with higher Q-value indicate higher possibility of successful transmission, so these slots are selected for transmission. Similar RL model has also been applied in (Mihaylov et al. 2012) [30].

3.1.2 Forster's intra-cluster routing scheme

Forster and Murphy [14] propose an intra-cluster routing scheme for clustered networks A(3) using the myopic RL model (see Table 3), and it has been shown to increase throughput P(1), as well as to reduce energy consumption P(3). The purpose is to enable a member node to select a next-hop neighbor node, which provides a route with lower number of hops and higher residual energy towards the clusterhead in a multi-hop cluster. The proposed scheme helps to achieve a balanced energy consumption among member nodes in a cluster in order to prolong network lifetime.

Table 5 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the cost of a route leading to the clusterhead node using Q-value $Q^i_{t+1}(a^i_t)$. The state is not represented. The action a^i_t represents the selection of a next-hop node j to forward packets to clusterhead. The reward $r^i_{t+1}(a^i_t)$ represents the link cost to the next-hop node.

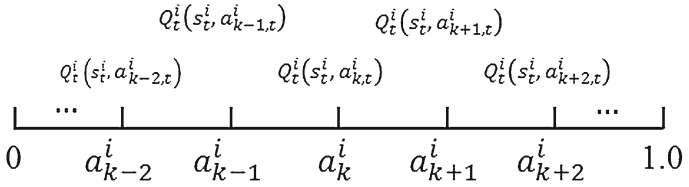

Fig. 3 An interval of (0,1) is partitioned into discrete actions

Table 6 RL algorithm with continuous action space representation

Repeat

(a) Choose action a_t^i:

$$a_t^i = \frac{a_{k,t}^i Q_t^i\left(s_t^i, a_{k,t}^i\right) + a_{k+1,t}^i Q_t^i\left(s_t^i, a_{k+1,t}^i\right)}{Q_t^i\left(s_t^i, a_{k,t}^i\right) + Q_t^i\left(s_t^i, a_{k+1,t}^i\right)}$$

Determine Q-value $Q_t^i\left(s_t^i, a_t^i\right)$ for a_t^i:

$$Q_t^i\left(s_t^i, a_t^i\right) = Q_t^i\left(s_t^i, a_{k,t}^i\right) + \frac{Q_t^i\left(s_t^i, a_{k+1,t}^i\right) - Q_t^i\left(s_t^i, a_{k,t}^i\right)}{a_{k+1,t}^i - a_{k,t}^i} \left(a_t^i - a_{k,t}^i\right)$$

(b) Observe state s_{t+1}^i and reward $r_{t+1}^i\left(s_{t+1}^i\right)$

(c) Update Q-value $Q_t^i\left(s_t^i, a_t^i\right)$ for action a_t^i using Eq. (1)

(d) Update Q-values $Q_t^i\left(s_t^i, a_{k,t}^i\right)$ and $Q_t^i\left(s_t^i, a_{k+1,t}^i\right)$ for actions $a_{k,t}^i$ and $a_{k+1,t}^i$:

$$Q_{t+1}^i\left(s_t^i, a_{k,t}^i\right) = \frac{a_{k+1,t}^i - a_t^i}{a_{k+1,t}^i - a_{k,t}^i} Q_t^i\left(s_t^i, a_t^i\right)$$

$$Q_{t+1}^i\left(s_t^i, a_{k+1,t}^i\right) = \frac{a_t^i - a_{k,t}^i}{a_{k+1,t}^i - a_{k,t}^i} Q_t^i\left(s_t^i, a_t^i\right)$$

3.2 Algorithm 2: RL model with continuous space representation

RL suffers from the curse of dimensionality in which the accuracy of state and action space representations increases with smaller sizes of space partitions, respectively. However, memory capacity is limited at each sensor node, so RL model with continuous action space is applied to address this. An example of the Q-values of actions for a particular state is shown in Fig. 3, in which an interval of (0,1) is partitioned into a set of discrete actions and their corresponding Q-values. In this RL model, a continuous action is computed based on the average of two adjacent discrete actions weighted by their respective Q-values [31].

Table 6 presents the RL algorithm with continuous action space representation. In step (a), with reference to Fig. 3, the continuous action a_t^i is chosen based on state s_t^i by averaging the discrete actions $a_{k,t}^i$ and $a_{k+1,t}^i$ weighted by their respective Q-values $Q_t^i(s_t^i, a_{k,t}^i)$ and $Q_t^i(s_t^i, a_{k+1,t}^i)$. Also, in step (d), the updates of the Q-values $Q_t^i(s_t^i, a_{k,t}^i)$ and $Q_t^i(s_t^i, a_{k+1,t}^i)$ are weighted by the linear distance between $a_{k,t}^i, a_{k+1,t}^i$ and a_t^i.

Table 7 RL model for Niu's sleep-wake scheduling scheme [31]

State	$s_t^i \in S = \{0, 1, 2\}$ represents the changing trend of the state. State $s_t^i = 0$ represents an empty buffer queue; $s_t^i = 1$ and $s_t^i = 2$ represent a decreasing and an increasing buffer queue length, respectively
Action	a_t^i represents the selection of the probability of transmission
Reward	$r_{t+1}^i(a_t^i) = w_1 \cdot e_t^i + w_2 \cdot q_t^i + w_3$ where w_1, w_2 and w_3 represents weights; e_t represents energy consumption; and q_t represents the number of packets in the buffer queue. $w_3 > 0$ is a positive constant

3.2.1 Niu's sleep-wake scheduling scheme for MAC protocol

Niu and Deng [31] propose a sleep-wake scheduling scheme A(1.1) for MAC protocol using the RL model with continuous action space representation (see Table 6), and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2) and energy consumption P(3). The purpose is to select the probability of transmission during the data transmission stage in a time-slotted MAC protocol.

Table 7 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the probability of transmission during the data transmission stage using Q-value $Q_{t+1}^i(s_t^i, a_t^i)$. The state represents the changing trend of the number of packets in the buffer queue. The continuous action a_t^i is to select the probability of transmission during the data transmission stage. The reward $r_{t+1}^i(a_t^i)$ depends on energy consumption levels and the number of packets in the buffer queue (and hence packet latency), and so the scheme aims to achieve a balanced performance in energy consumption and packet latency.

