IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 7, 2018, accepted September 15, 2018, date of publication September 19, 2018, date of current version October 17, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2871271

SDQ-6WI: Software Defined Quadcopter-Six Wheeled IoT Sensor Architecture for Future Wind Turbine Placement

AMMAR K. AL MHDAWI AND HAMED S. AL-RAWESHIDY, (Senior Member, IEEE)
Wireless Networks and Communications Centre, Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London, London UB83PH, U.K.

Corresponding author: Ammar K. Al Mhdawi (eepgaka1@brunel.ac.uk)

ABSTRACT Although wind-generated power was estimated to be 4% of the entire world electricity usage, wind turbines are considered to be a growing technology with many experts considering new approaches to wind turbine design and farmland selection to increase the wind turbine output efficiency. When implementing a new wind turbine install on a farm, many concerns are taken into consideration such as environmental challenges and cost. Thus, the power of the wind turbine can be increased at least 10 times when the most efficient wind power location is selected before the wind turbine is placed. In this paper, we propose a novel software-defined quadcopter–6 wheeled industrial IoT (SDQ-6WI) architecture that is based on a developed quadcopter system to collect wind speed data from a mobile IoT vehicle base station that is based on a developed open flow (DOV) protocol operation. The mobile ground vehicle act as a wind speed measurement system that travels on a given set of waypoints to measure the best optimal wind speed quality and send the collected information to the quadcopter-based SDN controller then to the cloud for further processing. Our proposed system can handle a heterogeneous environment that lacks Wi-Fi and cellular coverage and uses the minimum total transmission power when sending data. The experiential results showed that the measured wind speed data could be collected in a time-efficient manner compared to a traditional process which is considered to be costly, time wasting, and non-effective. Our extensive testing showed that about 23.19% in power was reduced in the wind measurement process in comparison with the fixed sensor nodes. In essence, the proposed architecture help reduce the high cost of relocating wind turbines to an efficient location and increase the generated power by selecting the best optimal windy location.

INDEX TERMS Wind turbine, OF, SDN, control plane, power management, quadcopter, IoT, wind speed, iPRDR, SDQ-6WI.

I. INTRODUCTION

Software Defined Network (SDN) has been attracted by many researchers and network developers [2], [3] for its amazing key concept of operation. The idea of SDN is to separate control plane which is the brain of the network controller from the forwarding plane. The segregation process allow better and simple network construction and management. The control plane have global overview of the network elements, in addition, data plane is the forwarding capability that can perform data forwarding based on SDN rules that are preset in the configuration. The Open Flow (OF) protocol is the main protocol that is used between the SDN controller and the OF switch. Open Flow has been intended to be a promising approach to the future generation with easy programmable implementation. OF switches forward packet based on a matching mechanism that is implemented in the table. If a match occurred, then the packet is forwarded to the output interface, otherwise, traffic is sent up to the controller for further analysis. According to the performance process of the OF protocol, that major traffic requests can be classified in two sections: (1) packet-in request and (2) packet-out push. The packet-in is send to the controller to seek rules on how to manage the packet due to table-miss on the OF side. A packet-in request is sent by the OF switch to the controller due to match failed in the forwarding table. The SDN controller is to investigate what to do with this packet by calculating a new route based on the packet header information. Moreover, the SDN respond back to the OF switch with the packet and new rules to be installed in the forwarding table related to this packet. There are several implementations of open flow switches that can be used to run a testbed or virtualized environment [4]. SDN controller can handle large network environment easily without complications as configuration can pushed down to all open flow switches without the need to configure each node separately. Sensor networks [5] can be combined with SDN, but this may come with some power consumption challenges. Unmanned Aerial Vehicles (UAV) or Quadcopters [6] now days are getting a high portfolio as they are being used for a wide range of applications in the current industry. UAV are widely used mainly in heterogeneous environments where humans cannot reach. Moreover, UAV provided more opportunities to replace traditional systems such as disaster management and agricultural support. Networks that are based on UAV systems can provide resolutions to issues that current infrastructures cannot resolve. In our proposed mentioned approach, a new SDN implementation is based on quadcopter system. The proposed quadcopter will have the behavior of SDN controller by managing and controlling the IoT mobile OF switch which will lead to an efficient wind readings within a shorter period of time. Furthermore, by using single ground mobile node, the congestion on the controller will be reduced significantly which can lead to less power consumption in the SDQ quadcopter, and that eventually will increase hovering time of the SDQ which acts as a mobile control station to provide efficient network management and uplink/downlink communications to and from the 6WI mobile node. In comparison with the fixed traditional nodes, UAV can provide on-the-fly network management with open flow capabilities by establishing a line of sight communication with the sensor node with low transmit power according to an algorithm that we have developed to manage sensor readings according to specific way-points. Our algorithm provide efficient and precise power consumption management for the quadcopter and the 6WI sensor node. The main contribution of our study can be summed up as follows:

- We propose a new prototype system to find the best optimal location for strong winds in a selected wind farms before the wind turbine is placed. The prototype consist of Software Defined architecture that is based on a Quadcopter IoT system with 6 × wheeled (6WI) mobile node. The architecture consist of 4 × rotor quadcopter that operate as an SDN controller and a 6 × wheeled vehicle that is built as a mobile sensor node to collect wind speed measurement readings and send them back to the SDN Quadcopter. By using single 6WI node, power consumption can be reduced significantly by 23.1% instead of depending on a fixed sensor nodes that is used in a traditional process which is considered to be non-efficient and costly process. The motion feature of the wind speed mobile node is implemented via the SDQ controller based on a proposed (DOV) algorithm. We prove that our architecture works consistently better than current wind measurement system by comparing the install time, sensor deployment, cost and wind strength which can be significantly reduced using our proposed system. The collected data can be logged in areas that lack Wi-Fi or cellular coverage, or it can be uploaded to our previous proposed cloud infrastructure (iPRDR) [7] for further data processing.
- We modeled the power consumption of the SDQ using power optimization techniques. We developed the modeling techniques that are presented in [8] to accommodate with our system so that the power consumption is calculated based on a given set of parameters. The extension of our power consumption formulas based on calculating the sleep and wake times of the sensor node for the controller and the ground sensor node. In our calculations, we focused on the elements that are power-hungry which may cause future system failures. The used mathematical parameters are listed in the notion table with their corresponding description.
- We conducted extensive testing on the quadcopter in a vast farmland. The hardware used to build the quadcopter and the wheeled vehicle was off-the-shelf hardware. We observed that the wind strength optimal location was selected quickly and easily among a set of given waypoints.
- We developed an algorithm that is based on the open flow concept called Developed Open Flow (DOV) algorithm. Our algorithm manages node communications and controls active, and sleep sessions for sensor nodes as this will optimize power consumption levels.

The remainder of the manuscript is organized as follows. In Section I, an introduction is presented. In Section II, previous work is briefly reviewed. In Section III, an SDQ-6WI framework is described and presented. Section IV illustrates the power consumption formulation and control algorithm. In Section V, the system model and problem formulation presented. Section VI shows the algorithm formulation. In Section VII, field experiments and results. Finally, Section VIII is the conclusion of the work.

