Situation-Aware Protocol Switching in Software-Defined Wireless Sensor Network Systems

Sudip Misra, *Senior Member, IEEE*, Samaresh Bera, *Graduate Student Member, IEEE*, Achuthananda M. P., Sankar K. Pal, *Fellow, IEEE*, and Mohammad S. Obaidat, *Fellow, IEEE*

***Abstract*—In this paper, a situation-aware protocol switching scheme is proposed for software-defined wireless sensor networks (SDWSNs) to support application-specific requirements in real-time. The proposed scheme consists of two phases — *decision making* and *protocol deployment*. In decision making, we use the supervised learning approach to choose the suitable routing protocols to be deployed in different time periods according to application-specific requirements. In the second phase, the chosen protocol is deployed in the network by the SDN controller in an adaptive manner. It is noteworthy that the proposed scheme can be integrated on top of the SDN controller in WSN to deploy a suitable routing protocol dynamically in the network. Extensive simulation results are analyzed to show the effectiveness of the proposed scheme, while varying the application-specific requirements. We see that the proposed scheme outperforms the existing schemes, in which a particular protocol is used in different time periods, in terms of energy consumption, throughput, packet delivery ratio, and delay in the network. It is shown that situation-aware protocol switching is capable of enhancing the network performance of SDWSNs.**

***Index Terms*—Network performance, software-defined networking (SDN), supervised learning, wireless sensor networks (WSNs).**

I. Introduction

WIRELESS sensor networks (WSNs) are widely used for military applications, environment monitoring, wild-habitat monitoring, target tracking, intelligent traffic monitoring, and energy management [1]. Consequently, multiple sensor nodes are deployed in a region to get real-time information. According to the received information, users are capable of taking adequate decisions for improved decision making. Recently, different mechanisms are introduced to change the activities of a sensor node in run-time. The software-defined networking (SDN) technology can be used in WSN to change the activities of sensor nodes in real-time to meet application-specific requirements [2].

Due to the growing interests of supporting application-specific requirements, it is required to manage the deployed nodes in WSNs dynamically in real-time from different aspects. For example, the AODV [3] routing protocol may be suitable for use on DSDV [4] in a specific time period for energy-efficient WSN applications. However, the latter one may be useful for minimizing network delay over the former one. Therefore, it is required to manage the routing protocols used in the network in different time periods depending on the requirements in order to get optimal network performance. However, the existing WSN frameworks [5] do not support such features to change the protocols in real-time. In contrast, SDN-enabled WSNs can be configured in real-time, while separating the control logic from the physical sensor devices [6]. Consequently, different application-specific requirements can be supported in real-time, which are platform-independent. However, there is a research lacuna on how to choose optimal routing protocols to get optimal network performances, and then how to deploy them in the network as well. The existing schemes either focused on the static requirements from the users or considered value-based information forwarding, i.e., the sensed information is forwarded if it crosses a predefined threshold value.

A. Contribution

To address the above mentioned issues, we propose a situation-aware protocol switching scheme in software-defined wireless sensor networks (SDWSNs). The proposed scheme consists of two phases—*determination of an appropriate routing protocol* and *deployment of the protocol at the nodes*. In the first phase, we determine the suitable routing protocol to be deployed for which network performance increases. To determine the protocol, we use supervised learning approach [7] at the controller end. The controller collects network statistics from the sensor nodes and application-specific requirements from application layer to take adequate decisions. In the second phase, the determined protocol is deployed at the individual sensor nodes. It is also noteworthy that multiple routin protocols can be used in a specific time period, as the software-defined framework supports protocol independent packet processing techniques [8]. Consequently, a WSN can

Manuscript received November 19, 2016; revised April 27, 2017 and October 6, 2017; accepted November 12, 2017. Date of publication December 14, 2017; date of current version August 23, 2018. The work of S. K. Pal was supported in part by the DAE R. Rammana and in part by the J. C. Bose National Fellowship. *(Corresponding author: Sudip Misra.)*

S. Misra, S. Bera and A. M. P. are with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India (e-mail: smisra@sit.iitkgp.ernet.in; s.bera.1989@ieee.org; achuthadivine@gmail.com).

S. K. Pal is with the Indian Statistical Institute, Kolkata 700108, India (e-mail: sankar@isical.ac.in).

M. S. Obaidat is with the Department of Computer and Information Science, Fordham University, New York NY 10458 USA (e-mail: mobaidat@fordham.edu).

Digital Object Identifier 10.1109/JSYST.2017.2774284

be divided into multiple subnetworks, and multiple routing protocols can be deployed depending on the application-specific requirements to get optimal network performance. Extensive simulation results show that the proposed scheme is useful to optimize network performance from the aspects of energy consumption, throughput, packet delivery ratio (PDR) and delay, while changing the routing protocols according to application-specific requirements. In brief, the *contributions* in this work are as follows.

1) We propose a situation-aware routing protocol switching scheme in SDWSN to meet application-specific requirements of users.
2) We interweave supervised learning-based algorithms for protocol selection and deployment, which train the SDN controller, so that it adaptively switches between routing protocols, as per the application-specific requirements. This contribution is a carefully artifacted rhetoric, which elicits the embedding of adaptive learning to improve the performance of SDWSN.
3) We present the framework for decision making and protocol deployment, which can be integrated with the existing SDWSN framework to improve overall network performance, without affecting the underlying architecture.