3.3 Algorithm 3: RL model with directed exploration

Traditionally, an agent adopts an undirected exploration approach (e.g. ε-greedy), in which the agent explores actions in a random manner during exploration. An enhanced approach called directed exploration enables an agent to explore actions in a guided manner using domain-specific knowledge (e.g. rewards) or rules in order to improve the convergence rate to the optimal action [21]. For instance, in [21], the exploration probability is adjusted according to two conditions in regards to the variations of rewards caused by uncertainties in the operating environment. Firstly, the learning speed increases with the variations of rewards. Secondly, an agent exploits at all times until there are variations in the rewards, which initiate the exploration procedure.

3.3.1 Alberola's sleep-wake scheduling scheme for MAC protocol

Alberola and Pesch [21] propose a sleep-wake scheduling scheme A(1.1) for MAC protocol using the RL model with directed exploration, and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2) and energy consumption P(3). The purpose is to achieve a balanced tradeoff between the sleeping

Table 8 RL model for Alberola's sleep-wake scheduling scheme [21]

Action	$a_t^i \in A$ represents the selection of the number of time slots for sleeping within a time frame
Cost	$r_{t+1}^i(a_t^i) = \begin{cases} 0, & \text{if } q_t^i < q_{Th}, t_{IL}^i = 0 \\ -1, & \text{if } q_t^i > q_{Th} \\ -t_{IL}^i, & \text{otherwise} \end{cases}$

time and the waking or active time (or duty cycle) in a time-slotted MAC protocol in order to reduce energy consumption.

Table 8 shows the RL model for the scheme, and it is embedded in a centralized node that collects data from sensor nodes in a single hop to determine the optimal active time duration using Q-value $Q_{t+1}^i(s_t^i, a_t^i)$. The centralized node collects network statistics during active time duration to estimate the incoming traffic level from neighboring nodes, and subsequently learn the optimal duty cycle. The state is not represented. The action a_t^i is to select the number of time slots for sleeping within a time frame. Higher a_t^i value indicates lower duty cycles and so there is lower energy consumption. Denote the idle time duration by t_{IL}^i, as well as the number of packets in the buffer queue and its threshold by q_t^i and q_{Th}, respectively. The cost is $r_{t+1}^i(a_t^i) = 0$ if the active duration is fully utilized with idle time $t_{IL}^i = 0$, and there is no buffer overflow (or $q_t^i < q_{Th}$). The cost is $r_{t+1}^i(a_t^i) = -1$ if there is buffer overflow, which indicates that the active duration is too short or insufficient. The cost is $r_{t+1}^i(a_t^i) = -t_{IL}^i$ if the active time duration is too long resulting in energy wastage. The myopic RL algorithm in Table 3 (see Sect. 3.1) is applied to update the Q-values.

In [21], the RL model with directed exploration specifies a set of strategies, namely random, round-robin and greedy strategies. The random strategy is applied during exploration, and it is as follows:

$$\pi_{1,i}(a) = \begin{cases} \text{rand}_{a<a_{t-1}^i}(a), & \text{if } r_t^i(a_t^i) > r_{t-1}^i(a_{t-1}^i) \\ \text{rand}_{a\geq a_{t-1}^i}(a), & \text{otherwise} \end{cases}$$

In this strategy, $\text{rand}_{a<a_{t-1}^i}(a)$ chooses a lower number of time slots for sleeping a_t^i in a random manner in order to reduce the inactive time duration if the reward has increased at time t, or $r_t^i(a_t^i) > r_{t-1}^i(a_{t-1}^i)$, which indicates that the incoming traffic level has increased.

The round-robin strategy is applied during exploitation whenever there are extreme cases in which the centralized node either has buffer overflow or is always in idle listening $t_{IL}^i = 1$, and the rule is as follows:

$$\pi_{2,i}(a) = \begin{cases} a_{t-1}^i - 1, & \text{if } q_t^i > q_{Th} \\ a_{t-1}^i + 1, & \text{if } t_{IL}^i = 1 \end{cases}$$

In this strategy, $a_{t-1}^i - 1$ reduces its sleeping time slots by 1 if $q_t^i > q_{Th}$; while $a_{t-1}^i + 1$ increases its sleeping time slots by 1 if $t_{IL}^i = 1$. The exploration probability is increased by $\varepsilon_t^i = \varepsilon_{max}/10$ to increase exploration.

The greedy strategy is applied during exploitation in cases other than the two aforementioned extreme cases.

$$\pi_{3,i}(a) = \begin{cases} \text{argmax}_{a<a^i_{t-1}} Q^i_t(a), & \text{if } r^i_t(a^i_t) > r^i_{t-1}(a^i_{t-1}) \\ \text{argmax}_{a\geq a^i_{t-1}} Q^i_t(a), & \text{otherwise} \end{cases}$$

In this strategy, $\text{argmax}_{a<a^i_{t-1}} Q^i_t(a)$ chooses a lower number of time slots for sleeping a^i_t with the maximum Q-value $Q^i_t(a^i_t)$ in order to reduce sleeping time duration if the reward has increased at time t, or $r^i_t(a^i_t) > r^i_{t-1}(a^i_{t-1})$. The learning rate is increased by $\alpha^i_t = \alpha_{max}/10$ to speed up learning; however, the learning rate and exploration probability are decreased whenever the current and previous actions are similar, or $a^i_t = a^i_{t-1}$, to avoid oscillations in action selection.

3.4 Algorithm 4: cooperative RL model

Traditionally, an agent makes decisions on action selection, which may be locally optimal, in an independent manner without communicating with neighboring agents. In order to make decisions on globally optimal action selection, the cooperative RL approach enables agents to observe the local operating environment, exchange information (e.g. states and Q-values) among themselves, and subsequently select local actions as part of the optimal joint action for network-wide performance enhancement. Hence, each local action can affect and can be affected by other agents. This cooperative approach is suitable for schemes that require collaborative efforts in a shared wireless medium. For instance, in routing, nodes along a route, as well as their respective neighboring nodes, must collaborate to reduce interference for end-to-end QoS enhancement. Section 3.4.1 presents cooperative RL algorithms. Section 3.4.2 presents application schemes that apply the cooperative RL algorithms.

