

Research Article

Energy-Efficient Monitoring in Software Defined Wireless Sensor Networks Using Reinforcement Learning: A Prototype

Ru Huang,¹ Xiaoli Chu,^{1,2} Jie Zhang,^{1,2} and Yu Hen Hu^{1,3}

¹*School of Information Science & Engineering, East China University of Science and Technology, Shanghai 200237, China*

²*Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, UK*

³*Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI 53705, USA*

Correspondence should be addressed to Ru Huang; huangrabbit@ecust.edu.cn

Received 27 July 2015; Revised 31 August 2015; Accepted 31 August 2015

Academic Editor: L. Javier García Villalba

Copyright © 2015 Ru Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software defined wireless networks (SDWNs) present an innovative framework for virtualized network control and flexible architecture design of wireless sensor networks (WSNs). However, the decoupled control and data planes and the logically centralized control in SDWNs may cause high energy consumption and resource waste during system operation, hindering their application in WSNs. In this paper, we propose a software defined WSN (SDWSN) prototype to improve the energy efficiency and adaptability of WSNs for environmental monitoring applications, taking into account the constraints of WSNs in terms of energy, radio resources, and computational capabilities, and the value redundancy and distributed nature of data flows in periodic transmissions for monitoring applications. Particularly, we design a reinforcement learning based mechanism to perform value-redundancy filtering and load-balancing routing according to the values and distribution of data flows, respectively, in order to improve the energy efficiency and self-adaptability to environmental changes for WSNs. The optimal matching rules in flow table are designed to curb the control signaling overhead and balance the distribution of data flows for achieving in-network fusion in data plane with guaranteed quality of service (QoS). Experiment results show that the proposed SDWSN prototype can effectively improve the energy efficiency and self-adaptability of environmental monitoring WSNs with QoS.

1. Introduction

Wireless sensor networks (WSNs) are application-oriented information-centric networks, which are characterized by limited energy and constrained radio resources [1]. One typical application of WSNs is environmental monitoring, where data-gathering based environmental monitoring tasks are executed by nodes with heterogeneous sensing and programmable functions. Each node in WSNs could be equipped with multiple sensors for different sensing purposes, for example, temperature, humidity, light, and vibration. In WSNs, the time-varying wireless communication environment and random interference may lead to unreliable communication links, while switching on/off of network nodes due to energy constraints can cause unpredictable topology changes, making it difficult to guarantee reliable and adaptive data-gathering for monitoring applications.

Software defined wireless networks (SDWNs) enable programmable control in network and virtualization of network equipment by decoupling control plane and data plane [2]. The logic centralization and simplified abstraction of control plane can improve the scalability and multitasking efficiency [3]. The combination of SDWNs based architecture and WSNs, that is, software defined wireless sensor network (SDWSN), would bring the following advantages:

- (i) SDWNs based abstraction of network control plane can effectively reduce the cost of WSN expansion and operation.
- (ii) SDWNs based virtualization of network equipment and programmable control of common hardware and software enabled flexible task configuration, high resource utilization, and simplified network management in WSNs.

However, to realize the above advantages of SDWSN for monitoring applications is not without difficulties. The control-data decoupled structure of SDWNs relies on cross-plane control traffic, which may result in excessive communication overhead and transmission delay. In SDWSN, although different virtual networks can work together on top of the same physical infrastructure, the centralized control plane may lead to high energy costs due to information collection for reaching a global view, and the multiple virtual networks may compete for common physical network resources. If a large number of flows simultaneously request a switch to forward data, network congestion or even crash may occur. Furthermore, energy- and resource-constrained WSNs might not have the sufficient network resources to realize the dynamic resource allocation and QoS of SDWNs. Therefore, the energy and resource utilization of SDWSN need to be carefully designed for resource-constrained and application-oriented WSNs.

Most existing works on SDWSN focus on providing QoS guarantee or optimizing network management for monitoring applications. The software defined information centric network (SDN-IC) [4] floods the network with packets so as to leave reverse path information at routers, but that method will cause frequent duplication of packets and lead to huge communication loads, which increase not only end-to-end delay but also energy consumption. The resource allocation in a software-driven wide-area network (SWAN) was optimized by an agent-based traffic engineering scheme [5], which requires excessive information exchange between the controller and switches for tracking network topology and traffic distribution changes. With the increase of network density, the SWAN would be plagued by large overhead caused by collisions between candidate relays contending for media. The software defined vehicular ad hoc network (SDV) [6] uses network virtualization to allocate network traffic in a programmable fashion, where surveillance packets are delivered following a position-aided data-gathering mechanism with greedy perimeter stateless routing (GPRS) [7] in case of controller failure. However, the SDV controller needs to gather and maintain a large amount of information for transmission power control, which is not practical for large-scale monitoring WSNs. In [8], the energy consumption of a multitask SDWSN was minimized for monitoring applications with guaranteed quality-of-sensing by solving a mixed integer linear programming problem at a high computation complexity. In [9, 10], the load-balancing routing algorithms for WSNs construct an optimal routing tree by minimizing the total weight of routing paths, where the path weights are modeled as a function of energy consumption. However, none of these works has adequately considered the application-oriented features of flows and in-network data fusion in complex and dynamic monitoring environments for SDWSN, thereby significantly limiting their energy efficiency and environmental adaptability.

In this paper, we develop an energy-efficient cognitive SDWSN prototype for environmental monitoring application, where high computational complexity management of data fusion and data routing are centralized in control plane, while low computational complexity execution of algorithms

is implemented in data plane. The cognitive mechanism based on reinforcement learning (RL) [11] is embedded in control plane for information processing, where the interactions (in terms of reward or punishment) between agents and the environment are utilized to enhance the intelligence in policy decision making and to improve the self-adaptability of the energy-saving mechanisms in dynamic environments. Particularly, we propose to mine the application-specific value redundancy of flows in periodic transmissions of monitoring data using an autoregressive moving average (ARMA) [12] based time series forecast model. We design RL based mechanisms to perform value-redundancy filtering and load-balancing routing according to the values and distribution of flows, respectively, in order to improve the energy efficiency and self-adaptability to environmental changes of WSNs. Furthermore, the actions of control plane are mapped to low-complexity vector calculations and rule matching in switch's flow table. The rules in flow table are designed to curb the control signaling overhead and balance the distribution of data flows for achieving in-network fusion in data plane.

The novel aspects of the proposed energy-efficient SDWSN prototype are

- (i) energy saving with guaranteed QoS is achieved by mining the application-specific value redundancy and distribution of data flows in SDWSN, taking into account the inherent constraints of WSNs in terms of energy, radio resources, and computational capabilities;
- (ii) the RL based mechanisms for value-redundancy filtering and load-balancing routing can adapt to the varying environment and network status, thus improving the self-adaptability of SDWSN for monitoring applications.

The rest of the paper is organized as follows. Section 2 elaborates the cognitive SDWSN prototype and its functional architecture. Section 3 presents a specific implementation of the proposed prototype. In Section 4, performance of the proposed SDWSN prototype in terms of energy efficiency and self-adaptability is evaluated through experiments in comparison with existing WSN approaches for monitoring. Finally, conclusion is drawn in Section 5.

2. Functional Architecture of SDWSN Prototype

In this section, we propose a cognitive SDWSN prototype, where RL is incorporated into the network information process for an integrated consideration of the energy- and resource-constrained trait of WSNs, complex features of monitoring applications, and dynamic nature of WSN deployment environments.

