SDQIM: Software-Defined Quadcopter Inter-domain Management Protocol with Decentralized Inductive Power Transfer (DIPT) platform

Ammar K. Al Mhdawi, *Member, IEEE,* Hamed Al-Raweshidy, *Senior Member, IEEE*

Abstract—**Due to the flexible deployment, low-cost and management, unmanned aerial vehicles (UAV), quadcopters have attracted significant interest lately in expanding wireless communications to areas that lack to cellular coverage network. Currently, SDN is considered a viable alternative that improves network scalability and management. Formerly, we designed a novel software-defined quadcopter (SDQ) with mobile IoT vehicle [1] to measure the best efficient wind power location for effective future wind turbine placement. However, large wind farms require complex and scalable wind measurement platform that consists of multiple quadcopters and ground mobile nodes that operate under a decision-making softwarized protocol. To this end, this paper proposes a novel cognitive routing protocol called Software-Defined Quadcopter Inter-domain Management (SDQIM) Protocol. The SDQIM structure consists of multiple SDN controllers with failover capability. Our proposed protocol operate with multiple phases as each phase is used to cover a specific scenario that could occur in the inter-domain network. The SDQ-main will have a total overview of the network topology and the SDQ-slave will be responsible for communication management with all other local sub-domains. However, for large-scale farmland, quadcopter power level is essential for the continuity of the system services. Thus, we additionally propose a Decentralized Inductive Power Transfer (DIPT) platform that operates autonomously using software defined controlling approach. In essence, The DIPT platform is considered to be automated and low-cost approach that encompasses a custom-designed SDN controller for all inter-domain quadcopters. Our system has been implemented by using an experimental testbed. The experimental results showed that scalability of the SDQ system can be increased to cover large wind farms with uninterrupted power management platform.**

Index Terms—**Software-Defined Quadcopter (SDQ), Software-Defined Network (SDN), Power management, Cognitive routing, Decentralized Inductive Power Trasfer (DIPT), IoT.**

I. Introduction

FORMERLY, traditional networks have been designed as all-in-one stack platform where control, data, and application plane are defined in the same hardware. This implementation caused the network to be more complex and prone to fails due to the extreme overhead that is supplied on each device. However, Software-defined Networks (SDN) have brought attention to many researchers and companies [2]. The SDN concept is implemented by separating the network control function represented by the control plane from the forwarding engine represented by the data plane. The process of segregation allows flexible network management, vendor-agnostic, cost-efficient and innovative network architecture. Additionally, SDN structure provides a set of application programmable interface (APIs) [3]. In SDN architecture, the intelligence is centralized in the control plane, while the downlink layer becomes the packet forwarding plane which can be programmed and managed via an interface called open flow (OF) interface. The segregation of the control plane allows flexible deployment of large architectures and applications with straight-forward network function virilization and easy implementation of routing policies and configuration instead of relying on distributed control devices. Therefore, by implementing such a concept, the network size can be reduced to multiple decision-making controllers with sets of forwarding engines. Moreover, the SDN devices contain rules to decide what to do with each incoming packet. The rules are injected from the southbound interface (SBI) to the forwarding switch. However, if an incoming packet has entered the forwarding device and there are no rules specified for this packet, then the forwarding switch will send a query-request to the controller to either supply new rules for the packet to the forwarding engine or drop it. The SDN controller acts as the final decision-making unit regarding policy rules management. Additionally, SDN controllers are configured via the northbound interface (NBI) where the full configuration is supplied as initial configuration or modification in network policy.