3.4.1 Cooperative RL algorithms

This section presents two cooperative RL algorithms applied to WSNs.

Liang's cooperative function [13] has been applied in cooperative communication scheme A(2); and the discussion of this algorithm is based on this application. Denote the current-hop cooperative nodes and next-hop nodes as H_n and H_{n+1} respectively, which can be viewed as the current and neighboring groups of agents. Node $i \in H_n$ forwards a packet to node $j \in H_{n+1}$, and keeps track of its Q-value $Q^i_{t+1}(s^i_t, a^i_t)$ as follows:

$$\begin{aligned} Q^i_{t+1}(s^i_t, a^i_t) \leftarrow\ & (1-\alpha) Q^i_t(s^i_t, a^i_t) \\ & + \alpha \big[r^i_{t+1}(s^i_{t+1}) + \gamma\, w(i,j) \max_{a_j \in H_{n+1}} (Q^j_t(s^j_t, a_j)) \\ & + \gamma \sum\nolimits_{i' \in H_n, i' \neq i} w(i,i') \max_{a_{i'} \in H_{n\setminus i}} (Q^{i'}_t(s^{i'}_t, a_{i'})) \big] \end{aligned} \tag{10}$$

where $w(i, j)$ represents the weight of node i on node j's Q-value, in which higher $w(i, j)$ indicates greater effects; while $w(i, i^{'})$ represents the weight of node i on node i's cooperative nodes in H_n. Note that, with respect to Eq. (10), the third term depends on the maximum Q-value of node $j \in H_{n+1}$; and the fourth term depends on the maximum Q-values of all nodes in H_n except node i itself. Hence, nodes exchange and apply information (i.e. Q-values) with neighboring nodes, and maximize their own and their respective neighboring nodes' Q-values in order to maximize the global Q-value. At the next time instant, node $i \in H_n$ with the highest Q-value is selected as the forwarding node; while the rest of the nodes $i^{'} \in H_n \backslash i$ become cooperative nodes.

Distributed value function (DVF) approach has been applied in task scheduling A(6) [18]. Node i calculates and exchanges local value function $V^i(s_t^i)$ (see Eq. (2)) with its neighbor nodes $j \in J$, and keeps track of its Q-value $Q_{t+1}^i(s_t^i, a_t^i)$ as follows:

$$Q_{t+1}^i(s_t^i, a_t^i) \leftarrow (1-\alpha) Q_t^i(s_t^i, a_t^i) + \alpha \left[r_{t+1}^i(s_{t+1}^i) + \gamma \sum_{j \in J} w(i, j) V^j(s_{t+1}^i) \right] \tag{11}$$

where $w(i, j)$ represents the weight of node i on neighbor node $j \in J$'s Q-value. For instance, in [17], the weights for all neighbor node j's Q-values are equal, specifically $w(i, j) = 1/|J|$.

3.4.2 Application schemes with cooperative RL algorithms

This section presents three schemes that apply the cooperative RL model.

Liang's cooperative communication scheme

Liang et al. [13] propose a cooperative communication scheme A(2) using Liang's cooperative function (see Sect. 3.4.1.1), and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2). The purpose is to select a forwarding node $j \in H_{n+1}$ for data transmission from node i in order to minimize packet loss as shown in Fig. 4, where $K = |H_n|$ is the number of a set of cooperative nodes at the current hop H_n. Generally speaking, cooperative nodes that form the set K can hear two-way routing messages (i.e. Route Request, RREQ and Route Reply, RREP) between nodes i and j [32]. A packet may pass through many sets of forwarding and cooperative nodes as it traverses across the network from a sensor node to a sink node.

Table 9 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the progress of a packet in terms of the physical distance towards the sink node over time using Q-value $Q_{t+1}^i(a_{k,t}^i)$. Specifically, it is the progress being made from the current hop H_n to the next hop H_{n+1}. The state represents the hop of a packet with respect to the current hop H_n in which node i resides. The action $\mathbf{a_t^i}$ is to select a forwarding node to forward the data packet. All nodes in the current hop H_n receive positive rewards and update their Q-values accordingly when the nodes hear further transmission from H_{n+1} to H_{n+2}, which indicates a successful transmission from H_n to H_{n+1}, otherwise the nodes receive negative rewards. Higher

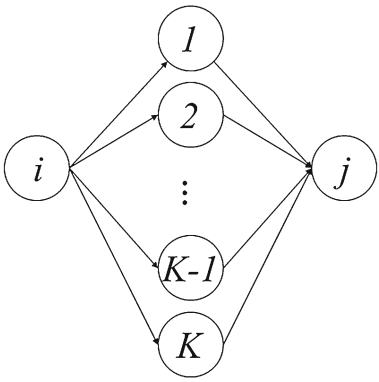

Fig. 4 Node i selects node $j \in H_{n+1}$ as forwarding node for data transmission. Nodes in H_n, where $K = |H_n|$, become cooperative nodes in the current hop H_n

Table 9 RL model for Liang's cooperative communication scheme [13]

State	$s_t^i \in S = \{H_{n-1}, H_n, H_{n+1}\}$ represents the hop of a packet with respect to the current hop H_n in which node i resides. State $s_t^i = H_{n-1}$ represents the previous hop of H_n, $s_t^i = H_n$ represents the current hop; and $s_t^i = H_{n+1}$ represents the next hop of H_n
Action	$\mathbf{a_t^i} = (a_{1,t}^i, a_{2,t}^i, \ldots, a_{K,t}^i) \in A_1 \times A_2 \times \cdots \times A_K$, each subaction $a_{k,t}^i \in A_k = \{0, 1\}$ represents the selection of node k as the forwarding node. Specifically, $a_{k,t}^i = 1$ if node k is selected, and vice-versa
Reward	$r_{t+1}^i(s_t^i, \mathbf{a_t^i}) = \begin{cases} Amount\ of\ progress\ to\ sink, & \text{if successful transmission} \\ -Amount\ of\ timeout\ duration, & \text{if unsuccessful transmission} \end{cases}$

positive rewards indicate higher transmission quality in which the packet has made greater progress in terms of the physical distance towards the sink node over time; while higher negative rewards indicate the amount of timeout duration being wasted as a result of unsuccessful packet transmission. Both positive and negative rewards are normalized values. All nodes in the current hop H_n update with the similar positive (or negative) rewards because all of them have made the correct (or incorrect) action in the selection of a forwarding node. Node $i \in H_n$ forwards a packet to node $j \in H_{n+1}$, and keeps track of its Q-value $Q_{t+1}^i(s_t^i, a_t^i)$ using Eq. (10).