As shown in Figure 1, the fundamental functionalities of SDWSN prototype include an information QoS setting module, cognitive information middleware (CIM), and an

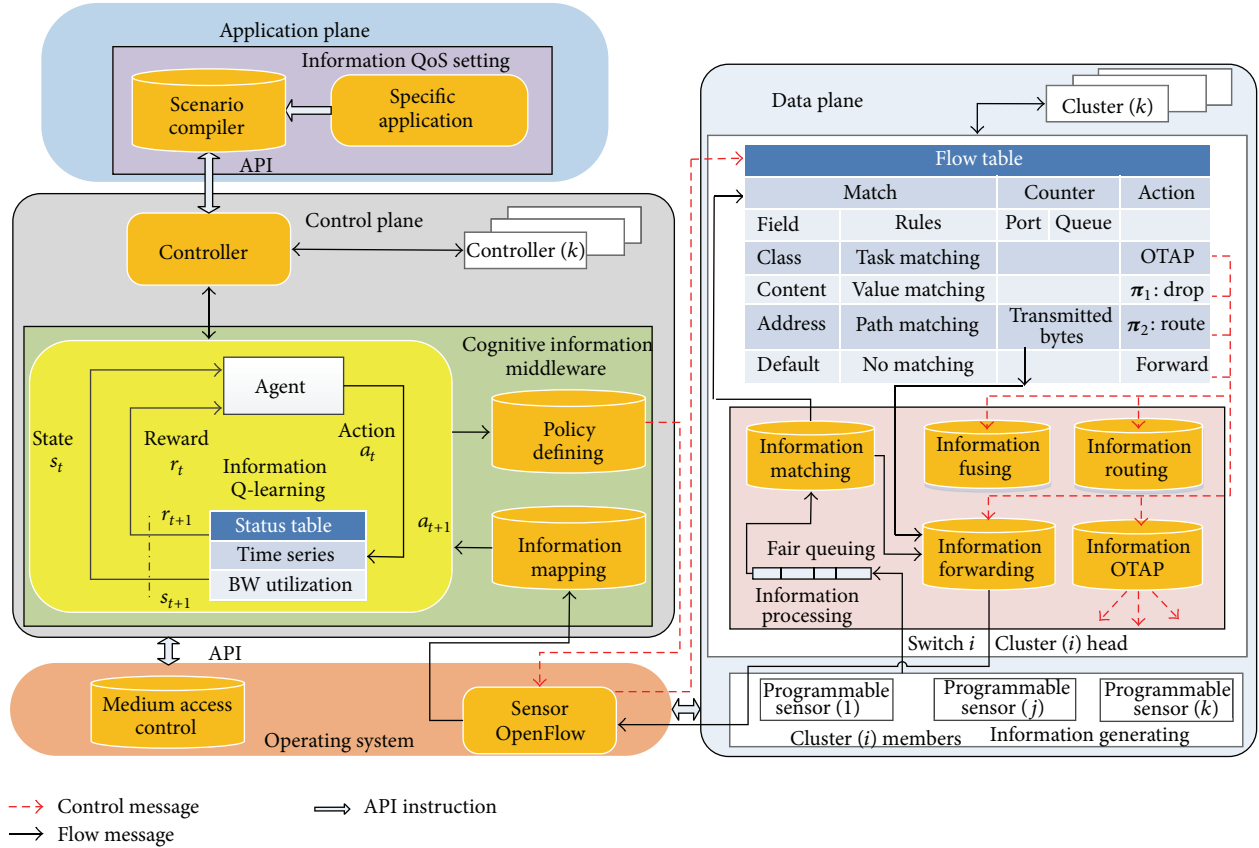


FIGURE 1: Cognitive SDWSN prototype.

information processing module. Following the design principles of SDWNs, the application plane of SDWSN prototype is designed to meet the QoS requirements of monitoring applications, supported by the hardware of sensors. Application plane interacts with control plane through an application programming interface (API). The functionalities of data plane are dynamically configured using Over-the-Air Programming (OTAP) technique [13], which can run multiple tasks simultaneously with QoS and can reduce the energy consumption in online task scheduling.

The data plane is abstracted into a weighted directed graph, $G = (V, L, E)$, which forms a reverse multicast tree with V being the set of vertexes, L being the set of links, and E being the set of link weights. Each node in the vertex set V maintains a hierarchical cluster, where a switch acts as the cluster leader and other programmable nodes become cluster members constituting the monitoring information generating (IG) module. The data flows generated by members of the same cluster form a programmable set of packets that share certain properties, because the packets of a flow are handled by matching "Field" and "Rules" in a switch's flow table (see Figure 1) and by imposing the "Action" set to execute the preset policy. The weight of each link indicates the status of flow property distribution and frequency bandwidth allocation during a certain data-gathering round and is defined as the function of link bandwidth utilization (BWU) (see Section 3).

In the control plane, CIM is a part of the controller that performs adaptive data mining of network information using machine learning schemes with QoS guarantee. Information mapping (IM) module in CIM is responsible for preprocessing information received from data plane (i.e., information mining). It has two main duties: to perform online evaluation of the value of current monitoring data flow using an ARMA model and to build a flow distribution map and a network interconnection map in G. Information Q-learning (IQ) module utilizes the results from IM module to produce optimal strategies, which are then transformed by policy defining (PD) module into a set of policies to be inserted into switch's flow table for a lightweight implementation of data plane.

Routing decisions are made by CIM in the controller and then translated into rules and actions to be deployed in flow tables. APIs are used to configure flow tables for routing, in conjunction with a floodless service discovery mechanism. As part of the operating system, Sensor OpenFlow (SOF) [14] channel is used to establish an end-to-end connection between the controller and a switch. SOF also supports queries on packet streams and automatically splits queries between the data plane and the control plane, thus avoiding the increase of traffic in the data plane due to queries. Value matching and path matching are designed by CIM and executed by lightweight actions of flow tables in an on-demand driven energy-saving mode. This reduces the amount of

information exchange between the operating system and the data plane. After data-gathering routing has been established, data packets can be forwarded and processed in the data plane. Subsequent (follow-up) packets in a flow are forwarded in the data plane based on the configured routing in flow tables without any further participation from the control plane. This can reduce the data-gathering traffic in the data plane and decrease control overhead in the control plane. The features specified in the MAC layer (as part of the operating system) are logically partitioned into two different modules: the lower MAC module, which depends on the proprietary Hardware Abstraction Layer (HAL) and controls time critical functions to achieve value-redundancy fusing in the data plane based on service differentiation access control; and the upper MAC module, which is responsible for delay-tolerant control plane functions.

The proposed cognitive SDWSN prototype takes into account the inherent constraints of WSNs in terms of energy, radio resources, and computational capabilities. Energy saving is achieved through the design of value-redundancy data fusing and load-balancing data routing technologies in CIM. By using machine learning schemes for energy saving in control plane and by incorporating lightweight execution using a flow table at each switch in data plane, intelligence and controllability can be achieved in all stages of the information operation chain in SDWSN. The introduction of in-network processing with low computational complexity in data plane facilitates the centralization of QoS management in control plane, thus reducing the total amount of overheads for cross-plane communications. Moreover, low-complexity numerical operations of the flow entries are enabled by IM module, which matches CIM's outputs to vector constant parameters.

Based on the proposed SDWSN prototype, programmability and resource reutilization in data plane can be improved through OTAP. The overhead for cross-plane control signaling can be reduced by introducing a data fusion mechanism into data plane, which also improves the controllability of packet routing and the efficiency of resource utilization.