The rest of the paper is organized as follows. Section II discusses the existing works from the perspectives of WSN. Detailed system model is presented in Section III. Section IV describes the proposed solution approach. Simulation results are presented in Section V. Finally, Section VI concludes the paper with some future research directions.

II. Related Work

In this Section, we discuss the existing works from two different perspectives—reconfigurable WSN [9]–[12] and software-defined WSN [6], [13]–[17]—which are useful to change the activities of a sensor node in real-time.

A. Reconfigurable WSN

Bouabdallah *et al.* [9] proposed an energy consumption minimization scheme for sensor nodes deployed in a vehicular network. The authors claimed that energy consumption can be minimized by sending data traffic through multiple paths, rather than using a single path in the network. A load balancing approach is studied to determine the multiple paths, in order to minimize energy consumption. FPGA-based reconfigurable sensor nodes are developed by Krasteva *et al.* [10]. The developed systems consist of reconfigurable hardware platforms which can be configured in real-time, while introducing a middle-ware. Likewise, Guevara *et al.* [12] proposed a design for hardware-centric reconfigurable wireless sensor nodes. Additionally, transducer electronic data sheet architecture and management policy are proposed for the nodes. Gao and Piao [11] proposed a dynamic routing protocol deployment strategy for WSNs in real-time. In such a scheme, the use of the protocols for information routing can be changed in real-time, depending on the requirements and network conditions.

Although different useful schemes are proposed to configure activities of sensor nodes in real-time, they are either distributed in nature or constrained by their capacity. Due to the distributed nature of decision making, the existing solutions may not be adequate in a global scenario to meet application-specific requirements.

B. Software-Defined WSN

Luo *et al.* [6] proposed flow-table implementation rules, named as *Sensor-OpenFlow*, for use in sensor networks. Two different flow-table rules are proposed—value-based and ID-based. In the value-based approach, the value of the sensed information is compared before forwarding it to other nodes in the network. On the other hand, in the ID-based approach, the ID of sensor node is compared to forward the sensed information to sink nodes in the network. Galluccio *et al.* [13] designed a prototype for SDWSN, in which sensor nodes can be reconfigured in a stateful manner, while reducing the message exchange between the node and the controller. Anadiotis *et al.* [14] proposed an SDN-enabled WSN framework in order to deploy map-reduce functions optimally in the network. In such a scheme, the desired map and reduce functions are deployed at individual sensor nodes in the network using the SDN concept. Zeng *et al.* [15] proposed an energy consumption minimization scheme in WSN, in which a node consists of multiple sensor devices to perform different tasks. Therefore, the sensors can be activated according to the application-specific requirements. The authors proposed a hierarchical controller/manager architecture for the proposed scheme. Similarly, an SDN-based WSN architecture is proposed by Wang *et al.* [17] to control sleep-scheduling of sensor nodes in the network in an energy-efficient manner. Recently, Bera *et al.* [16] developed a platform, named as *Soft-WSN*, for controlling and monitoring WSN using the concept of SDN. The proposed system provides facility to control device-specific and network-specific tasks in a WSN. The authors claimed that the proposed system can be integrated into an existing WSN, without affecting underlying technologies.

We *synthesize* that most of the existing SDN-based schemes focused on the device-specific reconfiguration, which can be done in real-time to meet application-specific requirements. However, suitable information routing strategy also plays an important role to maximize the network performance, such as minimization of energy consumption and delay, and maximization of throughput and PDR. Therefore, we intend to propose a scheme to select and deploy suitable routing protocol, in order to maximize network performance.

III. System Model

A. Architecture

Fig. 1 presents a schematic architecture of SDN-enabled WSN. We follow the generalized architecture of SDN, which consists of infrastructure, control, and application layers. At the infrastructure layer, sensor nodes and access devices (ADs) are deployed. The sensor nodes send the sensed information to the ADs, and the ADs forward the information to the data center for computation. Leveraging the SDN concept in WSN, the sensor nodes can be controlled in a centralized manner, depending

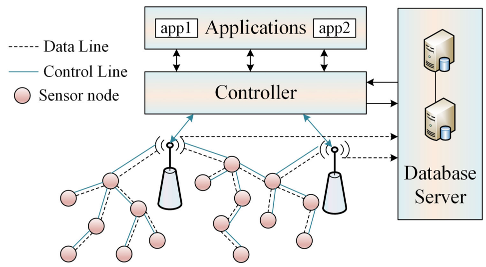

Fig. 1. Schematic architecture of SDWSN.

TABLE I
LIST OF SYMBOLS

Symbol	Description
E^{rx}_{ckt}	Circuitry energy consumption due to reception
E^{tx}_{ckt}	Circuitry energy consumption due to transmission
$D^{\text{proc}}_{i,k}$	Processing delay of node i at kth hop
$D^{q}_{i,k}$	queuing delay of node i at kth hop
$D^{tx}_{i,k}$	Transmission delay of node i at kth hop
D^{prop}_{k}	Propagation delay at kth hop
Υ	network throughput
$W_{i,k}$	TCP window size at kth hop
ρ	Packet delivery ratio
$\mathcal{U}_r$	Network cost with routing protocol $r \in R$
T_{elsp}	Current elapsed time
T_{itr}	Time interval to update network status
T	Total time
R	Set of routing protocols

on the application-specific requirements. Therefore, the controller takes adequate decisions and controls the entire network. On the other hand, application-specific requirements are provided to the controller from application layer.