Similar RL model and algorithm have been applied to achieve other performance enhancements in cooperative communications [32–34] and routing [26]. The RL model is redefined in [32,33]. As an example, with respect to node i in a cooperative communication scheme A(2) [33], the state represents QoS satisfaction/violation levels; the action represents the selection of a cooperative node; and the reward represents the improvement on packet delivery rate and packet latency brought about by indirect transmission. As another example, with respect to node i in a routing scheme A(3) [26], the state represents a set of neighboring node and QoS requirements of packets; the action represents the selection of a next-hop node; and the reward represents the inverse of packet latency, and so higher rewards indicates shorter single-hop delay.

Next, Sect. 3.4.2.2 presents the extension of the RL model [32] presented in this section [13,33] to reduce energy consumption and enhance QoS.

Liang's cooperative communication scheme with QoS enhancement

Table 10 RL model for Liang's cooperative communication scheme with QoS enhancement [32]

State	$\mathbf{s_t^i} = \left(K^i, F^i\right) \in S$, where $k^i \in K^i$ is a set of potential forwarding and cooperative nodes at current hop H_n, and $f^i \in F^i$ is a set of data flow at node i
Action	$\mathbf{a_t^i} = (a^i_{f^i,t}, P^i_{f^i,t}) \in A$, each subaction $a^i_{f^i,t} \in A_{f^i} = \{0, 1\}$ represents whether to forward data flow f^i; specifically, $a^i_{f^i,t} = 1$ if node i chooses to forward packets from data flow f^i, and vice-versa; and each subaction $P^i_{f^i,t} \in A_{P^i} = \{P_0, P_1, \ldots, P_{\max}\}$ represents the selection of a transmission power level
Reward	$r^i_{t+1}(\mathbf{s_t^i}, \mathbf{a_t^i}) = w_1 \cdot c^i_t + w_2 \cdot d^i_t + w_3 \cdot e^i_t$, where c^i_t, d^i_t and e^i_t represent improvement on packet delivery rate, packet latency and energy efficiency in indirect transmission compared to those in direct transmission, respectively; and w_1, w_2, and w_3 are weight factors

Liang et al. [32] propose a cooperative communication scheme A(2) using Liang's cooperative function (see Sect. 3.4.1.1) to provide QoS enhancement, and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2) and energy consumption P(3). The purpose is to select a forwarding node $j \in H_{n+1}$ for data transmission from node i, and to adjust its transmission power level, in order to reduce energy consumption and enhance QoS.

Generally speaking, to provide QoS enhancement, Liang et al. [32] use the same RL algorithm in Sect. 3.4.2.1, and the difference is that, the RL model is redefined. Table 10 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the contribution, which is based on the successful packet transmission, packet latency, and energy efficiency, of indirect transmission compared to direct transmission, using Q-value $Q^i_{t+1}(a^i_{k,t})$. The state $\mathbf{s_t^i}$ represents a set of potential forwarding and cooperative nodes in the current hop H_n, and a set of data flow at node i. The action $\mathbf{a_t^i}$ is to select whether to forward packets of a data flow and the transmission power level; hence, a node may transmit packets using an appropriate transmission power level adaptively according to the channel condition. The reward represents the improvement on packet delivery rate, packet latency, and energy efficiency brought about by indirect transmission; and this information is indicated in the acknowledgement (ACK) packets sent by the forwarding node $j \in H_{n+1}$ to node i.

Tham's sensing coverage scheme Seah et al. [4], Tham and Renaud [17], and Renaud and Tham [35] propose a sensing coverage scheme A(5) using DVF (see Sect. 3.4.1.2), and it has been shown to increase sensing coverage P(4), as well as to reduce energy consumption P(3). The purpose is to select a sensing coverage level for each sensor node i in monitoring tasks in order to minimize overlapping of sensing coverage with neighbor node $j \in J$.

Table 11 shows the RL model for the scheme, and it is embedded in each sensor node to keep track of the coverage of each grid point in the surrounding area of the sensor node using Q-value $Q^i_{t+1}(s^i_t, a^i_t)$ [4]. The state s^i_t represents the coverage of a grid point. The action a^i_t is to select an action whether to hibernate (inactive) or sense (active). The reward $r^i_{t+1}(s^i_t, a^i_t)$ represents a ratio of the gain received for each grid point being covered to the state of the respective grid point. Higher positive

Table 11 RL model for Tham's sensing coverage scheme [4]

State	$s_t^i \in S = \{s_0, s_1, s_2\}$ represents the coverage of a grid point. State $s_t^i = s_0, s_1, s_2$ represents a grid point is not covered, covered by a single sensor, and covered by more than a sensor node, respectively
Action	$a_t^i \in A = \{a_0, a_1\}$ represents whether to hibernate (inactive) or sense (active), respectively
Reward	$r_{t+1}^i(s_t^i, a_t^i)$ is a ratio of the gain received for each grid point being covered to the state of the respective grid point (i.e. 1 for s_1, and 2 for s_2) if $s_t^i = s_1$ or $s_t^i = s_2$; otherwise $r_{t+1}^i(s_t^i, a_t^i) = 0$ if $s_t^i = s_0$

Table 12 RL model for Tham's sensing coverage scheme [35]

State	$s_t^i \in S = \{s_0, s_1\}$ represents the coverage of a grid point. State $s_t^i = s_0, s_1$ represents a grid point is not covered, and covered by at least a single sensor node, respectively
Action	$a_t^i \in A = \{a_0, a_1, a_2\}$. Action $a_t^i = a_0, a_1, a_2$ represent whether to hibernate (inactive), sense (active) with a short-range coverage, or sense with a long-range coverage, respectively
Reward	$r_{t+1}^i(s_t^i) = G_{t+1}^i(s_t^i) - C_t^i(a_t^i)$, where $G_{t+1}^i(s_t^i)$ represents the gain achieved by the current sensing coverage, and $C_t^i(a_t^i)$ represents energy consumption associated with action a_t^i

rewards indicate higher gain and the grid point is covered by a single sensor node only; and hence, lower energy consumption. Each sensor node keeps track of the Q-value $Q_{t+1}^i(s_t^i, a_t^i)$ using Eq. (11).