In recent years, wind turbines have become one of the most efficient and economical power generation technology that provides sustainable energy. However, wind location is considered to be a very critical factor when it comes to selecting the best location of wind turbines. A good windy site can generate energy that compares to the traditional energy production. The selection process of wind turbine site is considered the key element of any renewable energy implementation. Some factors come in play such as the wind class, wind strength that has to adhere to the optimal wind speed standards [4]. Although it may seem clear, the location of the wind turbine in the bottom of a hill or inside a valley is not a good option, while best location can be classified in high terrain hills or in open fields. The location selection of best wind speed is proportional to the amount of power generated by a wind turbine. That's mean increase in wind speed from 6m/s to 7m/s can result in 60% extra power from the turbine, as well as, increase in 36% of annual power production [5]. Companies, in general, install wind speed measurement units in a site before the wind turbine is placed. The wind measurement process takes between 6-12 months just to figure out the most efficient steady wind location. Although, This process poses economic and technical challenges and considered a time-wasting process. Quadcopters now days are getting a high portfolio due to their various uses in wide range of applications in the current industry. For this reason, we have introduced in our previous paper a quadcopter system with ground mobile nodes to measure the best efficient location of wind power before the wind turbine is placed. Current Unmanned Aerial Vehicles (UAV) systems can be used in heterogeneous environments that humans cannot reach such as disaster management and agricultural support. In our proposed approach, we expanded the implementation of SDQ from single to multiple nodes over a large farmland area for efficient scalability implementation. Furthermore, we proposed an inter-domain management protocol that is based on multiple SDN controllers with redundancy capabilities. The protocol is considered cognitive because it has the ability to manage and control data flow steering of the mobile ground stations with the ability of decision making in case of failure. Additionally, the power source lifetime of the SDQ can be crucial when it comes to long hours of wind speed measurement, thus, we proposed a quad- inductive charging station that is based on SDN platform with distributed charging techniques. In comparison with the current traditional wind speed measurement system, we found that our proposed architecture can provide on-the-fly centralized network management with no interruptions in hovering time. Additionally, our algorithm provides efficient and precise multi-domain management with power levels tracking and charging capabilities.The main contribution of our study can be summed up as follows:

- We propose a new architecture called Software-Defined Quadcopter Inter-domain Management (SDQIM) protocol. Our system comprises of a 4 x rotor IoT based quadcopter with SDN capabilities. For our farmland scenario, we divide the farmland into four regions, so we needed 6 x SDQs devices with SDN implemented in each quadcopter. The main quadcopter called SDQ_{main} which is responsible for managing the entire network topology. The second redundancy quadcopter is called SDQ_{slave}, which is in charge of local inter-domain management and provide tracking table and topology table copies to the SDQ_{main}. Any fails or out of range nodes (Domain controller quadcopters) can be recovered immediately with using our proposed routing protocol. The routing protocol is cognitive, thus having the ability to manage the failure scenarios and recover quickly with no interruptions. For our current architecture, we used one slave and one SDQ main controller. However, more slave controllers can be used depending on the requirements of the system and farmland size. We prove that our architecture works better than current wind speed measurement system by employing effective and reliable live sensing platforms for optimal wind power selection. The proposed system can work in environments that lack to Wi-Fi and cellular network via using low power personal area networks signal.
- We modeled the power consumption of the SDQIM architecture for the main and local controllers based on power optimization techniques that are specified in our proposed protocol. The modeling technique is an extension and development for the work presented in [6] to work with our system based on a given set of factors. The development of the power consumption formulas is based on managing the sleep and active times of the ground mobile station. When the mobile ground station is in movement to the designated coordinates, there is no need to measure wind speed, so the transmission RF node can go to sleep and start transmission ones the mobile station is in a proper position. This process will help in reducing power consumption for both the mobile ground station and the receiver quadcopter as depicted in the SDQIM algorithm in Fig 10, step (22-26). The modeled calculation was focused on the power-consuming parameters that may affect the inter-domain measurement continuity. Furthermore, we modeled the quad-coil inductive circuit power and current formulations. The used mathematical formulas are listed in the notation tables with their corresponding description.
- To deal with the problem of charging the SDQs as hovering is considered a power-hungry process. Moreover, the cost associated overhead with replacing and charging each SDQ after a specific run. Thus, we propose a decentralized dynamic software defined charging station. The controller is responsible for managing the charging process when the power is wirelessly activated and also to measure the power levels of the quadcopter and the main power source with a cloud connection. The wireless receiving coils are distributed over the four ends of the quadcopter to allow increased energy charge within a short period of time.
- We conducted extensive testing on our proposed architecture in a large farmland. The hardware that we used to build the SDQs and DIPT station is based on off-the-shelf IoT hardware. We observed that with using our proposed architecture, a wind measurement over large wind farm was managed effectively and with very low-cost budget compared to fixed sensor install charges and other overhead time-wasting process.
  The remainder of the manuscript is organized as follows: In Section II, previous work is briefly presented. In Section III, SDQIM protocol framework is defined and illustrated. In Section IV, DIPT platform design is presented. In Section V, testbed design and implementation are described. In Section VII, experimental results and analysis are illustrated. In summary, the conclusion and future work are prescribed in Section VIII.