B. Objectives

The objective of the proposed scheme is to maximize network performance. Therefore, we consider four metrics to form the objective function—energy consumption, PDR, throughput, and delay, which are discussed below.

1) Energy Consumption: Energy consumption at a sensor node depends on the required energy for transmission and reception. In addition to the circuitry energy consumption in transmission and reception, the transmission energy also depends on the distance between sender and receiver. Therefore, total energy consumption of a sensor node for transmitting data to a neighbor node located at a distance d is calculated as $E_i = E^{rx}_{\text{ckt}} + E^{tx}_{\text{ckt}} + \frac{\epsilon}{\eta} d^{\sigma}$ [18]. ϵ is a constant, and η denotes the drain efficiency. The parameter d is the distance between the sending and receiving nodes, and σ is the path loss exponent. For simplicity, we consider that the path loss exponent is always constant. Consequently, energy consumption for a given pair of source and destination nodes (consider it as a path l) located at h multihop distance is calculated as follows:

$$E(l) = \sum_{k=1}^{h-1} E^{rx}_{k,\text{ckt}} + \sum_{k=1}^{h} \left(E^{tx}_{k,\text{ckt}} + \frac{\epsilon}{\eta} d^{\sigma}_{h} \right) \tag{1}$$

$(h-1)$ hops are considered for energy consumption due to reception, as destination node is typically powered by external source. Consequently, the objective of the controller is to minimize the energy consumption for all pairs of source and destination nodes in the network, while deploying the suitable routing protocol $r \in R$. Mathematically

$$\text{Minimize} \sum_{t=1}^{T} \sum_{l=1}^{L} E(l,t,r), \; r \in R$$
$$\text{subject to } E^{tx}_{\text{ckt}}, E^{rx}_{\text{ckt}}, d > 0, \text{ and } \eta \leq 1. \tag{2}$$

Equation (2) denotes that E^{tx}_{ckt}, E^{rx}_{ckt}, and d are always greater than zero. On the other hand, η is always less than or equal to 1, as the maximum drainage efficiency is 100%. L and T denote the total number of paths used for routing in the network and total time, respectively. r is the routing protocol used at the node.

2) Packet Delivery Ratio: Packet delivery ratio is calculated as the ratio between the number of packets successfully received (P_{rx}) at the destination nodes and the number of packets transmitted (P_{tr}) at the source nodes, i.e., $\rho = P_{rx}/P_{tr}$. The objective is to maximize the PDR in order to improve the network performance. Mathematically

$$\text{Maximize} \quad \sum_{t=1}^{T} \rho(t,r), \; r \in R \; \text{ subject to} P_{tr} > 0. \tag{3}$$

The constraint $P_{tr} > 0$ confirms that the number of packets transmitted P_{tr} is always greater than zero.

3) Throughput: We calculate the network throughput for a path l as follows: $\Upsilon(l) = \sum_{k=1}^{h} \sum_{i=1}^{Ch} \frac{W_{i,k}}{RTT_{i,k}}/h$ [19]. Our objective is to maximize the throughput to improve the network performance. Mathematically

$$\text{Maximize} \sum_{t=1}^{T} \sum_{l=1}^{L} \Upsilon(l,t,r), \; r \in R$$
$$\text{subject to} W_{i,k} > 0, \text{ and } RTT_{i,k} > 0. \tag{4}$$

where Ch is the total number of available channels. $W_{i,k}$ is the received TCP window size, and $RTT_{i,k}$ is the round-trip time for ith channel in kth hop. It is noteworthy that performance of TCP in WSN is very poor. Therefore, we adopt the distributed TCP caching mechanism, in which the sensor nodes locally retransmit the lost segments [19]. Consequently, the lost segment is not retransmitted from the original source node, so that the required network performance is preserved.

4) Delay: We consider the delay as the combination of processing, queuing, transmission, and propagation delay. Therefore, total delay in a path between the given source and destination having h hops is calculated as: $\mathcal{D}(l) = \sum_{k=1}^{h} \left(D^{\text{proc}}_{i,k} + D^{q}_{i,k} + D^{tx}_{i,k} + D^{\text{prop}}_{k} \right)$. Consequently, the objective is to minimize the network delay for all paths L used for routing, which is represented as follows:

$$\text{Minimize} \sum_{t=1}^{T} \sum_{l=1}^{L} \mathcal{D}(l,t,r),\ r \in R$$
$$\text{subject to} D_{i,k}^{\text{proc}}, D_{i,k}^{q}, D_{i,k}^{tx}, D_{k}^{\text{prop}} > 0. \tag{5}$$

The constraint in (5) denotes that all the delays are always greater than zero.

In the proposed scheme, we consider that there always exists a path between a source and a destination. We do not focus on the connection establishment problem in sensor network, as main objective of the proposed scheme is to determine an appropriate routing protocol to be deployed in the network, so that overall network performance increases.