Similar RL model and algorithm have been applied to achieve the same purpose in [35], in which there are two levels of sensing range. Generally speaking, the RL model is redefined in [35], as shown in Table 12. The state s_t^i represents the coverage of a grid point. The action a_t^i is to select whether to hibernate (inactive), or sense (active) with a short-range, or a long-range coverage. The reward $r_{t+1}^i(s_t^i)$ represents the gain received for each grid point being covered minus the cost associated with energy consumption. Long-range sensing incurs higher cost. Higher positive rewards indicate higher gain and the grid point is covered by lesser number of sensor nodes; and hence, lower energy consumption.

3.5 Algorithm 5: model-based RL model

Convergence to an optimal policy can be achieved after some learning time; however, due to the dynamicity of the operating environment, the convergence rate is unpredictable. While higher learning rate α may increase the convergence speed; the Q-value may fluctuate, particularly when the dynamicity of the operating environment is high because the Q-value is more dependent on its recent estimates now, rather than its previous experience [37]. Model-based RL model has been applied to increase the convergence speed. This approach estimates the state transition probability matrix **T**,

Table 13 RL model for Hu's routing scheme [36]

State	$s_t^i \in S = \{1, 2, \ldots, N\}$, each state s_t^i represents a node i in which a packet resides. N represents the number of nodes in a network
Action	$a_t^i \in A = \{1, 2, \ldots, J\}$, each action a_t^i represents a next-hop neighbor node j. J represents the number of node i's neighbor nodes
Reward	$r(s_t^i, a_t^i) = -g - w_1\left[c_r(s_t^i) + c_r(s_{t+1}^j)\right] + w_2\left[r_a(s_t^i) + r_a(s_{t+1}^j)\right]$ represents the cost incurred by forwarding a packet from node i to node j. g represents resource consumption (i.e. transmission and reception energies) incurred by a packet forwarding. Lower $c_r(s_t^i)$ and $c_r(s_{t+1}^j)$ indicates higher amount of residual energy at nodes i and j. Higher $r_a(s_t^i)$ and $r_a(s_{t+1}^j)$ indicates higher residual energy at nodes i and j compared to the average amount of residual energy among neighboring nodes. w_1 and w_2 are weight factors

which forms the model and represents the operating environment, and subsequently updates the Q-values using **T**. The state transition probability matrix **T** is a matrix comprised of the probability of transitioning from one state to another in a single time instant.

3.5.1 Hu's routing scheme

Hu and Fei [36] propose a routing scheme A(3) using a model-based RL model, and it has been shown to increase throughput P(1), as well as to reduce energy consumption P(3). The purpose is to select a next-hop neighbor node with higher residual energy, which subsequently sends packets towards the sink node.

Table 13 shows the RL model for the scheme. Note that, the model is a representation for a particular packet. In other words, the RL model is embedded in each packet. The state s_t^i represents the node in which a particular packet resides (or node i). The action a_t^i represents the selection of a next-hop neighbor node j. The reward $r(s_t^i, a_t^i)$ represents various types of energies, including transmission and residual energies, incurred for forwarding a packet to node $a_t^i = j$. Taking into account the residual energy avoids highly utilized routes (or hot spots) in order to achieve a balanced energy distribution among routes.

Node i's Q-function, which indicates the appropriateness of transmitting a packet from node i to node $a_t^i = j$, is updated at time $t + 1$ as follows:

$$Q_{t+1}^i(s_t^i, j) = r(s_t^i, a_t^i) + \gamma\left(P_{s_t^i s_t^i}^{a_t^i} \max_{k \in a_t^i} Q_t^i(s_t^i, k) + P_{s_t^i s_t^j}^{a_t^i} \max_{k \in a_t^j} Q_t^j(s_t^j, k)\right) \quad (12)$$

where $P_{s_t^i s_t^i}^{a_t^i}$ is the transition probability of an unsuccessful transmission from s_t^i (or node i) after taking action a_t^i, while $P_{s_t^i s_t^j}^{a_t^i}$ is the transition probability of a successful transmission from s_t^i to s_t^j (or node j) after taking action a_t^i. The transition probabil-

ities, which form the system model, estimate $P^{a_t^i}_{s_t^i s_t^i}$ and $P^{a_t^i}_{s_t^i s_t^j}$ using the historical data of each link's successful and unsuccessful transmission rates based on the outgoing traffic of next-hop neighbor nodes.

3.6 Algorithm 6: hierarchical RL model

The hierarchical RL model segregates the entire system into the upper and lower levels, and applies two separate RL approaches simultaneously in each level to achieve global and local optimal actions, respectively. Hence, a large and complex problem, such as routing [25], can be segregated into smaller problems, which can be solved simultaneously. Hence, the hierarchical RL model helps to reduce the state space, and so it improves scalability and convergence rate.

3.6.1 Hu's hierarchical routing scheme

Hu and Fei [25] propose a hierarchical routing scheme A(3) for clustered networks using a hierarchical RL model, and it has been shown to increase throughput P(1), as well as to reduce end-to-end delay P(2) and energy consumption P(3). The purpose is to select a next-hop node, which subsequently sends packets towards the sink node, with higher successful transmission rate.