II. PREVIOUS WORK

Indeed, Unmanned Ariel Vehicles (UAV) or Quadcopters play a significant role in the internet of things (IoT). In general, IoT devices cannot handle communication over considerable distance due to power constraints. However, quadcopters can be implemented towards IoT by implementing them to collect sensor data in a mesh network type. Nonetheless, quadcopters play as an aggregator for the IoT nodes, but with all that being said, there are some technical challenges must be addressed such as power consumption and communication in the disconnected areas. Numerous research has been conducted by engineers to optimize the power consumption of a network system for efficient network management.

Park *et al.* [8] presented a quadcopter formation algorithm. The analysis of the quadcopter was based on the capacity allocation. The altitude and the coverage area relation was discussed concerning the exact location of each quadcopter. The algorithm proves to maximize the strength of the coverage infrastructure. However, the power consumption management was not discussed, as this type of algorithm requires a high power supply to maintain adequate hovering time. Chen *et al.* [9] proposed the Q-charge system, a wireless charging architecture based on quadcopter system which can supply power wirelessly to sensor platforms. The system consists of a three parts, energy transfer module, a programmable board, and cloud server. The remote energy transmission is done via the wireless charging platform on the quadcopter. According to the author, the platform has successfully implemented in multiple areas such as building structures, farmlands and proven to be active and efficient. Chiang *et al.* [10] developed a light detection and ranging (LiDAR) UAV system for disaster management and environment reconstruction. The author has introduced a new iterative closest point set algorithm to overcome the registration problems in the inhomogeneous points cloud sets. According to the author, LiDAR achieves meter-level precision and implements environment reconstruction with a dense point cloud. Mozaffari *et al.* [11] presented the efficient development of a UAV system was implemented using an aerial base station for data collection. A new framework has proposed for jointly optimizing 3D placement and mobility of UAV systems. The framework implemented with sensor IoT network in a time-varying mesh network where the ideal motion forms of the UAV as analyzed. The author provided that the transmission power of the IoT nodes be reduced to 45%. Besides, the system showed it has a maximum of 28% reliability enhanced. Azizian *et al.* [13] presents a new architecture for managing updates on vehicles while implementing SDN. With using SDN, a programmability interface and dynamic configuration were implemented via using a solution with different frequency bands that are assigned to different graphs to improve network performance. Ohsugi *et al.* [14] introduced a power consumption model of NDN-based multicore router controller. The paper discussed the power management of the NDN network. The developed power consumption model of a multicore software controller routing engine was to satisfy the following requirements which is that the model should reflect loads on a hardware platform and the consumed power should be a function of the load. The author concluded that cashing can drop power consumption in regards to the supplied load. Zhang *et al.* [15] proposed a cloud mobile video architecture based on SDN to improve user experience and reduce overhead associated cost. The author has developed an algorithm based on dual decomposition approach. The results collected shows that the approach can reduce the total cost and guarantee user experience. The above mentioned related work on quadcopter UAV systems with SDN and power consumption management are intended to enhance the performance of the network and reduce power consumption levels. Our study proposes a new SDN-quadcopter platform that can calculate the best optimal wind strength location for future wind turbine placement in farmland or heterogeneous environment.

III. FRAMEWORK ARCHITECTURE

The developed SDQ-6WI prototype as a solution for finding the most efficient strong wind location for future wind turbine placement. Our prototype can be differentiated into two implemented sections. They are described as Hardware Implementation and Software platform. The hardware implementation is focused on the design of the quadcopter and the wheeled vehicle, while the software part is the programmable application for the hardware.

A. QUADCOPTER PLATFORM

Our developed quadcopter system consists of the main frame which is made from a light wood material with four arms (each arm is 40cm in length) connected from the center. Additionally, Four DC brushless motor 2830S series attached at the end of each arm with propellers base and blades that operate in clockwise (CW) and counter-clockwise (CCW). A baseboard is placed in the center of the frame to hold the hardware components. Each motor is connected to an Electronic Speed Controller (ESC) to manage the speed of the blades in order to provide thrust lift. Additionally, Power Distribution Board (BDP) is used to provide power to other hardware modules that operate on different voltage levels. An Arduino Uno board [16] is programmed and implemented as an SDN controller along with the XBee module for RF transmission and receiving wind speed measurements. The motion of the quadcopter is generated from different directional elements of the rotors. The first is called the roll movement which represents the rotation of the quadcopter on the front and back of the axis. The second rotation is called the pitch, which represents the rotation between the side to side lateral axes. The final motion is called the yaw motion which moves the quadcopter in a clockwise or anti-clockwise rotation as it keeps side by side to the ground. The motions above are implemented via KK21.5 multi-rotor LCD flight control board with 6050MPU stability gyro system and AMega644PA which operate on 4.8-6.0 v with five-channel operation from the receiver. We used a wireless module with six-channel operation at specified MHz frequency per vendor specifications which is used to send motion directions to the quadcopter. The quadcopter power source is based on a 2200mAh 1.5C Turnigy LiPo Battery. Each rotor receives their current from the ESC which is fed via the PDB board. As far as current draw levels, each ESC consumes about 0.5 amp of current. The signal wires are used to send rotor directions to the motors from the KK2 board. Fig. 1. Shows a block diagram of the entire SDQ-6WI architecture. The main block is the flight controller block that contains the SDN controller with related RF components. The second block is the ground mobile station that represents the open flow switch which holds all the sensor data in the forwarding table. The block on the left represents the control station that is used to send new rules and modifications to the controller.

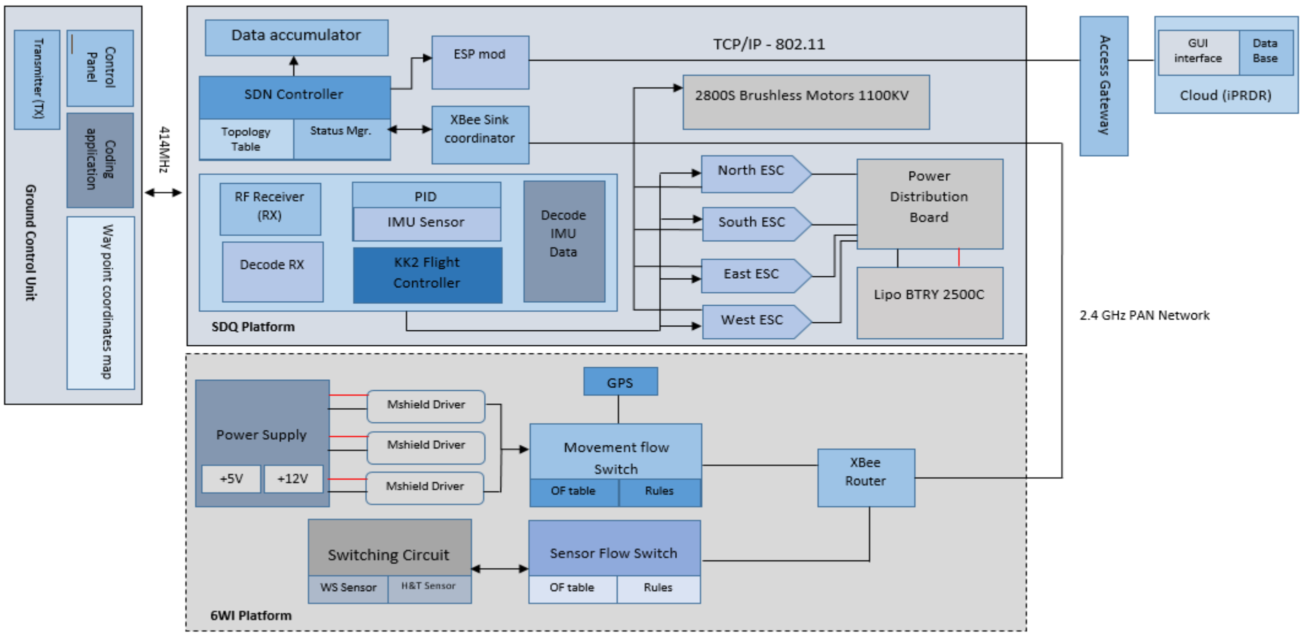

FIGURE 1. SDN Architecture overview.