C. Overall Objective Functions

We combine all the individual objective functions together as a multiobjective function. We consider weight-based approach to form the multi-objective function. Mathematically

$$\text{Minimize}$$
$$\sum_{t=1}^{T} \left(\sum_{l=1}^{L} \Big(\zeta\mathcal{D}(l,t,r) + \alpha E(l,t,r) - \beta\Upsilon(l,t,r) \Big) - \gamma\rho(t,r) \right)$$
$$\text{subject to} (2),\ (3),\ (4),\ \text{and}\ (5),$$
$$\alpha + \beta + \gamma + \zeta = 1,\ \text{and}\ r \in R. \tag{6}$$

where α, β, γ, and ζ are the coefficients used to consider different weights of the individual objective functions. The values of the coefficients are determined based on the application-specific requirements of the users, and summation of all the coefficients is equal to unity. In the proposed framework, the sensor nodes are mobile in nature, and also resource constrained. Therefore, the distance between two nodes changes over time, which is considered to be one of the constraints. Then, the window size and round-trip time also change over time due to the changes in the distance between nodes. Further, propagation, transmission, queuing, and processing delay cannot be less than zero. Consequently, they are also considered as the constraints in the optimization problem. It is noteworthy that overall optimization is done over the routing protocol deployed at the sensor nodes. Therefore, the SDN controller deploys the appropriate routing protocol for which the value of objective function is minimized.

IV. Solution Approach

The proposed solution approach consists of two phases—(a) the determination of a suitable routing protocol, and (b) the deployment of the protocol in the network. Fig. 2 presents the overview of the proposed scheme. It is noteworthy that the proposed decision making process can be integrated on top of the SDN controller. Therefore, the proposed scheme can also be deployed in SDWSN, in order to improve the network performance. We propose a feature extraction and classification approach at the SDN controller end to determine an appropriate routing protocol, in order to get optimized network performance. The presented classification approach determines the

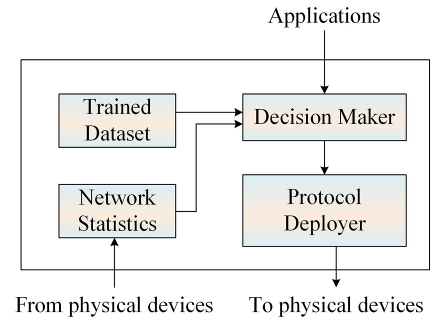

Fig. 2. Overview of the proposed scheme.

routing protocol considering the features, while depending on the application-specific requirements such as throughput, delay, energy consumption, and PDR, as presented in the Section III. Therefore, we consider the application-specific requirements in the form of energy consumption, PDR, throughput, and delay associated in the routing process.

Assumption 1: SDN controller periodically collects network statistics, such as network connectivity, node energy, PDR, delay, and throughput. Therefore, the existing software-defined WSN (SDWSN) platforms can be used to collect network statistics. Consequently, the proposed scheme does not add any additional overhead to the system for network statistics collection.

A. Determination of Suitable Protocol

We design a decision making scheme for determining the suitability of a routing protocol to be used to get optimal network performance. For this purpose, we use supervised learning approach [20] to train the controller through which the SDN controller can take adequate decisions with given conditions, such as the number of nodes, available energy, node speed, and application-specific requirements. Further, the training phase consists of three phases—feature extraction, classification, and selection of the best classifier.

1) Feature Extraction: We consider different network-specific parameters, such as network size, pause time, node speed, communication range, and packet sending rate, to extract various features from different network statistics. All the parameters are elaborated below.

1) Network Size: It is defined as the number of nodes present in the network. Typically, in a WSN environment, the total energy consumption in the network increases with an increase in the number of nodes for information routing. Therefore, we consider network size as one of the important parameters to extract the features from network statistics.
2) Pause Time: It defines whether the network is static or dynamic. If the nodes maintain an equal pause time, the nodes are static in nature. Otherwise, the nodes are dynamic. Pause time is considered to deal with the nodes' movement patterns in order to take adequate decisions.
3) Node Speed: This indicates the speed of the nodes in the network. If the speed of nodes is high, then the network topology also changes very frequently. Consequently, flow

tables[1] are required to be updated frequently at individual nodes, which, in turn, maximizes the energy consumption. Therefore, we need to select a suitable routing protocol to deal with the node speed in the network.

4) Communication Range: It is used to calculate the neighbor lists of the nodes. We assume that the nodes are distributed in a uniform-random manner in the network, and initially, the network is connected.

5) Packet Sending Rate: This parameter indicates the number of packets sent from the source in the network at each time period. This also affects the amount of energy consumption and PDR in the network.

Therefore, the above mentioned parameters are used to extract the features in order to classify the network statistics. The controller collects the network statistics consisting of different tuples as follows: <*Pause Time*, *Speed*, *Energy*, *Communication Range*, *Packet Sending Rate*>. The nodes periodically send these information to the controller.