3. Design of RL Based Energy-Saving Mechanisms

In this section, the implementation of the proposed SDWSN prototype for environmental monitoring applications will be discussed with a focus on RL based energy-saving mechanisms.

3.1. Design of Energy-Saving Mechanisms in Control Plane. In event-detection based monitoring applications, the periodic transmissions of monitoring data usually have low duty cycles and high time-domain correlation, resulting in data value redundancy. In the following, we exploit the data value redundancy to achieve transmission energy saving.

RL is an agent based learning approach, which uses the trial and error method to find a reward maximizing behavior in a dynamic environment. RL can adapt to the dynamic environment with a relatively low complexity, rendering itself

perfectly applicable to WSNs with limited resources and operating in unpredictable environments. Therefore, we design the energy-saving policy Γ (see (4)) in control plane based on RL. The key design of Γ is the utilization of contention window (CW) in QoS-aware media access control (MAC), which exploits the concept of service differentiation MAC [15]. CW introduces a MAC back-off counter, called Failed Times (FT), to count the number of failures before winning the contention, which can avoid long time occupancy of media caused by a large CW. The threshold of FT sets the retry limit in MAC and can be considered as CW size. According to Γ , we propose the value-redundancy filtering mechanism π_1 and the load-balancing routing mechanism π_2 , where π_1 performs online estimation of flow values and implements an optimal in-network fusion strategy, and π_2 performs online analysis of flow distribution and adaptive optimization of path weights.

During a specific data-gathering round B ($B \in \mathbb{Z}^+$), the monitoring data flow generated by node i can be modeled as a limited time series $\mathbf{X} = \{x_t^i\}$, $t \in \mathbf{T}_B$, $\mathbf{T}_B = \{j, j+1, \dots, j+m-1\}$, $i \in \mathbf{V}$, $m \in \mathbb{Z}^+$, where $j = (B-1)m+1$ denotes the sampling time instant and m denotes the learning queue length. The time span of each data-gathering round is mT_s , where T_s is the sampling period.

Since the ARMA model captures the statistical characteristics of a time series, which can be used to mine the sampled data value redundancy and to perform real-time value evaluation of data flows, we adopt ARMA to predicate the value \hat{x}_t^i and calculate the corresponding prediction error $\|x_t^i - \hat{x}_t^i\|_p$, where $\|\cdot\|_p$ denotes the normalized Frobenius norm [16], for example, $\|A\|_{p=2} = [\text{Tr}(A^*A)]^{1/2}$. Note that if some important events occur, the distribution of monitoring data flows among nodes would become uneven. According to the information entropy theory [17], greater variations in data flow values indicate a larger average amount of information contained in the data flows and the higher probability of important events occurring under the premise of no external interference/influence. Thus, we define the value factor F_B^i to estimate the underlying value of data flow \mathbf{X} generated by node i in the B th data-gathering round as follows:

$$\begin{aligned} F_B^i &= \delta_t^i * \lambda_t^i \\ \text{s.t. } \delta_t^i &= \frac{\sum_{t=j}^{j+m-1} \|x_t^i - \hat{x}_t^i\|_p}{m}; \\ \lambda_t^i &= \left\| \left(\sum_t o_t^i - k_{\text{th}} \right)^+ \right\|_p; \\ o_t^i &= \begin{cases} 1, & \|x_t^i - \hat{x}_t^i\|_p > \delta_t^i \\ 0, & \text{otherwise;} \end{cases} \\ (\cdot)^+ &= \max(\cdot, 0); \end{aligned} \tag{1}$$

$$t \in \mathbf{T}_B, i \in \mathbf{V}, \forall x_t^i \in \mathbf{X}, j \in \mathbb{Z}^+, k_{\text{th}} \in \mathbb{Z}^+, p = 2,$$

where δ_t^i is the mean of prediction errors, λ_t^i is an anti-interference factor designed to avoid misjudgment of values

caused by environmental disturbance, and $o_t^i = 1$ if the prediction error at time instant t is larger than δ_t^i ; otherwise $o_t^i = 0$. If the number of significant prediction errors in data flow \mathbf{X} reaches the threshold k_{th} , that is, $\sum_t o_t^i \geq k_{th}$, then \mathbf{X} is considered as a high value level; otherwise, those significant prediction errors are considered as the result of external interference. The anti-interference capability increases with the value of k_{th} .

After the value of data flow \mathbf{X} having been predicted, the historical forwarding record of \mathbf{X} during the B th data-gathering round needs to be extracted based on link statistics obtained from the counter field in switch's flow table, in order to analyze the link state information and calculate the BWU. Accordingly, the real-time state element for RL can be obtained as

$$\begin{aligned} \theta_B^i &= \left\| \begin{pmatrix} F_B^i \\ e_B^{k_i} \end{pmatrix} \right\|_p \\ \text{s.t. } e_B^{k_i} &= \frac{N_B^{k_i}}{l_B^{k_i}}; \\ l_B^{k_i} &= \frac{1}{|O^{(i)}|} \sum_h l_B^{(i,h)} \\ \forall i \in \mathbf{V}, k_i = \langle i, h \rangle \in \mathbf{L}, h \in O^{(i)} \subset \mathbf{V}, t \in \mathbf{T}_B, e_B^{k_i} \in \mathbf{E}, p = 2. \end{aligned} \quad (2)$$

In (2), θ_B^i reflects the local status of flow value and flow allocation used for resource utilization in SDWSN, where $e_B^{k_i}$ is the BWU and $N_B^{k_i}$ denotes the traffic throughput of node i on link k_i in the B th data-gathering round, which can be calculated as the incremental number of "transmitted bytes" from node i on link k_i in the B th data-gathering round. $l_B^{k_i}$ indicates the mean bandwidth of links in set $\{\langle i, h \rangle\}_h$, where node h belongs to $O^{(i)}$, which denotes the neighborhood of node i with size $|O^{(i)}|$. Then, the threshold of FT in CW is set as an inversely proportional function of θ_B^i using

service differentiation based MAC retransmission protocol [15]. Therefore, the value of data flow \mathbf{X} and its historical forwarding records can be mapped to the corresponding probability of channel access in MAC layer. The adaptive optimization of CW can be formulated as an average reward Markov decision process (ARMDP) [18]. Accordingly, we design an ARMDP based RL mechanism to optimize CW for the purpose of finding the optimal energy-efficient policy. The RL mechanism is executed by CIM, which adaptively adjusts the channel access probability to inhibit the transmission of value-redundant loads and balance the distribution of transmission loads among nodes while guaranteeing QoS.

In the RL mechanism, the dynamic environment is characterized by a 4-tuple $(\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R})$, where \mathbf{S} is the set of network states updated in each data-gathering round and consisting of value factors F_B^i and BWU $e_B^{k_i}$ stored in the status table embedded in CIM; that is,

$$\begin{aligned} s_B &= (F_B^i, e_B^{k_i}) \in \mathbf{S}, \\ \forall i \in \mathbf{V}, k_i = \langle i, h \rangle \in \mathbf{L}, h \in O^{(i)} \subset \mathbf{V}, e_B^{k_i} \in \mathbf{E}, B \in \mathbb{Z}^+, \end{aligned} \quad (3)$$

thereby providing a real-time observation of the environment for Q-learning; \mathbf{A} is the action set produced by PD module and injected into the "Match Field" of flow table in the switch (see Figure 1); $\mathbf{P} : \mathbf{S} \times \mathbf{A} \times \mathbf{S}$ denotes the state transition probability; and the reward function $\mathbf{R} : \mathbf{S} \times \mathbf{A} \rightarrow \mathbf{R}$ indicates the environmental reward to the corresponding action for improving energy efficiency. The global reward is accumulated by maximizing the local reward in each data-gathering round by CIM. For a given QoS, the local reward calculated by (4) increases with a higher value F_B^i of the data flow and a lower value of the BWU $e_B^{k_i}$. Conversely, if the current value of data flow is low and/or the bandwidth is overutilized, then RL mechanism will generate a zero income or a negative reward as punishment.