Traditionally, in flat (or non-clustered) networks, Q-values keep track of a route cost comprised of multiple hops; however, any updates on a link cost must be propagated along a route, which may be slow in large networks. Hence, in [25], a clustered network using hierarchical routing A(3) is proposed. There are two types of routing schemes, namely intra- and inter-cluster routing schemes. The clusterhead performs inter-cluster routing to search for the best route to the sink node. The member nodes perform intra-cluster routing to search for the best route to a gateway node in the cluster. Gateway nodes are the member nodes located at the fringe of a cluster, and since they can hear from neighboring clusters, they provide inter-cluster communications. With respect to the hierarchical RL model, the upper and lower levels represent the inter-cluster and intra-cluster routing schemes, respectively. Hence, the upper and lower layers are comprised of clusterheads and member nodes, respectively. The upper layer supervises the lower layer so that member nodes search for the best route to the gateway node selected by the upper layer; while the lower layer provides evaluation feedback to the upper layer on the selection of the gateway node. This approach improves the sensitivity of a route towards changes in the network topology because an update now traverse a smaller number of hops and is confined to a particular cluster.

Table 14 shows the RL model for the scheme. Note that, the model is a representation for a particular packet. The state s_t^i represents the node in which a particular packet resides (or node i). The action a_t^i represents the selection of a next-hop neighbor node j; and so a series of actions will lead the packet to the gateway nodes rather than the sink node in traditional networks. Note that, there are two levels of Q-learning for intra- and inter-cluster routing schemes respectively; and Table 14 shows intra-cluster routing being implemented within each cluster. The reward representation is

Table 14 RL model for Hu's hierarchical routing scheme [25]

State	$s_t^i \in S = \{1, 2, \ldots, N\}$, each state s_t^i represents a node i in which a packet resides. N represents the number of nodes in a network
Action	$a_t^i \in A = \{1, 2, \ldots, J\}$, each action a_t^i represents a next-hop neighbor node j. J represents the number of node i's neighbor nodes
Reward	$r_{t+1}^i(a_t^i) = \begin{cases} -1, & \text{if successful transmission} \\ -c, & \text{if unsuccessful transmission} \end{cases}$

$r_{t+1}^i(a_t^i) = -1$ and $r_{t+1}^i(a_t^i) = -c$ if successful and unsuccessful packet transmissions, respectively. The negative reward indicates resource consumption and network performance deterioration (i.e. energy consumption and packet latency) for each packet transmission.

4 Performance enhancements

RL has been shown to achieve the following performance enhancements as shown in Table 15:

P.1 *Higher throughput.* Higher throughput indicates higher packet delivery rate, higher successful packet transmission rate, lower packet loss rate and lower number of packet retransmissions.

P.2 *Lower end-to-end delay/packet latency.* Lower end-to-end delay and packet latency in single-hop and multi-hop transmissions, respectively, indicate lower number of packets in the buffer queue.

P.3 *Lower energy consumption.* Lower energy consumption increases network lifetime. Since each sensor node operates on battery power, energy consumption is a common performance metrics. Other performance enhancements, such as higher throughput and lower end-to-end delay, may indicate lower energy consumption due to lower packet loss rate and number of packet retransmissions.

P.4 *Higher sensing coverage.* Higher sensing coverage indicates that, larger parts of an area is covered by at least a single sensor node, and so there is higher rate of detection of the events of interest. A sensing coverage scheme A(5) must minimize the overlapping of sensing coverage with neighboring nodes to reduce energy consumption.

P.5 *Higher route discovery rate.* Higher route discovery rate indicates higher success rate of finding a favorable route from a source node to a sink node. In [28], a favorable route must be free from malicious nodes, which drop packets received from previous hops.

P.6 *Higher in-contact time.* Higher in-contact time indicates greater possibility of a sensor node to discover the presence of a mobile data collector node, as well as longer duration for data transmission, in a sleep-wake scheduling scheme A(1.1) [12].

Table 15 Performance enhancement of RL-based schemes in WSNs

Application schemes	References	Performance enhancements					
		P.1 Higher throughput	P.2 Lower end-to-end delay	P.3 Lower energy consumption	P.4 Higher sensing coverage	P.5 Higher route discovery rate	P.6 Higher in-contact time
A.1 MAC	Alberola and Pesch [21]	×	×	×			
	Chu et al. [35,37]	×	×	×			
	Gummeson et al. [10]	×					
	Hsu et al. [11]			×			
	Liang et al. [13]	×	×				
	Liu and Elhanany [8]	×	×	×			
	Mao et al. [20]	×		×			
	Mao et al. [9]		×	×			
	Mihaylov et al. [30]		×	×			
	Niu and Deng [31]	×	×	×			
	Shah et al. [12]			×			×
A.2 Cooperative communications	Liang et al. [13]	×	×				
	Liang et al. [32]	×	×	×			
	Liang et al. [33]	×	×				
A.3 Routing	Dong et al. [5]	×	×	×			
	Liang et al. [26]	×	×				
	Hu and Fei [36]	×		×			
	Forster and Murphy [14]	×		×			
	Naputta and Usaha [28]					×	
	Villaverde et al. [24]	×	×	×			
A.4 Congestion avoidance	Tan et al. [16]	×	×	×			
A.5 Sensing coverage	Renaud and Tham [35]			×			
	Seah et al. [4]			×	×		

5 Open issues

This section discusses open issues that can be pursued in this research area.

5.1 Convergence rate and energy consumption

The convergence rate is affected by how much an agent learns, as well as the condition of the operating environment. Generally speaking, convergence rate increases with longer waking or active time (or duty cycle), as well as lower dynamicity and uncertainty levels of the operating environment. Shorter waking time reduces energy consumption and convergence rate; and it is suitable for operating environment with lower dynamicity and uncertainty levels. On the other hand, longer waking time increases convergence rate and energy consumption; and it is suitable for operating environment with higher dynamicity and uncertainty levels. Future research could be pursued to adjust waking duration in order to achieve a balanced tradeoff between convergence rate and energy consumption with respect to the condition of the environment.