2) Classification: After extracting different features, as mentioned in Section IV-A1, they are classified based on the optimization problem defined in Section III-C. We calculate a cost value for which the network performance is optimized, while application-specific requirements are given.[2] Therefore, for a given application requirements, we assign different weights of energy, throughput, delay, and PDR, as mentioned in Section III-C. Mathematically, it is represented as follows [21]:

$$h_\theta(x) = \sum_{i=1}^{k} \theta_i x_i \tag{7}$$

where θ_i denotes the weights considered for the application-specific requirements. k denotes the number of objective functions considered. In the proposed scheme, we have considered $k = 4$, i.e., energy, PDR, throughput, and delay. Accordingly, weight for θ_i is assigned, which is further reflected in the results and discussion (please refer to the Section V). It is noteworthy that θ_i s are the *parameters* (also known as *weights*) parameterizing the space of linear functions mapping from X to Y [21]. Therefore, Y is a linear function of X with coefficient θ_i. In order to learn the values of θ_i, the objective of the learner is to make $h_\theta(x)$ as much as possible accurate to meet the application-specific requirement. Therefore, (7) is represented as follows:

$$\text{Minimize } \mathcal{C}(\theta) = 1/2 \sum_{i=1}^{k} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2. \tag{8}$$

It is noteworthy that x stands for different parameters considered in the work, such as energy, delay, PDR, and throughput. y is the target value based on cost calculated using (8) for a particular routing protocol. Therefore, for given values of x, the objective is to minimize the target value y. Thus, the learner determines which routing protocol minimizes y for given x.

[1]Flow-tables are maintained at SDN-enabled devices to route information.

[2]It is noteworthy that different weights defined in Section III-C are used to consider different requirements such as energy consumption, delay, throughput and PDR. For example, if the user gives more priority on less energy consumption, the value of ζ is high. In a similar manner, different application-specific requirements are taken into account in the proposed scheme.

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY WITH DIFFERENT CLASSIFIERS

Name of the classifier	Percentage of accuracy
Naive Bayes	0.782
KNN(K = 10)	0.894
Random Forest	0.898
Best Fit Decision Tree	0.880
Classification via Regression	0.903
Function tree	0.854
Decision Table	0.901

Algorithm 1: Algorithm for classification.

Input: Set of routing protocols R, Network statistics
Output: Classify the features and stored them in the Class $\mathcal{C}$

Assign $\mathcal{U}_{min} = \infty$, $r = 1$, $\mathcal{C} = 0$;
while $r \leq |R|$ **do**
  $\mathcal{U}_r = \zeta\mathcal{D}(l,t,r) + \alpha E(l,t,r) - \beta\Upsilon(l,t,r) - \gamma\rho(t,r)$;
  if $\mathcal{U}_r \leq \mathcal{U}_{min}$ **then**
    $\mathcal{U}_{min} = \mathcal{U}_r$;
    $\mathcal{C} = r$;
  $r = r + 1$;
return $\mathcal{C}$;

Finally, value of θ_i is determined as follows [21]:

$$\theta_j = \theta_j + \lambda \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)} \tag{9}$$

where θ_j is the initial value, and the process is repeated until the value of θ_j converges. The parameter λ is the learning rate. Consequently, we get the values for θ_j, in order to meet application-specific requirements according to the training dataset. We select the ‘Classification via Regression’ approach, as it gives the best network performance (refer to Table II). The Mcnema’s significance test [22] is conducted to validate the accuracy of the extracted features from the sensed data. For simplicity, we limit our discussion on the Mcnemar’s significance test, as our prime objective concerns the deployment of a suitable routing protocol in the network, depending on the application-specific requirements.

We present the algorithm for classification in Algorithm 1.

B. Protocol Deployment

After selecting a suitable routing protocol decided in Algorithm 1 to optimize the network performance, we need to deploy it in the network. Algorithm 2 presents the protocol deployment in the network in a periodic manner similar to the network statistics collection. The algorithm checks the classes obtained using Algorithm 1 for which cost is minimized with the given network condition. Finally, the controller deploys the desired protocol in the network. The periodic update interval T_{itr} depends on users’ requirements.

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Number of nodes	100
Communication Protocol	IEEE 802.15.4
Node speed	0–20 m/s
Deployment strategy	Uniform Random
Communication range	0–100 m
Simulation time	100 min
Traffic	CBR
Mobility model	Random-waypoint
α, β, γ, and ζ	0.15 and 0.35

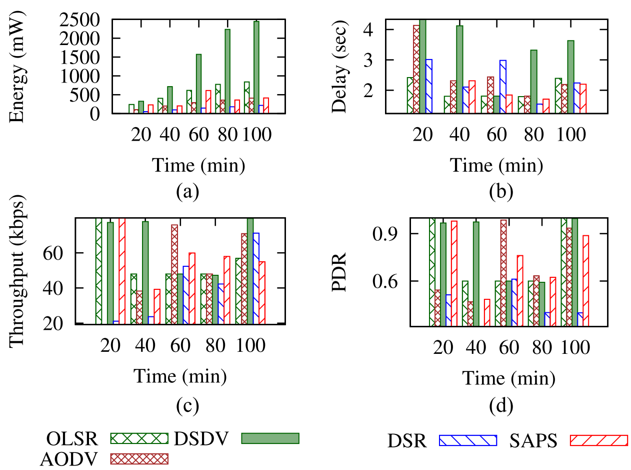

Fig. 3. Network performance with $\alpha = 0.15$, $\beta = 0.35$, $\gamma = 0.15$, and $\zeta = 0.35$. (a) Energy consumption. (b) Delay. (c) Throughput. (d) Packet delivery ratio.