The CW optimization strategy is shown in Algorithm 1, where Γ optimizes the size of CW $|\mathbf{W}^i|$ by maximizing the accumulated local reward and amending the criteria of action evaluation in an iterative manner:

$$\begin{aligned} \Gamma : \Psi_B &= \begin{cases} \Psi_{B-1} + \zeta, & Q_B(s_B, a_B) \geq Q_B(s_{B-1}, a_{B-1}) + \varepsilon \\ \Psi_{B-1} - \zeta, & Q_B(s_B, a_B) < Q_B(s_{B-1}, a_{B-1}) - \varepsilon \end{cases} \\ \text{s.t. } \Psi_1 &= \theta_1^i; \\ Q_B(s_{B-1}, a_{B-1}) &= Q_{B-1}(s_{B-1}, a_{B-1}) + \lambda * \left[r_B + \gamma * \max_{a_B} Q_{B-1}(s_B, a_B) - Q_{B-1}(s_{B-1}, a_{B-1}) \right]; \\ r_B &= \frac{F_B^i}{e_B^{k_i}}; \end{aligned} \quad (4)$$

$$\exists \varepsilon > 0, \exists \zeta \geq 0, \lambda \in (0, 1], \gamma \in (0, 1], B \in \mathbb{Z}^+, s_B \in \mathbf{S}, r_B \in \mathbf{R}, a_B \in \mathbf{A},$$

```

① Initialize Q table entry for each  $(s, a)$  pair and reward  $r$ 
Setting parameters  $\lambda, \gamma, s, a, t = 0, Q(s_B, a_B) = 0, B = 1$ ;
② Perform the following steps during each round:
{Do calculate the QoS factor of flow
  Observe the current state  $s_B \in \mathbf{S}$  according to (3)
  While (the current run < iteration threshold)
    {Adjust  $\Gamma$  with given  $Q$  according to (4);
    Select action  $a_B$  according to state transition probability according to (6);
    Execute the adjustment of window's size according to (5);
    Maximize per-round local reward  $r_B$ ;
    Observe the new state  $s_{B+1}$ ;
    Learn  $s, a, r$ 
    Execute the iterative process of  $Q$ 
    Select next step action  $a_{B+1}$ ;
    Update Status  $s_B \leftarrow s_{B+1}$ ;
  End while loop
}
Get the reward from the current run
 $B \leftarrow B + 1$ ;
If  $B$  is lower than the iteration threshold, then go to Step ②
End do loop
}
③ Obtain the optimal policy  $\Gamma$ , which is inputted into flow-table based on  $Q(s, a)$  and historical behavior rewarding.

```

ALGORITHM 1: Contention window optimization strategy based on RL.

where λ and γ denote the learning factor and the discount factor, respectively, s_B and a_B are the state and the action in the B th data-gathering round, and ε and ζ represent the tuning step sizes for updating values of Ψ_B and Q function [19], respectively.

Status table in CIM tracks how the operational environment evolves with time, where new states can be mined and new actions should be discovered. Policy Γ needs to be constantly updated to match the state-event pairs with the optimal actions. RL mechanism uses a random strategy to fully explore the state space at the beginning and adopts a greedy strategy to ensure convergence later on. According to Γ outputted by CIM, controller executes the optimal action with local maximum Q -value as follows:

$$\begin{aligned}
 &\text{Action: } a_B \\
 &= \begin{cases} |\mathbf{W}_B^i| \leftarrow |\mathbf{W}_B^i| + \Delta |\mathbf{W}^i|; & \theta_B^i < \|\Psi_B\|_p \\ |\mathbf{W}_B^i| \leftarrow |\mathbf{W}_B^i|; & \theta_B^i = \|\Psi_B\|_p \\ |\mathbf{W}_B^i| \leftarrow |\mathbf{W}_B^i| - \Delta |\mathbf{W}^i|; & \theta_B^i > \|\Psi_B\|_p \end{cases} \quad (5) \\
 &\text{s.t. } \Delta |\mathbf{W}^i| = \frac{1}{\theta_B^i * \omega_{\text{QoS}}}; \\
 &\omega_{\text{QoS}} \in [0, 1], \quad p = 2,
 \end{aligned}$$

where $\Delta |\mathbf{W}^i|$ indicates the adjustable size of CW. After the B th iteration, Ψ_B is obtained according to the comparison of Q value between the B th and the $(B - 1)$ th rounds, and QoS factor ω_{QoS} represents the QoS requirement of a given monitoring application. The value of ω_{QoS} can be adjusted by

the information QoS setting module in application plane and then be fed into the “Match Rules” in switch’s flow table (see Figure 1) via API to realize programmable network control. Accordingly, the size of CW is adjusted by node i executing action a_B based on the following transfer function:

$$p(a_B) = \frac{\chi_{\Gamma}^{s_B}(a_B)}{\sum_{a_B \in \mathbf{A}} \chi_{\Gamma}^{s_B}(a_B)}, \quad (6)$$

$p(a_B) \in \mathbf{P}, \quad a_B \in \mathbf{A}, \quad s_B \in \mathbf{S},$

where $\chi_{\Gamma}^{s_B}(a_B) = Q_B(s_B, a_B) / [\sum_B Q_B(s_B, a_B)]$ denotes the state assessment for action a_B under state s_B using Γ . Thus, CW size, that is, $|\mathbf{W}^i|$, can be adaptively adjusted according to dynamic environment status and application QoS, with the vibration amplitude $\Delta |\mathbf{W}^i|$ given by the inverse of θ_B^i . According to the MAC protocol for differentiated services [20], the probability of packet forwarding is inversely proportional to $|\mathbf{W}^i|$.

Based on the above analysis, we design π_1 and π_2 as follows:

- (i) π_1 reduces the total energy consumption in SDWSN by inhibiting the transmission of value-redundant loads. In π_1 , F_B^i is used as one important element to control $|\mathbf{W}^i|$. According to (5), a low-value flow will be configured with a high medium access delay caused by the corresponding large number of retries set by MAC back-off counter. This would lead to a low forwarding probability. The low-value flow will be discarded when the medium access delay

goes beyond a preset threshold of FT. Therefore, the probability of traffic forwarding can be adaptively controlled according to the flow value. Suppressing the transmission of low-value flows greatly decreases the amount of in-network traffic for data-gathering and thus achieves energy saving.

- (ii) π_2 balances the energy distribution across SDWSN by minimizing the total weight of routing paths in \mathbf{G} , that is, $\min\{\sum e_B^{k_i}\}, \exists k_i \in \text{Tree}_B^{\text{optim}} \subset \mathbf{L}$, to construct an optimal routing tree, $\text{Tree}_B^{\text{optim}} = \{k_i\}$, $k_i \in \mathbf{L}$, $i \in \mathbf{V}$, where the average link BWU $e_B^{k_i}$ is used as the link weight for controlling the size of CW. According to (2) and (5), a link with a higher BWU will be configured with a larger CW size, leading to a lower probability of traffic forwarding. Therefore, the distribution of network traffic can be balanced by adaptively adjusting the link weights, thereby optimizing the routing selection in SDWSN.