B. WHEELED VEHICLE PLATFORM

For our proposed six-wheeled open flow vehicle, the hardware consisted of multiple electronic boards which we can illustrate as follows: The master board consists of a microcontroller which is based on the Arduino Uno board that manages two types of sensors. The first sensor is the wind speed sensor (Vortex) [17] which is a rugged sensor that can handle 5 to 125mph speed readings. It consists of three cup rotor pressed on a steel bar. The internal structure of the rotor consists of a reed switch and a magnet that provide one pulse per rotation. The formula to convert pulses to wind speed readings is that each 2.5mph per Hz (one Hz equal to one pulse /sec). The second sensor is the DHT11 temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to manage the surrounding air and send a digital signal to the Arduino corresponding pin. The sensor can operate on 3-5 volts with 2.5mA max current. Additionally, the sensor can manage 20-80% readings with 5% precision and 0-50C readings with +/− 2C accuracy. The humidity and temperature are considered to be critical climatic variables that affect wind turbines operation as the kinetic energy [18] in the wind depend on the density of the air. In other words, the denser the air, the more power is received by the wind turbine. The density decreases slightly with elevated humidity. Moreover, the air is denser when it is cold than when it is warm. The open flow switch is based on Arduino Uno board programmed to receive waypoint sensor location via XBee RF module. We have used six geared DC brushed motors (2008 rpm, 19.8W) with sleeved bearings. The motor movement direction is implemented via an L298N H-bridge. The L298 is a dual motor controller with a max current of 2A. The board operates on +12V, and we can source 5v from it to feed the OF switchboard.

The power source for the vehicle is implemented using a LiPo battery with 3000mAH 40C power that can provide high current to enable motor movements with up to 22.2 volts.

IV. DESIGN AND IMPLEMENTATION

To verify the effectiveness of our design, we have implemented everything experimentally using off-the-shelf hardware and IoT components. Fig. 2. Shows the quadcopter system with all related components. The SDN controller on the quadcopter can handle a vast number of switches while maintaining a high level of communication. The SDN controller can provide a rich set of programmable application interface (API) to implement modification and rule updates on the open flow switches. We used 3 × L298 bridges to accommodate 6 × motor operation. Moreover, L298 has 4 × input pins with 2× enable pins. The enable pins are used to control the speed of the motors which ranges from 0-255. The flight controller is based on the KK2 board which maintains the necessary algorithms for control and movement.

For our testbed, the SDN controller is implemented on the middle of the quadcopter that is connected to an XBee module to send rules and modifications to the open flow switch as illustrated in Fig. 3. The open flow switch performs forwarding movement based on a set of waypoints given by the controller. The second component of the testbed is the 6WI which consist of an open flow switch based on Arduino technology. The open flow switch checks its forwarding table and supplies specific signal to the six motors to move to a

FIGURE 2. Quadcopter overview overview.

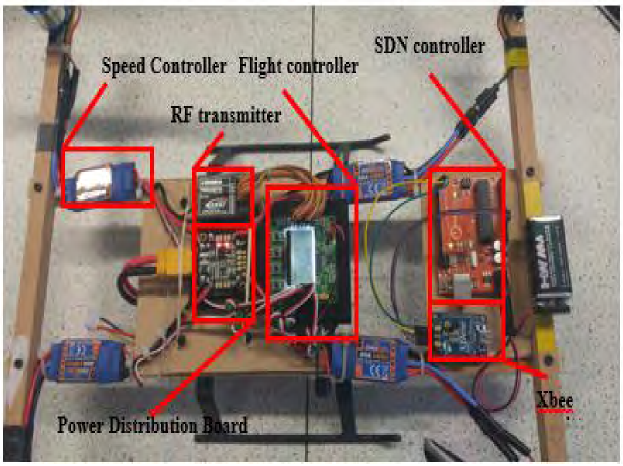

FIGURE 3. Proposed quadcopter hardware.

specific waypoint. However, if new rules need to be added, the SDN controller transmits new waypoints and overwrite the previous rules in the forwarding table. This process is called the mod_forward process. Fig. 4 demonstrates the experimental hardware design for the 6WI vehicle and its related components which consist of an anemometer, GPS, motorshiled and RF communication circuit with the open flow switch board. The anemometer provide an analog signal; however, the output of it has to be converted to a digital pulse signal with the voltage divider circuit to be fed to the open flow switch pin for accurate digital measurements.

As we described previously, our platform is based on two main parts. The first part is the quadcopter based SDN controller and the second part is based on IoT designed vehicle that implements specific wind measurement sensing. Additionally, both of the platforms can be seen in Fig. 5, which presents both developed electronic platforms. It is evident from our design that we are enabling real-world testing and analysis. Our quadcopter support cellular and non-cellular coverage and operates on an enhanced (DOV) algorithm that

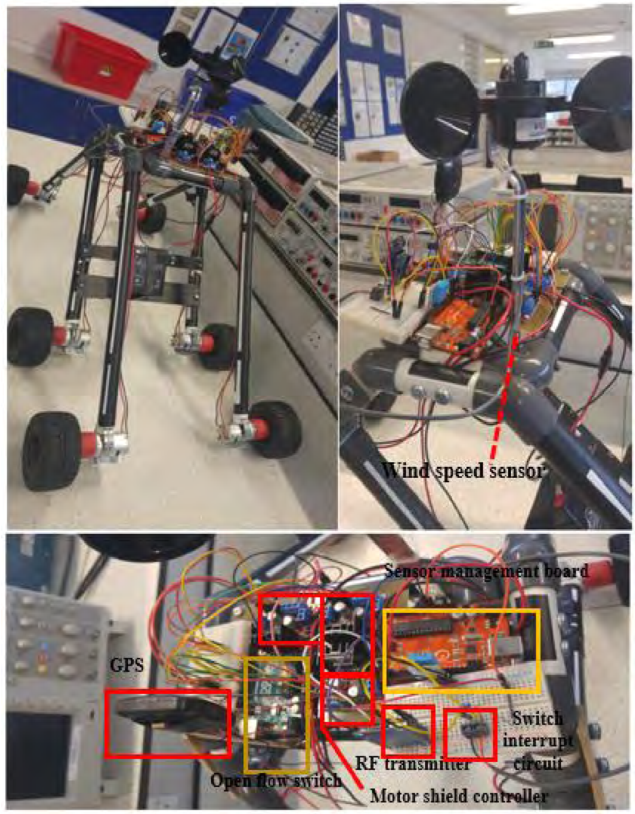

FIGURE 4. Proposed 6WI hardware design.