5.2 Enhancement on the scalability of RL

The Q-table is a two-dimensional lookup table comprised of $|S| \times |A|$ entries. There are *two* important considerations with respect to scalability. Firstly, the number of entries increases exponentially with the number of states and actions; however, there may be limited memory capacity at each sensor node. Secondly, large number of state-action pairs requires higher number of explorations to discover most Q-values, and so it reduces the convergence rate to optimal action and increases energy consumption associated with learning (see Sect. 5.3). In addition to the state-action pairs, each sensor node must estimate and keep track of state transition probability matrix T in model-based RL model (see Sect. 3.5). Future research could be pursued to reduce the number of state-action pairs, as well as the state pairs in model-based RL model, without jeopardizing the accuracy of state and action space representations in order to improve scalability, as well as to reduce energy consumption.

5.3 Minimization of learning cost

The condition of the operating environment affects the usefulness and the recentness of the knowledge (or Q-value). An agent must make the right decision whether to learn or not. Generally speaking, learning should only take place if these two conditions are fulfilled. Specifically, the new knowledge remains useful for a long enough period of time. This means that the operating environment remains consistent for a long enough period of time (or with a certain low levels of dynamicity and uncertainty), and the rewards (e.g. throughput) received must be greater than the learning costs (e.g. energy consumption and processing power). Future research could be pursued to investigate mechanisms to make the right decisions whether to learn or not, as well as to reduce the learning cost.

5.4 Enhancement on the security aspect of RL

The requirements of the sensors and sink nodes to observe and learn from the operating environment have inevitably opened new security vulnerabilities. Malicious nodes may manipulate the operating environment so that the nodes learn the incorrect information. The malicious nodes may have learning capabilities and launch attacks in a cooperative manner. Consequently, the sensor nodes may converge to suboptimal actions, take longer to converge, or even fail to converge to optimal actions. This security vulnerability may be particularly pronounced in the cooperative RL model (see Sect. 3.4), in which nodes that receive manipulated information may become malicious themselves since they exchange information (e.g. states and Q-values) among themselves. Consequently, manipulated nodes may make sub-optimal local actions as part of the joint action. Future research could be pursued to investigate mechanisms to enhance the security vulnerabilities associated with the application of RL in WSNs.

5.5 Reduction of message exchange overhead

The requirement of the sensors and sink nodes to exchange information (e.g. states and Q-values) among themselves in order to learn from each other and the operating environment have inevitably increased the amount of control message exchange and energy consumption. However, message exchange is essential for learning in cooperative and hierarchical RL models (see Sects. 3.4, 3.6, respectively). Nevertheless, by reducing the message exchange frequency, the convergence rate to an optimal joint action may decrease. Future research could be pursued to investigate mechanisms to reduce the message exchange frequency without jeopardizing network-wide performance.

6 Conclusions

Reinforcement Learning (RL) has been applied in Wireless Sensor Networks (WSNs) to provide network performance enhancement in a wide range of schemes. To apply RL, several representations including state, action, as well as delayed and discounted rewards, are defined. Additionally, several features, including the Q-function, as well as exploration and exploitation mechanisms, must be defined. Based on the context of WSNs, this article presents an extensive review on the enhancements of these representations and features. Most importantly, this article presents an extensive review on a wide range of RL models and enhanced RL algorithms in the context of WSNs. The enhanced algorithms provide insights on how various schemes in WSNs can be approached using RL. Performance enhancements achieved by the traditional and enhanced RL algorithms in WSNs are presented. Certainly, there is a great deal of future work in the use of RL, and we have raised open issues in this article.

Acknowledgments This work was supported by the Malaysian Ministry of Education (MOE) under Fundamental Research Grant Scheme (FRGS/1/2014/ICT03/SYUC/02/2).