4. Implementation of Flow-Table Based Policy in Data Plane

SDWSN are characterized by the decoupled control and data planes. Although the energy-efficient mechanisms π_1 and π_2 require a high computational complexity in control plane, they are executed by the information processing module of data plane at a low computational complexity. Based on Γ in (4), π_1 and π_2 are mapped to the parameter vector ψ of the value-redundancy filter and the path matrix Π of the load-balancing routing mechanism, respectively. The value-redundancy filtering parameter vector $\psi = (\|\Psi_B\|_p, \omega_{\text{QoS}})_{2 \times 1}$, where Ψ_B is the value-redundancy filter threshold calculated in (4) for use in (5) to curb CW size. Π consists of the identities of nodes belonging to the optimal routing tree $\text{Tree}_B^{\text{optim}}$ found by π_2 . In data plane, the specific implementation of energy-efficient mechanisms π_1 and π_2 involves only lightweight vector product and numerical comparison, which are both low-complexity matching operations, following the corresponding rules in the “Match Field” (i.e., Rule (1) to Rule (4) in Figure 2) of flow table. Figure 2 shows the detailed implementation of flow table at each switch. Flow table contains a prioritized list of rules to instruct the corresponding actions. Particularly, task scheduling has the highest priority, followed by value matching (i.e., value-redundancy filtering), and path matching (i.e., load-balancing routing) has the lowest priority. Each input flow will be matched to the prioritized rules. When multiple rules match an input flow, the rule with the highest priority will be selected first to execute the corresponding action set. If no rule matches an input flow, then the switch will request the controller to update its flow table, and the default-action set will automatically forward the flow to CIM in the controller for developing new energy-efficient policies.

When the current flow matching process ends and the next flow arrives, if the newly arrived flow contains the same contents as the previous one, it will be considered as redundant. In this case, flow table does not need to be updated

for value-redundancy filtering or forwarding path. Therefore, cross-plane communications and task reconfiguration can be greatly reduced, thus improving the energy efficiency in application-oriented SDWSN. When the real-time status (e.g., QoS and throughput) of SDWSN notably changes, the value-redundancy filtering parameters and routing paths can be dynamically adjusted by ψ , thus improving the environmental adaptability SDWSN.

In the proposed SDWSN-RL prototype, the control traffic from the controller to the data plane (i.e., downstream traffic) contains Packet-Out, Modify-State (configuration), and Read-State (request or query); the control traffic from the data plane to the controller (i.e., upstream traffic) contains Packet-In and Read-State (reply or report). The control traffic flow can be described as follows. Once a source host generates a query message, the controller responds with a reply message if the source host and the destination host are on the same island. Otherwise, the controller drops this query message. When network status changes, Packet-In event will be triggered by a request message in the data plane. Each switch sends a reply message containing the switch status to the controller via a secure channel supported by SOF. Meanwhile, Modify-State configuration messages are exchanged between the controller and switches via the secure channel as well. A Packet-Out message is generated by PD module and sent to switch to validate an entry in flow table. If no response is returned within a specified time, the potentially invalid entry will be deleted. The amount of Packet-In/Out messages for handling requests grows with the number of switches in the network.

5. Experiment Results

We perform experiments to evaluate the performance of the proposed RL based SDWSN prototype (SDWSN-RL) for environmental monitoring applications. The network simulator NS2 [21] is used to build the experiment environment. The parameter values used for the experiment setup are given in Table 1. We adopt the event radius (ER) model [22] to simulate the impulsive traffic triggered by temporally and spatially correlated monitoring events in a disk area. Following the ER model, the monitoring area of SDWSN is divided into an event gathering region, a data relaying region, and a decision making region. The first two regions belong to data plane, and the third one belongs to control plane. The monitoring center, that is, BS, is placed at the top right corner of the monitoring area with the coordinate (128 m, 162 m). The event center is located at the coordinate (48 m, 82 m) inside the event gathering area. The arrival of events follows a Poisson distribution in the time domain. Note that all the experiment results in Figures 3–8 include the energy consumptions of both data-gathering and control traffic. In the energy consumption calculation, we consider the energy consumption $E_{\text{data-gathering}}$ for data-gathering (including data fusion and processing) and the energy consumption $E_{\text{control-traffic}}$ for control overhead (including the control traffic for configuration, request-query, and reply-report). The evaluation for energy consumption in

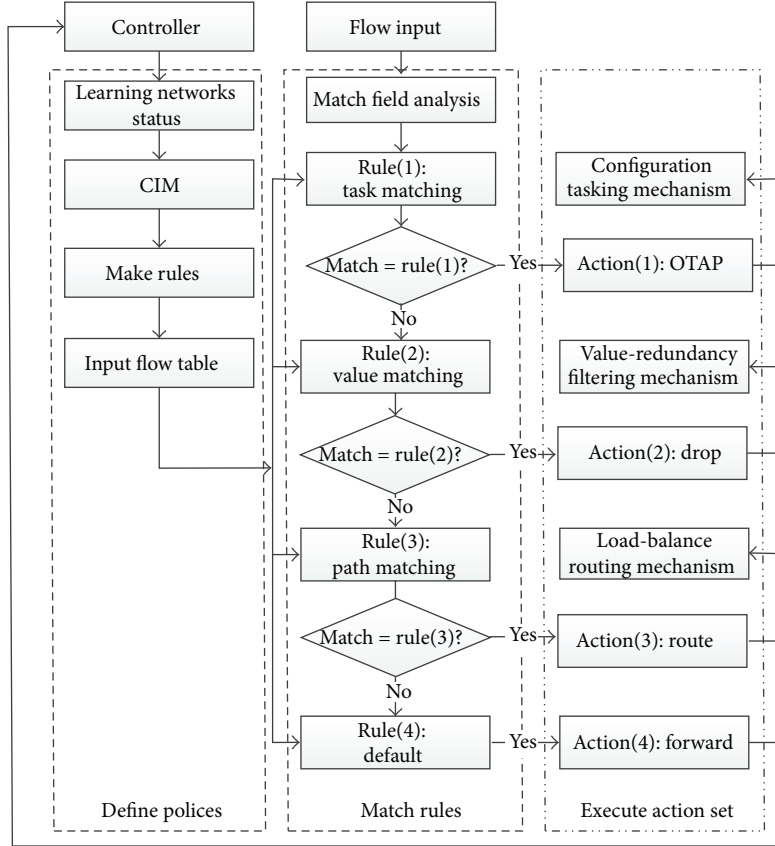


FIGURE 2: Implementation process of flow table.

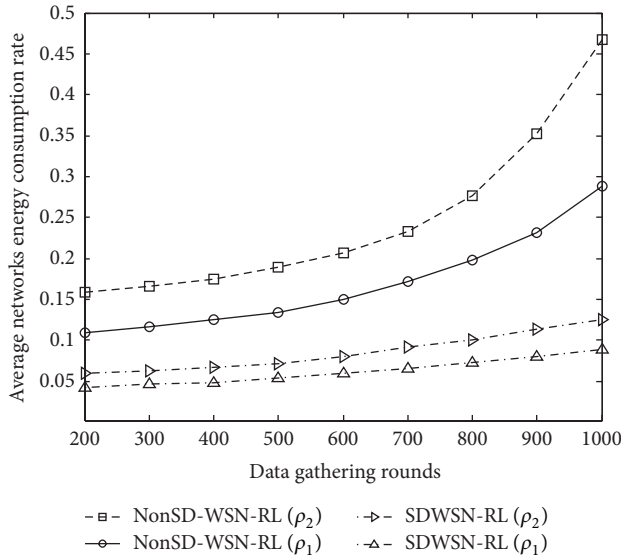


FIGURE 3: Energy consumption of WSN with and without software defined architecture.