Algorithm 2: Algorithm for appropriate protocol deployment.

Input: Set of classes $\mathcal{C}$, Time interval T_{itr}, Total time T
Output: Deployment the best routing protocol $r \in R$ in the network

Assign $r_{best} = 1$, Elapsed time $T_{elsp} = 0$;
Start the WSN with protocol r;
while $T_{elsp} <= T$ **do**
  if $T_{elsp} \% T_{itr} == 0$ **then**
    Collect network statistics $\mathbf{S}$;
    Choose the suitable protocol $r \in R$ with $\mathbf{S}$ from $\mathcal{C}$ obtained in Algorithm 1;
    if $r \neq r_{best}$ **then**
      $r_{best} = r$;
      Deploy the protocol r_{best} at individual nodes $i \in N$;
  else
    $T_{elsp} = T_{elsp} + 1$;

V. PERFORMANCE EVALUATION

We evaluated the performance of the proposed scheme, SAPS, using simulator NS-3 (www.nsnam.org), in which, required modules are developed to change the routing protocols in run-time. We use the term 'SAPS' to denote the proposed scheme in the rest of the paper. Different simulation parameters

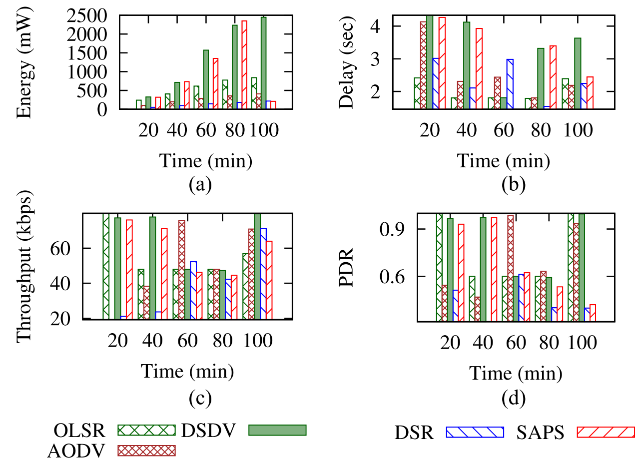

Fig. 4. Network performance with $\alpha = 0.15$, $\beta = 0.35$, $\gamma = 0.35$, and $\zeta = 0.15$. (a) Energy consumption. (b) Delay. (c) Throughput. (d) Packet delivery ratio.

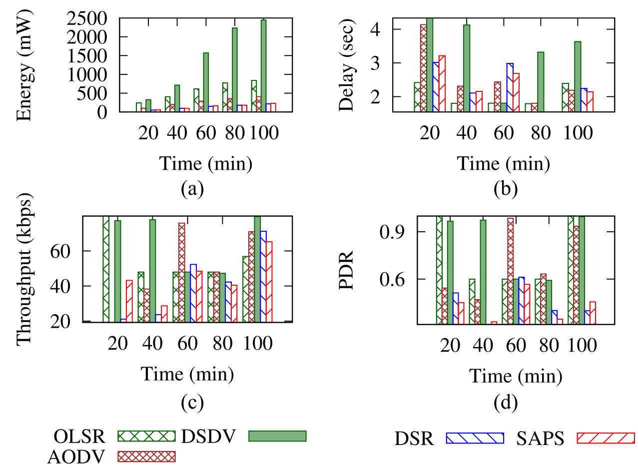

Fig. 5. Network performance with $\alpha = 0.35$, $\beta = 0.15$, $\gamma = 0.15$, and $\zeta = 0.35$. (a) Energy consumption. (b) Delay. (c) Throughput. (d) Packet delivery ratio.

are considered, as shown in Table III. We compare the performance of the proposed scheme with the following routing protocols: OLSR [23], DSDV [4], AODV [3], and DSR [24]. In SAPS, multiple routing protocols are deployed based on the decision taken by the controller in order to improve the network performance, while considering application-specific requirements. On the other hand, we present the results for benchmark schemes, while considering that a particular routing protocol is used in the entire simulation time. The controller dynamically checks for the suitability of a routing protocol to be deployed for which the network performance is optimized according to the extracted features from network statistics and classification, as described in Section IV. After determining the suitability of the routing protocol, the controller deploys it in the network. Consequently, we present three sets of results, which present energy consumption, throughput, PDR, and delay in the network with different weights to consider application-specific requirements, as presented in subsequent Sections V-A–V-C. The network status collection is done in a periodic manner, which is proactive in nature, as mentioned in Algorithm 2. However, this can also be