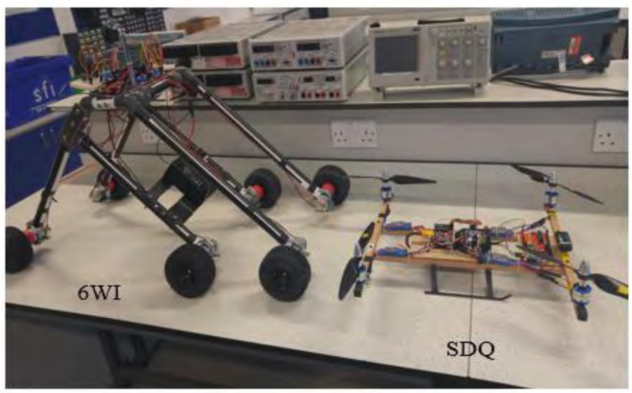

FIGURE 5. SDQ and 6WI testbed.

does power management on the quadcopter and the mobile wheeled node. Our system can handle more nodes if needed, but with the single wheeled node, it can cover what many of fixed nodes cannot manage due to smooth implementation and deployment of the proposed system.

V. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a sensor IoT platform that consist of a set of $\gamma = \{1, 2, \ldots, N\}$ of N number of sensor IoT devices. The IoT devices could be implemented for various purposes such as traffic control, farm management, smart parking, etc. For our testbed, a system of $\varphi = \{1\}$ of N quadcopters is

TABLE 1. Notations used in the problem.

η_{fc+esc}	power consumed by the KK2 and ESC
Δ_{SDN}	power consumption of SDN controller
ψ_{xbee}	power consumption of the RF XBee module
χ_{esp}	power consumption of the Wi-Fi gateway
Ψ	Capacity of SDN controller
$\mu(t)$	active controller in a slot of time
$X_{ij}(t)$	matrix of SDN and OF switches
ζ_{OF}	set of open flow switches
ϕ_{SDQ}	total power consumed by the quadcopter
$\theta_{load}(t)$	load supplied on the controller
$\omega_i(t)$	number of OF requests in a time slot
$\tau_j(t)$	average processing time of a i^{th} OF request
$\Omega(t)$	SDQ response time

deployed to read and manage sensor data collection from the underground IoT wheeled vehicle. The software-defined quadcopter can dynamically move close to the location of the wheeled IoT node and within the range of Line-of-Sight (LoS). We formulate the power consumed by the quadcopter platform based on a set of given parameters. We define the total power consumed by the quadcopter as ϕ_{SDQ}. More parameters can be sub-defined as follows in eq(1):

$$\phi_{SDQ}(\alpha_{fc+esc}, \beta_{SDN}, \delta_{xbee}, \omega_{esp}) = \eta_{fc+esc}\left(\alpha_{fc+esc}\right) + \Delta_{SDN}\left(\beta_{SDN}\right) + \psi_{xbee}\left(\delta_{xbee}\right) + \chi_{esp}(\omega_{esp}) \quad (1)$$

The equation is parameterized by the following parameters: the power consumed by the kk2 board and the electronic speed controllers α_{fc+esc}, the larger the ESC, the more current will draw and sometimes it requires to install a heatsink on them due to the high power overheating because of high current. The next major factor in power consumption is the Δ_{SDN} which may become very high if many IoT nodes will be used due to the packet processing and forwarding rules management for every single IoT node in the farm. Thus, with our approach, we implement a single system which can reduce the power consumption of the SDN controller considerably, as a result of that, hovering time efficiency will be affected as well. Additionally, RF module ψ_{xbee} consume power due to the update and wake packets sent via the

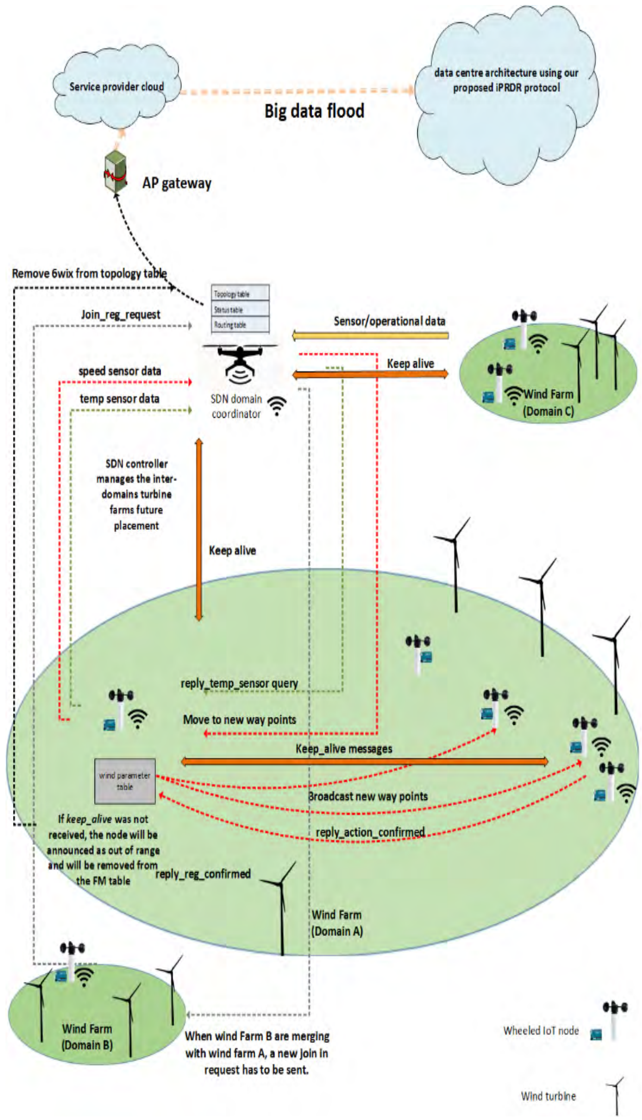

FIGURE 6. SDQ-6WI node management structure.

SDN controller. Any sensed data of wind has to be directed to the cloud for further processing. This is done via a Wi-Fi module ESPN8266 which can connect to the cloud in case there is gateway access point, and from there, the data are sent to our proposed data center (iPRDR).

Our proposed quadcopter is equipped with two network interfaces. One interface operates as a coordinator interface for the IoT, and the other one is non-overlapping and is used as a control interface. Additional hardware component can be added to the quadcopter to improve its performance and efficiency to handle larger networks which can be expressed as follows in eq(2):

$$P_{max} = \left(\sum \eta_{fc} + \sum\nolimits_{i=4}^{r} \eta_{esc}\right) + \sum\nolimits_{k=1}^{m} \Delta_{SDN} + \sum\nolimits_{l=1}^{v} \psi_{xbee} + \sum\nolimits_{p=1}^{h} \chi_{esp} \quad (2)$$

In a given network, it consists of a set of open flow switches denoted as $\zeta = \{z_1, z_2, \ldots, z_N\}$. On the other hand, in the

Algorithm: Developed OpenFlow (DOV)
Input: ζ_{OF}, Ψ_{SDN}
Output: Energy management for SDQ and 6WI