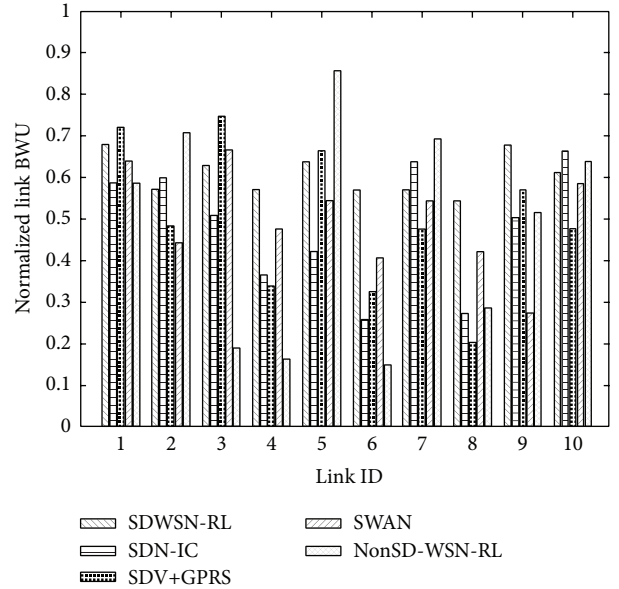


FIGURE 4: Comparison of load-balancing performance.

Figures 3, 4, 6, and 7 and for network lifetime in Figure 5 has all taken into account both $E_{\text{data-gathering}}$ and $E_{\text{control-traffic}}$.

Figure 3 shows the comparison of energy consumption rate between SDWSN-RL and a WSN without SDN

(called NonSD-WSN-RL). NonSD-WSN-RL is different from SDWSN-RL mainly in that there is no SDN architecture or SOF support. In NonSD-WSN-RL, each switch uses hybrid

TABLE 1: Experiment parameters.

Monitoring networks deployment		RL algorithm parameters		Communication parameters	
Application	ER model	Learning factor λ	0.5	MAC	802.11 edcf
Monitoring area radius	150 m	Discount factor γ	0.4	MAC frame	272 bits
Event area radius	80 m	QoS factor ω_{QoS}	(0, 1]	Frame interval	50 μ s
Mini detecting radius	10 m	Anti-interference threshold k_{th}	$[1, \infty)$	Payload	1500 bits
Max detecting radius	50 m	Tuning step ε	0.002	Carrier sense	30
Base station coordinates	(128 m, 162 m)	Tuning step ζ	0.018	PHY layer rate	46~512 kbit/s
Poisson arrival rate	5~15 packets/s	Round B	1~1800	Channel coding	16 bit
Number of nodes	90~580	CW initial size	10	PHY frame	128 bits
Deployment type	Random	Maximum back-off stage	7	Energy model	First-order
Antenna	Omni	FT upper limit	15	Transmit power	34.2 mW
Path loss exponent	2.4	m	[10, 80]	Receive power	22.1 mW
Network architecture	Homogeneous	Iteration threshold	2500	Sensitivity	20 dbm

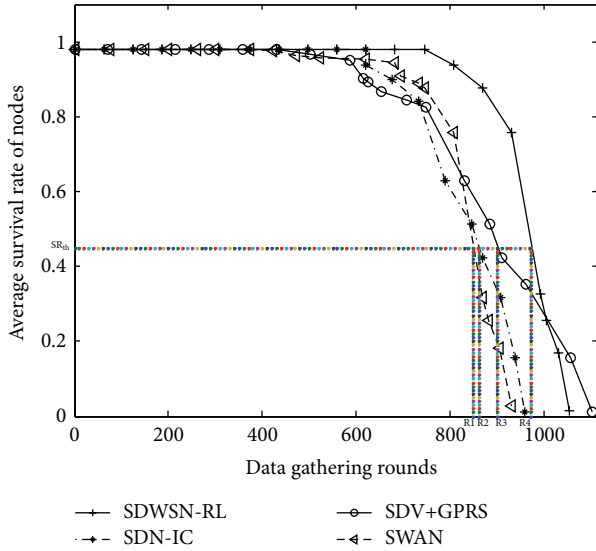


FIGURE 5: Network lifetime of SDWSN.

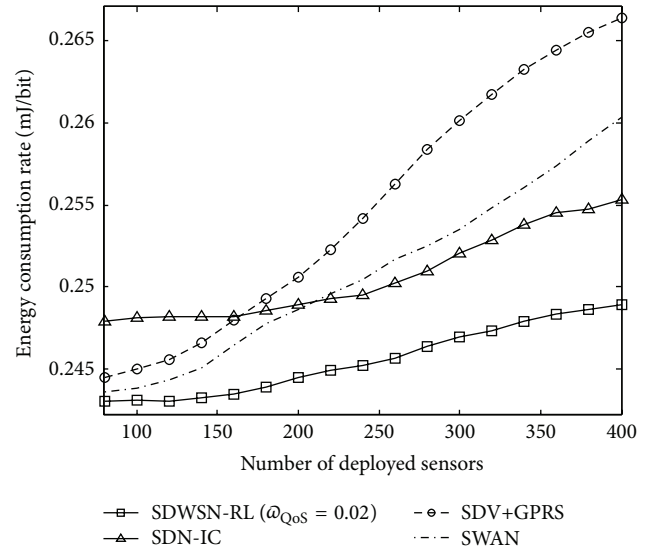


FIGURE 7: Comparison of energy consumption per bit.

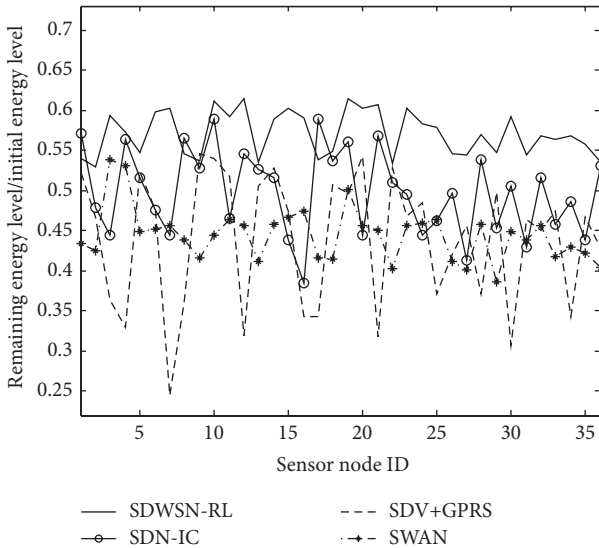


FIGURE 6: Comparison of node-level remaining energy.