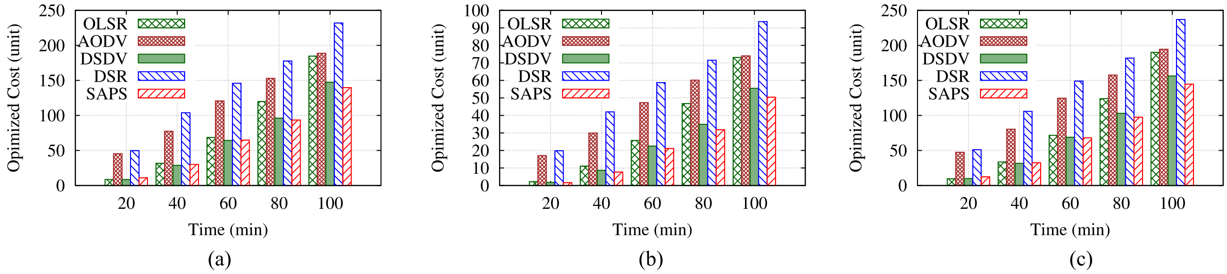

Fig. 6. Overall cost with different application-specific requirements. (a) $\alpha = 0.15$, $\beta = 0.35$, $\gamma = 0.15$, and $\zeta = 0.35$. (b) $\alpha = 0.15$, $\beta = 0.35$, $\gamma = 0.35$, and $\zeta = 0.15$. (c) $\alpha = 0.35$, $\beta = 0.15$, $\gamma = 0.15$, and $\zeta = 0.35$.

done in a reactive manner, i.e., if there is any sudden change in application-specific requirements, network status collection can also be done upon request without waiting for the next periodic update. Based on the application-specific requirements (i.e., the values of α, β, γ, and ζ) and the network conditions (i.e., network size, node-speed, communication range, packet sending rate, and pause time), the controller decides the protocol which is appropriate to be deployed. After deciding on the protocol to be deployed, the controller deploys it at the nodes in the network. Consequently, the nodes in the network route data based on the deployed protocol. There always exists a trade-off in deciding the value of this status collection interval. If the interval is large, the SDN controller misses many things happened in the physical network in real-time. On the other hand, if the interval is small, control message overhead in the network increases. Consequently, it depends on the network-specific requirements and deployment of the network. This is a well-known problem in SDN-based approaches [25]. In simulation experiments, we set the value for each period as 10 min. However, it can be changed according to the users' requirements.

A. Application 1: Priority on Throughput and Delay

We consider an application in which maximization of throughput and minimization of delay are important, while considering other constraints such as energy consumption and PDR. Therefore, we set the values for coefficients α, β, γ, and ζ as 0.15, 0.35, 0.15, and 0.35, respectively. Fig. 3 presents the results with desired application-specific requirements. We see that network throughput is maximized using SAPS over some of the existing protocols such as AODV and DSR, as shown in Fig. 3(c). On the other hand, it is almost equal with OLSR and DSDV, due to the proactive nature of the schemes. Moreover, network delay is minimized using SAPS over the existing schemes, while changing the routing protocols dynamically depending on the application-specific requirements, as shown in Fig. 3(b). On the other hand, compared to the existing schemes, energy consumption and PDR are moderate using the proposed scheme. Therefore, it is evident that the proposed scheme is capable of deploying suitable routing protocols in the network according to the application-specific requirements.

Fig. 6(a) depicts the results after solving the overall optimization problem, as presented in Section III-C. We see that the proposed scheme, SAPS, is capable of minimizing the cost compared to the fixed routing strategies.

In contrast, if we use a particular protocol in the entire simulation time, we see that network performance is not optimized, and it does not support the application-specific requirements as well. For example, using OLSR and DSDV, we can maximize the throughput, while incurring increased network delay, which is not sufficient to meet users' requirements.

B. Application 2: Priority on Throughput and PDR

In this scenario, network throughput and PDR are prioritized over energy consumption and network delay. Therefore, different values of the coefficients are considered as $\alpha = 0.15$, $\beta = 0.35$, $\gamma = 0.35$, and $\zeta = 0.15$. Fig. 4 presents the obtained results for energy consumption, throughput, PDR, and delay in the network. We see that SAPS is more capable of enhancing the network performance over the existing schemes, i.e., maximization of throughput and PDR in the network.

In this scenario, we see that the value of PDR increases using the proposed scheme, unlike Fig. 3(d), as it is prioritized. Similarly, network throughput is also maximized, as shown in Fig. 4(c). On the contrary, energy consumption and delay are not optimized, as our main focus is to maximize the network throughput and PDR in the network. Consequently, the proposed scheme maximizes the network performance, depending on different application-specific requirements. Additionally, the overall cost is also minimized using the proposed scheme, as depicted in Fig. 6(b).

C. Application 3: Priority on Energy Consumption and Delay

In the third scenario, we prioritize energy consumption and delay over throughput and PDR in the network. Therefore, we select the following values of the coefficients: $\alpha = 0.35$, $\beta = 0.15$, $\gamma = 0.15$, and $\zeta = 0.35$. Fig. 5 shows the results obtained corresponding to different performance metrics. As in other scenarios, the proposed SDN controller takes adequate decisions to deploy different routing protocols in different time periods. From Fig. 5(a) and (b), we observe that the energy consumption and delay in the network are minimized using the proposed scheme. Additionally, we get moderate results for throughput and PDR. On the other hand, although the use of AODV and DSR minimizes the energy consumption, the network delay is not optimized. Therefore, we see that the use of a particular protocol does not serve the purposes of application-specific requirements.

Fig. 6(c) depicts that the proposed scheme is also capable of minimizing the overall cost compared to the fixed routing strategies.

Consequently, it is evident that the proposed scheme outperforms the existing schemes from different sensor networking perspectives, such as energy consumption, throughput, PDR, and delay, as presented in Figs. 3–5. Additionally, it is also capable of minimizing the overall cost in the network as depicted in Fig. 6(a)–(c), according to the optimization problem presented in Section III-C.