1. **Initiate** SDQActivate() in zone 1
2. **While** !InCoverage() **do**
3. WakeupRequest()
4. **For all elements in a** Ψ_{SDN} **set do**
5. $\zeta_{OF} \longleftarrow$ OFInfoRequest()
6. $\Psi_{SDN} \longleftarrow$ JoinRequest()
7. Ψ_{SDN} = TopoTable()
8. ζ_{OF} = BuildForwardingRule()
9. $\zeta_{OF} \longleftarrow$ DataRequest()
10. $\Psi_{SDN} \longleftarrow$ WindSpeedReading()
11. **If** WindSpeedReading() != recevied **then**
12. $\zeta_{OF} \longleftarrow$ ResendRequest()
13. **End if**
14. **End while**
15. **While** !NotInCoverage() **do**
16. 6wiSleep()
17. **End while**
18. **If** ReadingSufficient(*time*) **then**
19. $\zeta_{OF} \longleftarrow$ NewWaypoint()
20. HoldWaypoint(*time*)
21. $\Psi_{SDN} \longleftarrow$ WindSpeedReading()
22. **If** MergeDomain() == **available() then**
23. $\Psi_{SDN} \longleftarrow$ New6wiJoin(*zone x*)
24. $\zeta_{OF} \longleftarrow$ OFInfoRequest()
25. $\zeta_{OF} \longleftarrow$ DataRequest()
26. Ψ_{SDN} = TopoTable()
27. ζ_{OF} = BuildForwardingRule()
28. **End if**
29. **Go to** 2

FIGURE 7. DOV algorithm.

FIGURE 8. SDQ fly test.

```
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 5.0 mph
*||Temp23- Humd43||*
+--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 12.5 mph
*||Temp23- Humd43||*
+--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 13.7 mph
*||Temp23- Humd42||*
*--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 13.7 mph
*||Temp23- Humd42||*
+--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 13.7 mph
*||Temp23- Humd42||*
*--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 13.7 mph
*||Temp23- Humd42||*
*--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 13.7 mph
*||Temp23- Humd42||*
*--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 11.2 mph
*||Temp22- Humd43||*
+--------------------------------------------------
*************** Open FLow table data ***************
*Current Waypoint Wind speed: 6.2 mph
*||Temp23- Humd42||*
*--------------------------------------------------
```

FIGURE 9. 6WI forwarding table in active mode.

control plane, a set of controllers with processing capabilities denoted as $\Psi = \{\Psi_1, \Psi_2, \ldots, \Psi_N\}$ where $\Psi_j \succcurlyeq 1$, j $= 1, 2, \ldots$ K. The control plane may consist of multiple set of controllers, represented as μ. In a time slot of t,the control place consists of $\mu(t)$ of active controllers. Table 1, summarize the notations used in the problem formulation. We can verify that the correlation between the SDN and the open flow switches can be denoted as a binary matrix of N $\times$ $\mu(t)$ matrix X(t), where each OF switch is connected to one controller or multiple controllers depending on the needs of the network scalability. Eq(3) shows the relationship between SDN and OF in network system.

$$X_{ij}(t) = \begin{cases} 1, & i^{th} OF\ switch\ is\ connected\ to\ j^{th}\ controller; \\ 0, & otherwise \end{cases} \tag{3}$$

Open flow switches aggregate traffic to the controller when a mismatch in the forwarding table is found. A particular request to the controllers has to be made before the packet is sent. We can denote the load on the controller as follows in eq(4):

$$\theta_{load}(t) = \sum\nolimits_{l=1}^{h} \omega_i(t) X_{ij}(t) \tag{4}$$

Within a time slot t,the request from OF switches can be held in a queue due to the limited processing time of the SDN controller. However, if we apply little's law, we can derive the sojourn average time of request waiting in the queue to be as follows in eq(5):

$$\tau_j(t) = \frac{1}{\Psi_j - \theta_{load}(t)} \cdot |\zeta|^2 \tag{5}$$

The SDN controller response time can be calculated based on the total node density of OF switches and the number of incoming OF switch requests sending the request as follows in eq(6):

$$\Omega(t) = \frac{\zeta_{OF}.\tau_j(t)}{\sum_{j=1}^{\mu(t)} \theta_{load}(t)} \tag{6}$$

Minimizing the total request sent by the OF switch helps in optimizing the response time. However, the activation of the IoT sensors depend on the service that they provide. For example, home automation IoT and environment monitoring need to report measured data to the controller depending on the activation periods. IoT response time may be random, so we adapt our quadcopter to be deployed to collect data while dynamically managing the pattern of data acquisition from the IoT nodes.

VI. ALGORITHIM FORMULATION

Our proposed (DOV) algorithm is designed specifically to control the duration of sensing and the duration of staying in active mode. The 6wi node does not need to measure data all the time as many of the data would be not needed. Thus, the algorithm manages when to read wind speed measurements

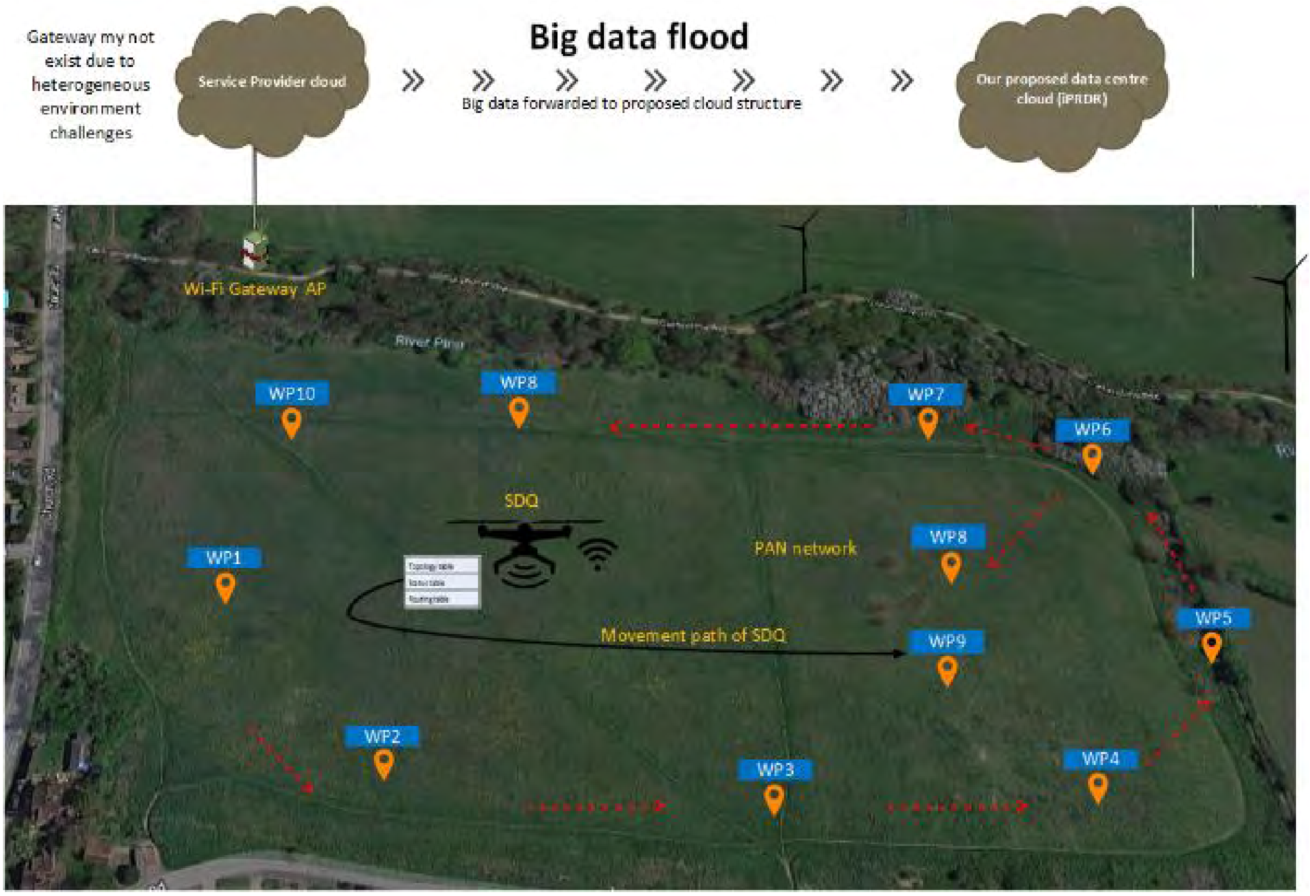

FIGURE 10. Experimental waypoint motion setup for 6WI sensor during SDQ fly test.