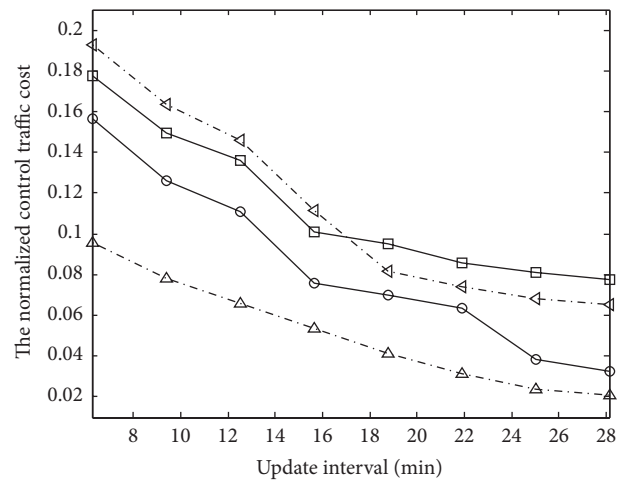


FIGURE 8: Comparison of control traffic cost.

energy-efficient distributed clustering routing (HEED) with a back-pressure mode [23], periodically computes the utility based on current queue gradients, and decides the next hop for each flow accordingly. The energy consumption rate is defined as the ratio of the energy consumption of SDWSN-RL (or NonSD-WSN-RL) to that of single-hop communication (without clustering or aggregation). Figure 3 shows that the average energy consumption rate of SDWSN-RL is much lower than that of NonSD-WSN-RL for each considered density of network nodes. The energy consumption rate of NonSD-WSN-RL increases faster with the increase of data-gathering rounds than SDWSN-RL. Furthermore, when the network node density increases from $\rho_1 = 0.7$ to $\rho_2 = 1.2\rho_1$, NonSD-WSN-RL has a much higher increase in energy consumption rate than SDWSN-RL. This is because the SDN-based controller can obtain real-time statistics on granular control and link status (which are not available in Non-SDN networks) for use in energy-efficient routing. Routing decisions made by CIM in SDWSN-RL are translated into rules and lightweight actions in flow tables to realize a floodless service discovery mechanism, thus limiting the amount of control messages. Without the support of SDN, routing tables are created (or reconstructed) in a collaborative way based on local exchanges of neighborhood information, which may require a lot of iterations before convergence.

Figure 4 shows the experiment results in terms of the normalized link BWU of 10 randomly selected links. For performance comparison with SDWSN-RL, we include three classic data-gathering schemes, SDN-IC, SDV+GPRS, and SWAN, which are content-centric, position-aided, and agent-based, respectively. We calculate the normalized variance of BWU (D_B) based on the 10 randomly selected links during data-gathering round B as follows:

$$\begin{aligned} D_B &= \|\text{var}(\mathbf{Y})\|_p \\ \text{s.t. } \text{var}(\mathbf{Y}) &= E\{[\mathbf{Y} - E(\mathbf{Y})]^2\}; \\ \mathbf{Y} &= \{e_B^{k_i}\}_i; \\ \forall i \in \mathbf{V}, k_i \in \mathbf{L}, p &= 2, \end{aligned} \quad (7)$$

where the 10×1 vector \mathbf{Y} contains the BWU of the 10 randomly selected links. Our calculations using the results in Figure 4 show that SDWSN-RL reduces the normalized variance of BWU by 8.9%, 12.8%, and 6.7% as compared to SDN-IC, SDV+GPRS, and SWSN, respectively, thus achieving improved load-balancing performance in SDWSN. The gap between the lowest and highest BWU across the 10 links of SDWSN-RL is 0.1357, which is the minimal among the five schemes. The results show that SDWSN-RL outperforms the other four schemes in terms of more balanced flow distribution in SDWSN by optimizing the weight of each link in the routing tree.

Figure 4 also includes NonSD-WSN-RL in the load-balancing performance comparison. We can see that the load-balancing performance of NonSD-WSN-RL is much worse than that of SDWSN-RL and the other SD based schemes. This is because the load-balancing routing mechanism in

SDWSN-RL utilizes global network information to construct optimal routing paths in a centralized manner. More specifically, SOF in SDWSN-RL provides a lightweight control protocol between the central controller and the switches in the data plane. The controller uses information in flow tables to calculate the load-balancing routes among all switches and sends the flow tables back to the switches to indicate the next hop towards each destination. SOF provides simple APIs at switches and allows the controller to program the switches through the APIs, which provide flexible lookup mode for deploying routing protocols. The SDN controller can obtain information about granular control, network topology, and link statistics, which is used in the centralized load-balancing routing, while such information is not available or difficult to obtain in traditional WSNs without SDN. NonSD-WSN-RL relies on a distributed neighbor discovery approach, which is not efficient in load balancing. Moreover, frequent next-hop neighbor discoveries and data packet forwarding based on distributed communications would lead to a sharp increase in control traffic with the increase of node density.

Figure 5 plots the average survival rate of nodes versus the number of data-gathering rounds for the four considered schemes. The survival rate of nodes in a network can be used to evaluate the total energy consumption of a data-gathering mechanism [24]. The lifetime of SDWSN is defined as the duration of normal network operations (e.g., data-gathering) while the survival rate of nodes is maintained above a threshold ($SR_{th} \in [0.4, 0.9]$). In Figure 4, we set $SR_{th} = 0.45$ and denote the network lifetime achieved by SWAN, SDN-IC, SDV+GPRS, and SDWSN-RL schemes as R1, R2, R3, and R4, respectively. We can see that $R1 < R2 < R3 < R4$, with SDWSN-RL achieving the highest average node survival rate among the four schemes. This is mainly because the value-redundancy filtering and load-balance routing of SDWSN-RL effectively inhibit the transmission of low-value flows and balance the distribution of loads across nodes, leading to a longer network lifetime of SDWSN as compared to other three schemes.

Figure 6 indicates the remaining energy level of a node after 80 data-gathering rounds normalized with respect to its initial energy level (which is fixed at 2 mJ for all nodes), for 36 different nodes randomly selected in an annular area centered at the coordinate (48 m, 82 m) (i.e., the event center) with the inner and outer radiuses of 10 meters and 20 meters, respectively. The experiment results show that for almost all the selected nodes, SDWSN-RL achieves the highest residual energy level among the four considered schemes. This would effectively prolong the lifetime of SDWSN. With each scheme, the normalized residual energy level varies across different nodes. A higher (lower) level of the remaining energy is due to the smaller (larger) amount of data flows that the node has forwarded. Compared with the three existing schemes, SDWSN-RL offers a more balanced distribution of energy consumption across the network nodes. This is mainly due to the proposed load-balancing routing mechanism, which utilizes global network information to construct optimal routing paths in a centralized manner.

Figure 7 plots the average energy consumed by the four considered schemes for forwarding a single bit of data

(mJ/bit) while meeting the same QoS requirement, versus the number of sensor nodes deployed in SDWSN. We can see that the energy consumption per bit of SDWSN-RL increases with the number of sensor nodes at a much slower rate than the three existing schemes, leading to a much lower energy consumption per bit of SDWSN-RL for large numbers of sensor nodes than the existing schemes. This is because the application-oriented in-network fusion in data plane of SDWSN-RL inhibits the transmission of value-redundant flows, and meanwhile the flow value determined by (1) is not notably affected by the number of traffic sources (i.e., nodes in the event gathering area), while the other three schemes would generate excessive traffic loads due to the large amount of local information exchange for executing distributed algorithms in data plane and the large amount of control overhead for cross-layer interaction, which degrade the energy efficiency especially for SDWSN with a large number of sensor nodes.