References

1. Ghataoura DS, Mitchell JE, Matich GE (2011) Networking and application interface technology for wireless sensor network surveillance and monitoring. IEEE Comm Magazine 49(10):90–97
2. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
3. Zhang D, Ma H (2007) A Q-learning-based decision making scheme for application reconfiguration in sensor networks. CSCWD'07 proc 11th Intl Conf Comp Supported Cooperative Work in Design. IEEE, Melbourne, Australia, pp 1122–1127
4. Seah MWS, Tham CK, Srinivasan V, Xin A (2007) Achieving coverage through distributed reinforcement learning in wireless sensor networks. ISSNIP'07 proc 3rd Intl Conf Intell Sensors, Sensor Net and Info. IEEE, Melbourne, Australia, pp 425–430
5. Dong S, Agrawal P, Sivalingam K (2007) Reinforcement learning based geographic routing protocol for UWB wireless sensor network. GLOBECOM'07 proc Global Telecomm Conf. IEEE, Washington, DC, pp 652–656
6. Yau K-LA, Komisarczuk P, Teal PD (2012) Reinforcement learning for context awareness and intelligence in wireless networks. Elsevier J Net Comp App 35(1):253–267
7. Chu Y, Mitchell PD, Grace D (2012) Reinforcement learning based ALOHA for multi-hop wireless sensor networks with informed receiving. WSS'12 proc IET Conf Wls Sensor Sys. IEEE, London, UK, pp 1–6
8. Liu Z, Elhanany I (2006) RL-MAC: a reinforcement learning based MAC protocol for wireless sensor networks. Inderscience Intl J Sensor Net 1(3/4):117–124
9. Mao J, Xiang F, Lai H (2009) RL-based superframe order adaptation algorithm for IEEE 802.15.4 networks. In: CCDC'09 proc Ch Ctrl and Decision Conf. IEEE, Guilin, China, pp 1–5
10. Gummeson J, Ganesan D, Corner MD, Shenoy P (2010) An adaptive link layer for heterogeneous multi-radio mobile sensor networks. IEEE J Sel Area Comm 28(7):1094–1104
11. Hsu RC, Liu CT, Wang KC, Lee WM (2009) QoS-aware power management for energy harvesting wireless sensor network utilizing reinforcement learning. CSE'09 proc Intl Conf Comp Sc and Engin. IEEE, Vancouver, Canada, pp 537–542
12. Shah K, Francesco MD, Anastasi G, Kumar M (2011) A framework for resource-aware data accumulation in sparse wireless sensor networks. Elsevier J Comp Comm 34(17):2094–2103
13. Liang X, Chen M, Xiao Y, Balasingham I, Leung VCM (2010) MRL-CC: a novel cooperative communication protocol for QoS provisioning in wireless sensor networks. Inderscience Intl J Sensor Net 8(2):98–108
14. Forster A, Murphy AL (2009) Clique: role-free clustering with Q-learning for wireless sensor networks. ICDCS'09 proceedings 29th IEEE Intl Conf Dist Comp Sys. IEEE, Quebec, Canada, pp 441–449
15. Saoseng JY, Tham CK (2006) Coordinated rate control in wireless sensor network. ICCS'06 proc 10^{th} IEEE Singapore Intl Conf Comm Sys. IEEE, Singapore, pp 1–5
16. Tan H, Zhao L, Liu W, Niu Y, Zhao C (2011) Adaptive congestion avoidance scheme based on reinforcement learning for wireless sensor network. ICCTA'11 proc IET Intl Conf Comm Tech and App. IEEE, Beijing, China, pp 228–232
17. Tham CK, Renaud JC (2005) Multi-agent systems on sensor networks: a distributed reinforcement learning approach. ISSNIP'05 proc Intl Conf Intell Sensors, Sensor Net and Info. IEEE, Melbourne, Australia, pp 423–429
18. Khan MI, Rinner B (2012) Resource coordination in wireless sensor networks by cooperative reinforcement learning. PERCOMW'12 proc IEEE Intl Conf Pervasive Comp and Comm Workshops. IEEE, Lugano, Switzerland, pp 895–900
19. Mahadevan S (1994) To discount or not to discount in reinforcement learning: a case study comparing R learning and Q learning. ICML'94: Proceedings of the 11th International Conference on Machine Learning. Morgan Kaufmann, Amherst, MA, pp 164–172
20. Mao S, Tang H, Zhou L, Ma X (2011) An energy conservation optimization strategy for wireless sensor network node based on Q-learning. ASCC'11 proc Asian Ctrl Conf. IEEE, Kaohsiung, Taiwan, pp 938–943
21. Alberola RdP, Pesch D (2012) Duty cycle learning algorithm (DCLA) for IEEE 802.15.4 beacon-enabled wireless sensor networks. Elsevier Ad Hoc Net 10(4):664–679
22. Arroyo-Valles R, Alaiz-Rodriguez R, Guerrero-Curieses A, Cid-Sueiro J (2007) Q-probabilistic routing in wireless sensor networks. ISSNIP'07: Proc 3rd Intl Conf Intell Sensors, Sensor Net and Info. IEEE, Melbourne, Australia, pp 1–6

23. Forster A, Murphy AL (2011) FROMS: a failure tolerant and mobility enabled multicast routing paradigm with reinforcement learning for WSNs. Elsevier Ad Hoc Net 9(5):940–965
24. Villaverde BC, Rea S, Pesch D (2012) InRout: a QoS aware route selection algorithm for industrial wireless sensor networks. Elsevier Ad Hoc Net 10(3):458–478
25. Hu T, Fei Y (2012) MURAO: a multi-level routing protocol for acoustic-optical hybrid underwater wireless sensor networks. SECON'12 proc 9th Ann IEEE Comm Soc Conf Sensor, Mesh and Ad hoc Comm and Net. IEEE, Seoul, South Korea, pp 218–226
26. Liang X, Balasingham I, Byun SS (2008) A multi-agent reinforcement learning based routing protocol for wireless sensor networks. ISWCS'08 proc IEEE Intl Symp on Wls Comm Sys. IEEE, Reykjavik, Iceland, pp 552–557
27. Liang X, Balasingham I, Byun SS (2008) A reinforcement learning based routing protocol with QoS support for biomedical sensor networks. ISABEL'08 proc 1st Intl Symp App Sc and Biomedical and Comm Tech. IEEE, Aalborg, Denmark, pp 1–5
28. Naputta Y, Usaha W (2012) RL-based routing in biomedical mobile wireless sensor networks using trust and reputation. ISWCS'12 proc 9th Intl Symp Wls Comm Sys. IEEE, Paris, France, pp 521–525
29. Chu Y, Mitchell PD, Grace D (2012) ALOHA and Q-learning based medium access control for wireless sensor networks. ISWCS'12 proc Intl Symp Wls Comm Sys. IEEE, Paris, France, pp 511–515
30. Mihaylov M, Borgne YAL, Tuyls K, Nowe A (2012) Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks. Inderscience Intl J Comm Net Distrib Sys 9(3/4):207–224
31. Niu J, Deng Z (2013) Distributed self-learning scheduling approach for wireless sensor network. Elsevier Ad Hoc Net 11(4):1276–1286
32. Liang X, Chen M, Leung VCM, Balasingham I (2010) Soft QoS provisioning for wireless sensor networks: a cooperative communications approach.In: CHINACOM'10: Proceedings of 5th Intl ICST Conf Commu and Net in China. IEEE, Beijing, China, pp 1–8
33. Liang X, Balasingham I, Leung VCM (2009) Cooperative communications with relay selection for QoS provisioning in wireless sensor networks. GLOBECOM'09 proc Global Telecomm Conf. IEEE, Honolulu, Hawaii, pp 1–8
34. Maalej M, Besbes H, Cherif S (2012) A cooperative communication protocol for saving energy consumption in WSNs. ComNet'12 proc Intl Conf Comm and Net. IEEE, Kunming, China, pp 1–5
35. Renaud JC, Tham CK (2006) Coordinated sensing coverage in sensor networks using distributed reinforcement learning. ICON'06 proc 14th IEEE Intl Conf Net. IEEE, Singapore, pp 1–6
36. Hu T, Fei Y (2010) QELAR: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. IEEE Trans Mob Comp 9(6):796–809
37. Yau K-LA, Komisarczuk P, Teal PD (2011) Achieving context awareness and intelligence in distributed cognitive radio networks: a payoff propagation approach. In: WAINA'11 proc IEEE Workshops Intl Conf Ad Info Net and App. IEEE, Singapore