and for how long. This technique differs from previous algorithms as it reduces power consumption management as in sleep mode. The electronic boards will only draw a micro level of amperage. Moreover, our proposed infrastructure can be used to manage inter-domain wind farms that implement the same structure under one SDN control unit. However, mobile nodes have to send a join request to the controller and register their data in the topology table. The defined architecture will help merge wind measurements of farmland under one central management instead of depending on multiple controllers. In Fig. 6, we present our proposed structure with node communications, whereas, in large wind farm more than one mobile station is required to be deployed to cover large area. The main mobile station will act as a sink node to send new rules and location modifications to the open flow switches including keep alive messages. The sink node will pass on any sensor data to the SDN controller. However, if a new farm area needed to be merged, a new join in request can be sent to the SDN controller to be part of the current farm domain. This will allow flexible fault-tolerant system in case the next SDN controller failed. Furthermore, Fig. 7 represents the Developed Open flow (DOV) Algorithm. The algorithm manages when to send sensor data to the controller. It also governs the sleep and active periods of the ground mobile station. However, if the SDN controller not in coverage, the mobile station does not need to send sensor data to preserve power consumption levels to minimum. Additionally, the SDN controller can identify when to receive sensor data and at what time slots that correlate with wind meteorological data.

VII. FIELD EXPERMINEMTS AND RESULTS

To verify the effectiveness of our proposed design, we conducted field experiments in a designated test zone. The experiment involved collecting wind speed measurements to detect the best efficient wind strength location alongside humidity and temperature readings of the air that affect on the air density which may impact the wind turbine energy production.

Fig. 8 shows the SDQ flying test with data collection running. In our experiment, we have deployed the quadcopter with the 6WI, and the system operated as expected. The 6WI was providing accurate measurement of wind speed, temp, and humidity for any given waypoint via the quadcopter. The quadcopter flies to the mobile 6WI node and sends an info request to the node to build the SDN topology table, and then it sends multiple requests for data collection and location management. The collected data is then uploaded to the cloud or in case there is no internet gateway

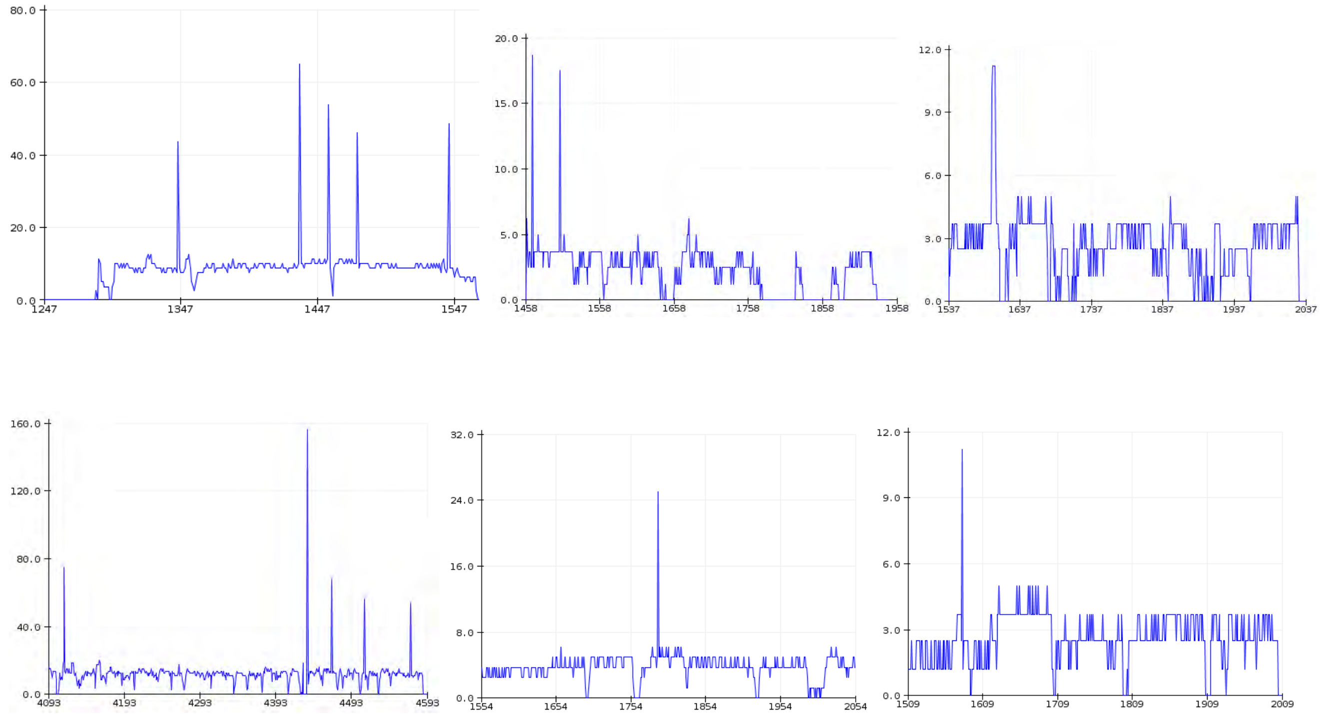

FIGURE 11. Comparison of measured wind speed in different way points.

point, the sensor data can be stored for further processing. In Fig. 9, we present a snapshot of the forwarding table of the 6WI switch while in active mode sensing. The forwarding table contains related sensor data information. The forwarding table contains weather parameters such as temperature, wind speed, air density, and open flow locations as well. All these data are feed to the SDN controller to have an overview of the entire network topology for efficient management.

The one-time slot navigation process takes about 780 seconds. We were able to collect multiple wind speed readings as we see in Fig. 10. In our test, we divide the farmland into random waypoints in a relatively close distance to measure the wind speed in each location. We can consider to have any array of location grids to cover e very possible grid in the farmland. The experimental parameters were shown in Table 2 with all the details included.

We conducted a comparison between the measured wind reading samples from different waypoints and found that the best efficient wind power is shown in Fig. 11 with wind readings above 15mph, while the other point showed low readings ranging from 3 to 8 mph.