II. PREVIOUS WORK

UAVs have played a big role in various industry sectors such as medical emergency, transport and disaster management as their benefits become more efficient when they are integrated with the Internet of Things (IoT). However, current IoT sensors cannot handle large distance communication as it is considered a limiting factor in data transfer. Nonetheless, researchers have been implementing new ideas to expand the scalability of the

UAV or quadcopter network. Although, limited work has been done on designing and implementing a necessary framework for several UAVs to work in the same environment. With all that being said, there are some technical challenges such as SDQ power consumption, wireless charging techniques and communications in heterogeneous areas that have to be addressed. In the following, we will present some of the work that was implemented by engineering researchers to solve some of the challenges in this field as follows:

Seongjoon *et al.* [6] proposed fantastic quadcopter formation algorithm with a mesh set of connected drones. Each quadcopter in the formation acts as an access point gateway. The algorithm objective is to reach the full capacity coverage in areas that lack to ad-hoc network coverage. The author provided that the test results have proven that the algorithm works better than terrestrial and organized formations. Another study was implemented as well on selecting the best efficient AP in regards to power consumption. However, power consumption management for entire quadcopter was not discussed as this type of protocol algorithm requires a high capacity communications link which is considered to be a power-hungry process. Jiming *et al.* [7] presented a quadcopter based wireless charging framework that can provide power through wireless communications to sensor-based applications. The project was implemented experimentally using an IoT testbed. The testbed consisted of power transfer module with pre-configured quadcopter with redesign implementation on power receiving antenna and energy conversion process. The wireless module is attached on the quadcopter to provide remote wireless charging. The testbed was tested experimentally in a dormitory building and on a farmland. With all that being said, the quadcopter, in this case, will need long hovering time to cover all the sensor application on a farmland. Thus, a more deep study is needed to prolong hovering time with a consistent energy source. Mohammad *et al.* [8] proposed a UAV system with a power-efficient mechanism using IoT system. The author investigated the process of minimizing the total transmission power from the IoT sensors. The author proposed a framework for optimizing the 3D allocation mobility of the UAV. The simulation results showed that the transmit energy was reduced by 45% with 28% system reliability comparing with the fixed nodes. Pingping *et al.* [9] proposed a WE-bridge platform to expand the inter-domain routing zones. The design was verified using a testbed platform with multiple SDN controllers. The author provided that some of the features that cannot be implemented in the traditional networks were implemented in the WE-bridge platform such as fine-granularity routing. However, the testbed can be expanded larger to accommodate large complex network where detailed network robustness and efficiency can be verified more accurately. Huma *et al.* [10] presented a novel intelligent routing protocol for vehicular network based on SDN technology.The algorithm that was used is the belief progression algorithm with channel selection. The algorithm consists of a couple stages that are the registration stage and the path prediction stage. The RSUs are designed as local controllers that are selected in the registration phase. The route prediction phase is implemented by supplying a reliable path to the requesting node by maximizing the route duration between source and destination. The author provided that the simulation results minimized the end-to-end delay with finer performance in packet delivery ratio. Han *et al.* [11] presented a dynamic wireless charging platform for substation inspection vehicular robots. The author investigated the centralized and decentralized energy pickup. A switching controller mechanism is proposed to implement the wireless charging. The author analyzed the configuration of the power receiving equivalent circuit. The author has concluded that the distributed energy pickup coils are better in performance than the centralized coil unit and higher efficiency are performed in cut-off charging points. Cheng *et al.* [12] presented a software-defined decision-making protocol for the internet of vehicles (IoV) with Q-learning capabilities. The author confirmed that the protocol can provide optimal routing via sensing and learning. The test was implemented on actual vehicular data set for validation and robustness verification. The author provided that the algorithm provide more efficient performance than traditional IoV protocol. The data set was implemented via OMNeT++ with SUMO traffic simulator. in conclusion, the proposed protocol dynamically achieve better results than GPSR and AODV protocols. Dharani emphet al. [13] proposed an emerging network architecture called SDN-enabled connectivity aware geo routing protocol (SCGRP) that is implemented via using mininet. The results were compared with CRP protocol for efficiency verification.The protocol is analyzed heterogeneous environment conditions for road sections before selecting the best path. Jian *et al.* [14] presented an energy-efficient protocol centroid-based routing protocol (EECRP) for IoT sensors. The protocol consists of sub-algorithms for clustering management. the first algorithm is for self-organization of local sensors and the second node is for rotating cluster head on the centroid to load balance energy load. The platform was simulated and results show that the protocol outperforms current protocols such as LEACH, LEACH-C, and GEEC. Sonam *et al.* [15] proposed an improved trickle routing protocol for low power networks. The proposed protocol build the destination-oriented acyclic graph by using information from the routing metrics, objective functions. The protocol timer works by sending control messages only if network instability detected, otherwise it uses a low sending rate with large window size. The author concluded that the scheme of the I-trickle protocol has been modified to provide better performance in terms of energy consumption and overall performance.