VI. Conclusion

In this paper, we proposed a situation-aware protocol switching scheme in SDWSN to optimize network performance, while considering different application-specific requirements. We designed an adaptive controller to take appropriate decisions based on the network condition and application-specific requirements. To take adequate decisions, we used a supervised learning approach at the controller end. Finally, the decided protocol is deployed in the network in real-time. We evaluated the performance of the proposed scheme under different application scenarios, and showed that the proposed scheme is capable of enhancing network performance over the existing schemes, in which a particular routing protocol is deployed for all the time.

In this work, we observed that it takes some time to deploy the updated routing protocol at each sensor node. Therefore, during the switching phase, few packets are unnecessarily retransmitted and may be lost, which, in turn, minimizes PDR in the network. We plan to address this issue as a future extension of this work. Further, due to the movement of the sensor nodes, there always exists a gap in status reporting to the controller from the physical nodes. Consequently, the controller does not have real-time information due to the reporting delay and changes in the network. This is a limitation of an SDN-based approaches. We also plan to address this issue as a future extension of this work. Additionally, discussion on control overhead in network status collection and corresponding results are also included as a future extension of this work.

References

[1] S. C. Misra, I. Woungang, and S. Misra, Eds., *Guide to Wireless Sensor Networks*. Berlin, Germany: Springer, 2009.

[2] T. Miyazaki *et al.*, “A software defined wireless sensor network,” in *Proc. IEEE CCNC*, Honolulu, HI, USA, Feb. 2014, pp. 847–852.

[3] C. Perkins and E. Royer, “Ad-hoc on demand distance vector routing,” in *Proc. IEEE Workshop Mobile Comput. Syst. Appl.*, 1999, pp. 90–100.

[4] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 1994, pp. 234–244.

[5] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[6] T. Luo, H.-P. Tan, and T. Q. S. Quek, “Sensor OpenFlow: Enabling software-defined wireless sensor networks,” *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012.

[7] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,”*in Machine Learning*. Berlin, Germany: Springer, 1997.

[8] P. Bosshart *et al.*, “P4: Programming protocol-independent packet processors,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014.

[9] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, “On balancing energy consumption in wireless sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 6, pp. 2909–2924, Jul. 2009.

[10] Y. Krasteva, J. Portilla, E. de la Torre, and T. Riesgo, “Embedded runtime reconfigurable nodes for wireless sensor networks applications,” *IEEE Sensors J.*, vol. 11, no. 9, pp. 1800–1810, Sep. 2011.

[11] S. Gao and Y. Piao, “DRRP: A dynamically reconfigurable routing protocol for WSN,” in *Proc. Int. Conf. Progress Inform. Comput.*, Shanghai, China, May 2014, pp. 460–465.

[12] J. A. Guevara, E. A. Vargas, A. F. Fatecha, and F. Barrero, “Dynamically reconfigurable WSN node basedon ISO/IEC/IEEE 21451 TEDS,” *IEEE Syst. J.*, vol. 15, no. 5, pp. 2567–2576, May 2015.

[13] L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, “SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks,” in *Proc. IEEE INFOCOM*, Kowloon, Apr./May 2015, pp. 513–521.

[14] A.-C. Anadiotis, G. Morabito, and S. Palazzo, “An SDN-assisted framework for optimal deployment of MapReduce functions in WSNs,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2165–2178, Sep. 2015.

[15] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, “Energy minimization in multi-task software-defined sensor networks,” *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3128–3139, Nov. 2015.

[16] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, “Soft-WSN: Software-defined WSN management system for IoT applications,” *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

[17] Y. Wang, H. Chen, X. Wu, and L. Shu, “An energy-efficient SDN based sleep scheduling algorithm for WSNs,” *J. Netw. Comput. Appl.*, vol. 59, pp. 39–45, Jan. 2016.

[18] Q. Wang, M. Hempstead, and W. Yang, “A realistic power consumption model for wireless sensor network devices,” in *Proc. IEEE Commun. Soc. Sensor Ad Hoc Commun. Netw.*, Reston, VA, USA, Sep. 2006, pp. 286–295.

[19] A. Dunkels, T. Voigt, J. Alonso, and H. Ritter, “Distributed TCP caching for wireless sensor networks,” in *Proc. 3rd Annu. Mediterranean Ad Hoc Netw. Workshop*, Jun. 2004, pp. 1–11.

[20] E. Alpaydin, *Introduction to Machine Learning, Second Edition*. Cambridge, MA, USA: MIT Press, 2009.

[21] A. Ng. CS229 Lecture notes - Supervised learning, pp. 1–30. [Online]. Available: http://cs229.stanford.edu/notes/cs229-notes1.pdf. Accessed on: Dec. 20, 2016.

[22] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[23] T. Clausen and P. Jacquet, “RFC 3626: Optimized link state routing protocol,” 2003.

[24] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” *in Mobile Computing*. Berlin, Germany: Springer, 1996, pp. 153–181.

[25] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, “PayLess: A low cost network monitoring framework for Software Defined Networks,” in *Proc. IEEE Netw. Oper. Manage. Symp.*, Krakow, Poland, 2014, pp. 1–9.

Authors’ photographs and biographies not available at the time of publication.