We notice in Fig. 12 that the power consumption in the traditional system increase due to the number of sensors which is considered to be power-hungry because of the massive updates and requests that each node require. However, in our proposed model, we maintained a gradually fixed power consumption levels and covered many waypoints that the traditional nodes cannot cover. Concurrently, in Fig. 13, we compared the power consumption levels for different scenarios, including active and sleep periods of the nodes and found that our proposed system consume less power when using our proposed (DOV) algorithm which manages when the sensor reading process is required. In Fig. 14, we compare the humidity and temperature in each waypoint as it may have a significant effect on the service life and availability of the wind turbine [19]. The area with low humidity is a perfect candidate location for future wind turbine install. Furthermore, the effect of corrosion on wind turbines after a period of interruption, especially air may contain a corrosive combination of moisture and elements of all types.

TABLE 2. Expermintal parameters and values.

Parameters	Values
Radio protocol	802.15.4
Channel frequency	2.4 GHz
Transmission power	18dBm / 63mW
Transmission rate	250kbit/s
MTU	127 bytes
Baud rate	9600
Test area size	337m x 215m
Routing protocol	Developed Open Flow (DOF)
Experimental time per run	780s
Total runs	130
Total SDQ weight	1100g

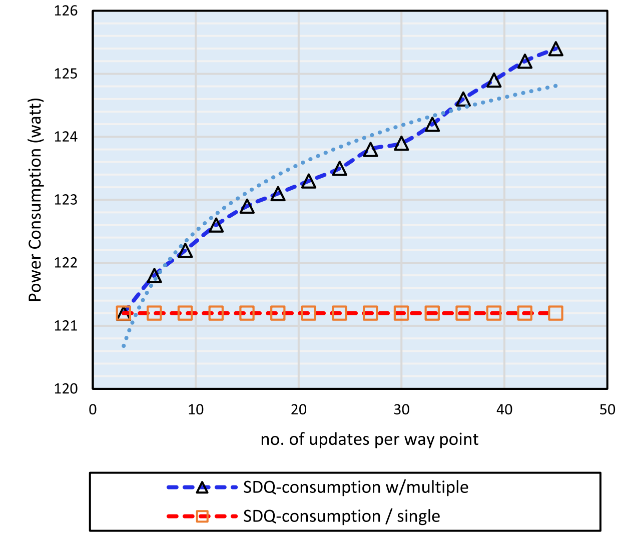

FIGURE 12. Node density vs. power consumption.

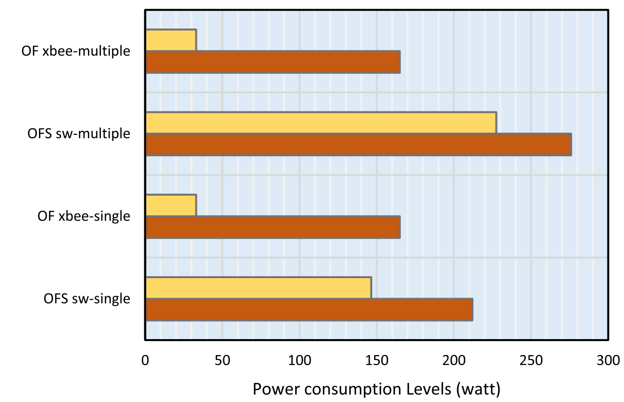

FIGURE 13. Power consumption scenarios.

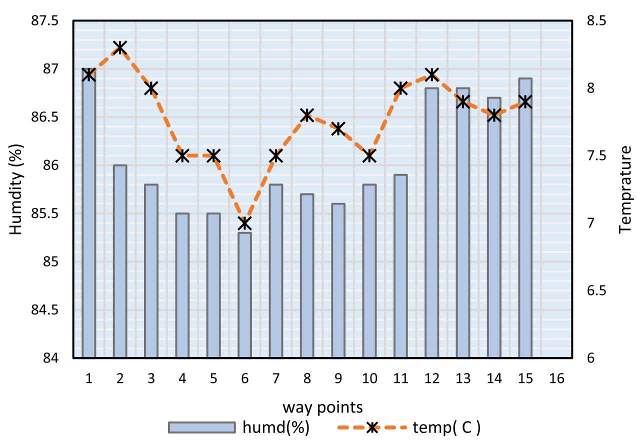

FIGURE 14. Humidity and temperature readings in a set of waypoints.

These elements in the air encourage and accelerate corrosion and may cause electrical fault such as a short circuit when turning up the wind turbine after a period of downtime. Thus, measurement of the two mentioned factors are very crucial alongside wind strength to allocate the most efficient optimal location for future wind turbine placement.

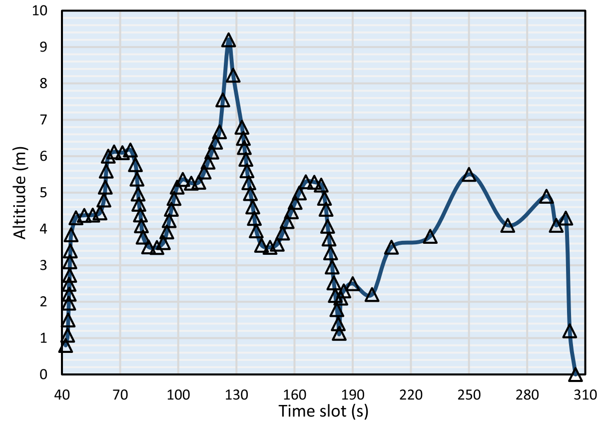

FIGURE 15. Q-Altitude variations per time slot(s).

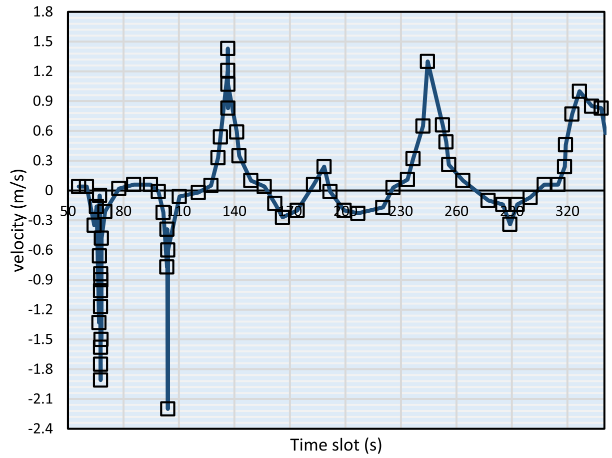

FIGURE 16. Q-velocity density levels.

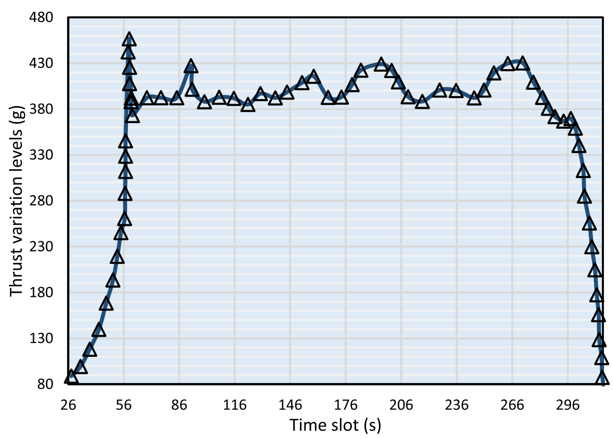

FIGURE 17. Q-thrust intensity per runtime.