Figure 8 shows the comparison of control traffic cost between our proposed SDWSN-RL and the other three schemes (SWAN, SDV+GPRS, and SDN-IC), where the control traffic cost is defined as the ratio of control overhead to network throughput, and the network throughput is defined as the rate of successful bit delivery from the IG module to the monitor center. Figure 8 plots the normalized control traffic cost versus the time interval between two successive updates of the parameter r_{update} , which is defined as

$$\begin{aligned} r_{\text{update}} &= \|r_{\text{topology}}\|_p * \|r_{\text{event}}\|_p \\ \text{s.t. } r_{\text{topology}} &= \Delta \text{density}_{\text{nodes}}; \\ r_{\text{event}} &= \Delta \text{rate}_{\text{event arrive}}; \\ r_{\text{topology}} &\in [0.05, 0.1], \quad r_{\text{event}} \in [5, 15], \quad p = 2, \end{aligned} \quad (8)$$

where r_{topology} denotes the change in network node density and r_{event} represents the variation in Poisson event arrival rate. Since r_{topology} and r_{event} are the two major factors influencing the network status, the update interval of r_{update} indicates the frequency of network status change. When the WSNs status in monitoring applications frequently changes (e.g., the redistribution of data traffic caused by changes in topology or event arrival rate), the control overhead will increase due to frequent reconfiguration processes.

We can see from Figure 8 that the control traffic cost decreases with the increasing update interval of r_{update} for all the considered schemes. The control traffic cost of the proposed SDWSN-RL is much lower than the other three schemes for all considered values of r_{update} update interval. This is because the value-redundancy filtering in SDWSN-RL inhibits the transmission of value redundant loads, thereby avoiding generating control traffic between CIM and IM when the changes in network topology and/or event arrival rate do not significantly affect the monitoring data value. Routing decisions are made by CIM in the controller and then translated into rules and actions to be deployed in flow tables. Network APIs are used to configure flow tables for routing, in conjunction with a floodless service discovery mechanism. After the controller has configured routing, data packets can

be forwarded and processed in the data plane. Subsequent (follow-up) packets in a flow are forwarded in the data plane based on the configured routing in flow tables without any further participation from the control plane. Such reductions in control traffic free up radio resources for more data packets to be successfully delivered, thus further lowering the control traffic cost. However, the other three schemes (i.e., SWAN, SDN-IC, and SDV+GPRS) adopt broadcast based service discovery mechanisms, where distributed systems collaboratively create the routing table and the amount of control messages grows while network throughput declines with the increase of network node density or event arrival rate.

Since the throughput is inversely proportional to the control traffic cost for given control overhead, the results in Figure 8 also indicate that SDWSN-RL achieves the highest throughput among the four schemes considered, because it significantly reduces local control message exchanges, thereby freeing up radio resources for more data packets to be successfully delivered. The other three schemes (SWAN, SDN-IC, and SDV+GPRS) use broadcast-based service discovery mechanisms and distributed protocol, where switches need to wait for the wireless medium to be free to send their packets and many data packets may have to be dropped due to the wait, thereby limiting the throughput. Moreover, the broadcast based service discovery mechanisms require high volumes of control messages to be exchanged and high packet processing overhead.

6. Conclusion

In this paper, we have proposed a SOF-based SDWSN prototype for improving the energy efficiency and adaptability of WSNs in environmental monitoring applications, taking into account the inherent constraints of WSNs in terms of energy, radio resources and computational capabilities, and the distributed data flows of monitoring applications. Experiments results have shown that the proposed SDWSN prototype can greatly improve energy efficiency by effectively inhibiting the transmission of value-redundant loads, reducing the amount of cross-plane communications and enhancing the load balance in SDWSN.

In our future work, we will improve the scalability of control-plane mechanisms using decentralized coordination to overcome the bottleneck of a single logical controller and develop an adaptive anti-interference mechanism to improve the robustness of SDWSN for diverse monitoring applications in wireless environments with severe interference.

Conflict of Interests

The authors declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] B. Sun, L. U. Beaumont, and L. Osborne, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks,"

- IEEE Transactions on Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.
- [2] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 107–115, 2015.
 - [3] Y. Cui, S. Xiao, C. Liao, I. Stojmenovic, and M. Li, "Data centers as software defined networks: traffic redundancy elimination with wireless cards at routers," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2658–2672, 2013.
 - [4] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, "A survey of naming and routing in information-centric networks," *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012.
 - [5] C.-Y. Hong, S. Kandula, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven WAN," in *Proceedings of the ACM Conference on SIGCOMM*, pp. 15–26, ACM, Hong Kong, August 2013.
 - [6] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined VANET: architecture and services," in *Proceedings of the 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET '14)*, pp. 103–110, IEEE, Piran, Slovenia, June 2014.
 - [7] B. Karp and H. T. Kung, "GPSR: greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM 20'00)*, pp. 243–254, August 2000.
 - [8] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, "Energy minimization in multi-task software-defined sensor networks," *IEEE Transactions on Computers*, 2015.
 - [9] S. Li, S. Zhao, X. Wang, K. Zhang, and L. Li, "Adaptive and secure load-balancing routing protocol for service-oriented wireless sensor networks," *IEEE Systems Journal*, vol. 8, no. 3, pp. 858–867, 2014.
 - [10] C. Petrioli, M. Nati, P. Casari, M. Zorzi, and S. Basagni, "ALBA-R: load-balancing geographic routing around connectivity holes in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 529–539, 2014.
 - [11] S. Girgin, F. Polat, and R. Alhaji, "Positive impact of state similarity on reinforcement learning performance," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, pp. 1256–1270, 2007.
 - [12] J. Wang, J. Che, D. Sun, and J. Liang, "ARMA model identification using particle swarm optimization algorithm," in *Proceedings of the International Conference on Computer Science and Information Technology (ICCSIT '08)*, pp. 223–227, Singapore, September 2008.
 - [13] Y.-F. Lee and C.-C. Shen, "A transaction-based approach to over-the-air programming in wireless sensor networks," in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT '07)*, pp. 1377–1382, Sydney, Australia, October 2007.
 - [14] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor openflow: enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.
 - [15] H. Zhu and I. Chlamtac, "Performance analysis for IEEE 802.11e EDCF service differentiation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1779–1788, 2005.
 - [16] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, section 5.2, Society for Industrial & Applied Mathematics, 2000.
 - [17] H. Wang, K. Yao, and D. Estrin, "Information-theoretic approaches for sensor selection and placement in sensor networks for target localization and tracking," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 438–449, 2005.
 - [18] S. P. Meyn, "The policy iteration algorithm for average reward Markov decision processes with general state space," *IEEE Transactions on Automatic Control*, vol. 42, no. 12, pp. 1663–1680, 1997.
 - [19] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, 2010.
 - [20] R. S. Sudhaakar, S. Yoon, J. Zhao, and C. Qiao, "A novel QoS-aware MAC scheme using optimal retransmission for wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2230–2235, 2009.
 - [21] H. Zheng, Y. Zhao, and C. Chen, "Design and implementation of switches in network simulator (ns2)," in *Proceedings of the 1st International Conference on Innovative Computing, Information and Control (ICICIC '06)*, pp. 721–724, Beijing, China, September 2006.
 - [22] E. F. Nakamura and E. L. Souza, "Towards a flexible event-detection model for wireless sensor networks," in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC '10)*, pp. 459–462, June 2010.
 - [23] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 841–854, 2011.
 - [24] Y. He, I. Lee, and L. Guan, "Distributed algorithms for network lifetime maximization in wireless visual sensor networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, pp. 704–718, 2009.