Figs. 15, 16 and 17, illustrates the quadcopter test parameters that is the max altitude, thrust and velocity rates within a period of runtime. However, we notice the variation of velocity when changing the grid location to perform new measurement in a new grid, while thrust is maintained since the quadcopter is keeping same level of altitude with only changing speeds. Moreover, Fig. 18, presents the heat map

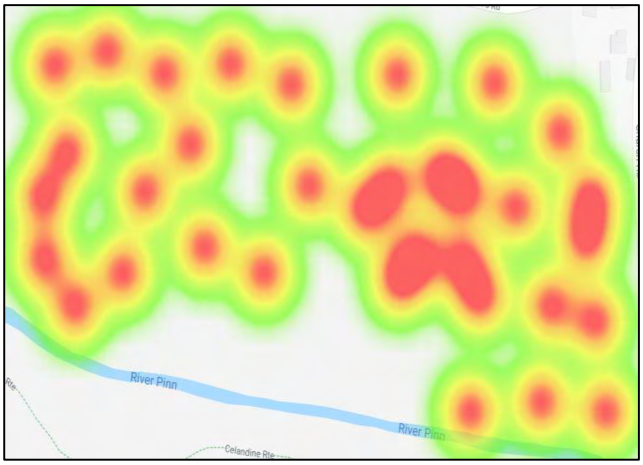

FIGURE 18. Waypoints heat map in test area per timeslot.

FIGURE 19. Humidity effect on wind turbines.

for measured waypoints that have been tested by our 6WI wind reader. The random points array were selected to cover as much as possible points that may have more efficient wind flow. Finally, Fig. 19, we notice that the corrosion [19] have affected the wind turbines after a period of downtime due to the increased humidity levels in the air. The corrosion is caused by a high humidity area what is not a good choice when planning to select an appropriate wind farm install.

VIII. CONCLUSION

The developed prototype SDQ-6WI has been designed to select the most efficient location for future wind turbine placement. With the analysis of various factors, we proved that our architecture could manage future wind turbine farms and find the best waypoint that comprises optimal wind speed and humidity levels. Additionally, we investigated the power consumption of the SDQ and presented the factors that may impact power consumption levels for our current system versus traditional architectures. We conducted extensive testing in open farmland. The experimental results demonstrated the effectiveness of our design by 23.19% of power consumption drop versus traditional fixed sensor nodes. In future work, we will enhance our SDQ network to work with multiple controllers from other domains to expand the network scalability with less power consumption techniques.

REFERENCES

[1] BP Global. (2018). *Wind Energy | Renewable Energy | Statistical Review of World Energy | Energy Economics | BP*. Accessed: May 15, 2018. [Online]. Available: https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy/renewable-energy/wind-energy.html

[2] F. Hu, *Network Innovation Through OpenFlow and SDN—Principles and Design*, 1st ed. Boca Raton, FL, USA: Auerbach, p. 455AD.

[3] B. Zhang, Z. Zhang, Z. Ren, J. Ma, and W. Wang, "Energy-efficient software-defined data collection by participatory sensing," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7315–7324, Oct. 2016.

[4] S. Azodolmolky, *Software Defined Networking With OpenFlow*, vol. 21. Birmingham, U.K.: Packt, 2003.

[5] Z. He, D. Zhang, and J. Liang, "Cost-efficient sensory data transmission in heterogeneous software-defined vehicular networks," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7342–7354, Oct. 2016.

[6] S. Dey *et al.*, "Ad-hoc networked UAVs as aerial mesh network for disaster management application and remote sensing: An approach," in *Proc. 8th IEEE Annu. IEMCON*, Oct. 2017, pp. 301–304.

[7] A. K. Al Mhdawi and H. S. Al-Raweshidy, "iPRDR: Intelligent power reduction decision routing protocol for big traffic flood in hybrid-SDN architecture," *IEEE Access*, vol. 6, pp. 10944–10955, 2018.

[8] S. Park, H. Kim, K. Kim, and H. Kim, "Drone formation algorithm on 3D space for a drone-based network infrastructure," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 1–6.

[9] J. Chen, S. Li, S. Chen, S. He, and Z. Shi, "Q-charge: A quadcopter-based wireless charging platform for large-scale sensing applications," *IEEE Netw.*, vol. 31, no. 6, pp. 56–61, Nov./Dec. 2017.

[10] K.-W. Chiang, G.-J. Tsai, Y.-H. Li, and N. El-Sheimy, "Development of LiDAR-based UAV system for environment reconstruction," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1790–1794, Oct. 2017.

[11] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[12] C. Rametta and G. Schembra, "Designing a softwarized network deployed on a fleet of drones for rural zone monitoring," *Futur. Internet MDPI*, vol. 9, no. 1, p. 8, 2017.

[13] M. Azizian, S. Cherkaoui, and A. S. Hafid, "Vehicle software updates distribution with SDN and cloud computing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 74–79, Aug. 2017.

[14] K. Ohsugi, J. Takemasa, Y. Koizumi, T. Hasegawa, and I. Psaras, "Power consumption model of NDN-based multicore software router based on detailed protocol analysis," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1631–1644, May 2016.

[15] W. R. Pdqdjh, W. K. H. Xqghuo, L. Q. J. Qhwzrun, Q. W. Sdshu, Q. Frqwhqw, and S. Wudi, "Joint content placement and traffic management for cloud mobile video distribution over software-D," *China Commun.*, vol. 13, no. 8, pp. 1–12, 2016.

[16] Arduino. *Ardunio Board*. Accessed: Mar. 21, 2018. [Online]. Available: https://www.arduino.cc/

[17] Inspeed.com. *Inspeed Wind Sensor Hand Held Wind Meter Anemometer Storm Chaser Weather Instruments*. Accessed: Mar. 14, 2018. [Online]. Available: www.inspeed.com

[18] Windpower.org. (2003). *The Energy in the Wind: Air Density and Rotor Area*. Accessed: Mar. 14, 2018. [Online]. Available: http://drÃÿmstÃÿrre.dk/wp-content/wind/miller/windpower web/en/tour/wres/enerwind.htm

[19] *Wind Turbine Need Protecting Aganist the Effects of Humdity*. Accessed: Mar. 19, 2018. [Online]. Available: cotes.com/wp-content/uploads/2015/11/Cotes-Wind-On-Shore.pdf

AMMAR K. AL MHDAWI is currently pursuing the Ph.D. degree in electronic and computer engineering with Brunel University London, U.K. He is also a Cisco Certified Professional Engineer with over 10 years' of experience in network engineering and telecommunications. His research interests include IIoT, software-defined networks, wireless sensor networks, delayed tolerant networks, renewable energy, and smart grid.

HAMED S. AL-RAWESHIDY (M'92–SM'97) received the B.Eng. and M.Sc. degrees from the University of Technology, Baghdad, Iraq, in 1977 and 1980, respectively, the Post Graduate Diploma degree from Glasgow University, Glasgow, U.K., in 1987, and the Ph.D. degree from the University of Strathclyde, Glasgow, in 1991. He was with the Space and Astronomy Research Centre, Baghdad, PerkinElmer, Waltham, MA, USA, British Telecom, London, U.K., Oxford University, Oxford, U.K., Manchester Metropolitan University, Manchester, U.K., and Kent University, Canterbury, U.K. He is currently the Director of the Wireless Network Communications Centre, Brunel University London.