III. SDQIM PLATFORM ARCHITECTURE

Our developed SDQ prototype is designed to find the best efficient wind power location for future wind turbine placement. The proposed platform can be classified into two sections that are hardware implementation and software configuration. The hardware part is related to the design of the quadcopter while the software is related to function coding and virtualization.

A. SDQ_{main} hardware components

We started first by designing the SDQ frame from a light wood material with four arms (each arm is 40 cm in length). The arms hold 4 x DC 2830s brushless motors. The motors are mounted on propeller base with 20 cm blades. The propellers operate in both CW (clockwise) and CCW (counterclockwise) directions. A central board is attached to the arms intersection point to hold the entire components of the system. Each brushless motor is connected an Electronic Speed Controller (ESC) to control the speed of the blades for appropriate thrust lift. Furthermore, the voltage levels of the components vary between (3.3-12)v and a Power Distribution Board (PDB) is required for this purpose. An Arduino board [16] is implemented as the main controller along with XBee for RF transmission and receiving of data from SDQ_{slave} and Software Defined Charging Station (SD_{cs}). The rotors of the quadcopter can generate different motions and each motion is implemented with the x-axis and y-axis. The first motion is called the throttle which controls the up and down movement of the drone. positive throttle will make the quadcopter fly higher and a negative value means the quadcopter is flying lower. The second motion is called the yaw, which is the left and right motion within the x-axis line. A positive value means right turn and a negative value means left turn. The third movement is called the pitch, which describes the forward and backward of tilt of the quadcopter. The final motion is called the roll, which is the side to side tilt movement. We used the KK2 1.5 multi-rotor LCD flight controller board with 6050 MPU (Motion Processor Unit) gyroscope stability platform and AMega644PA board with (4.8-6.0) v and 6 channel operation. The sender controller operates at 6 channel frequency which is used to send motion directions to the flight controller board. The channels are in 2.4GHz band with 19db RF power. The frequencies range from 2.408-2.475GHz. However, these subbands are specified by the type of transmitter manufactured by the vendor. Additionally, The SDQ power source is based on 2200 mAh 1.5C Turnigy Lipo Btry that is attached to 4 x wireless inductive coils (each 14μH inductance). The received voltage supply is about 5v with a max current of 1.2 Amps. The PDB board provide the brushless motors with 0.5 Amp of current to maintain hovering power in the air.

B. SDQ_{slave} system components

IV. DPIT SYSTEM DESIGN PLATFORM

The need for an uninterrupted power charging capability in our design is considered very essential for the continuity of SDQ quadcopter services and to make sure that wind measurements are continuously being read. Thus, we designed a decentralized inductive SDN-based power charging station SD_{cs} that can manage the charging process for the quadcopters in our proposed platform. To the best of our knowledge, this is a novel idea where software-defined network platform is interfaced with a wireless charging station. The component of our prototype circuit consists of an Arduino board that is equipped with a customized code for the SDN function process. The controller is responsible for preserving the power of the ground power supply so that power does not get dissipated with continuous wireless charging and also it communicates with the quadcopters for the efficient charging process. The Arduino is connected to the main power source via a voltage divider circuit with 2 x resistors (1MΩ and 100kΩ), the output of the circuit is connected to the pulse width modulation pin of the Arduino for converting the periodic pulses accurate voltage readings. The other digital pin is connected to 5v relay module that can handle up to 10Amp of current with AC 220v / DC 30v 10A. The relay has a reverse protection diode. The signal pin of the relay is connected to the Arduino for sensor readings. The two ends of the relay are connected to the inductive wireless charging module and coil. The SD_{cs} mainly manages the voltage readings of the main power source to indicate if the new power source is needed, also it continuously checks the levels of the quadcopter power source via Xbee low power module. Additionally, the controller provides overcharging protection for the quadcopters so that it doesn't get overloaded with extra charge. The effectiveness of the SD_{cs} that handles 4 x local domains with 2 main SDQs has been experimentally tested with our proposed testbed.

V. TESTBED DESIGN AND IMPLEMENTATION

To test the effectiveness of our proposed design, we implemented our idea via IoT testbed by using off-the-shelf components. Fig 2 shows the SDQ_{main} hardware setup. The main SDN controller is connected to the uplink via using an ESP Wi-Fi module that is connected to the cloud via AP gateway. The cloud location is remotely stationed and managed by a vendor. We can access the cloud via a web-based http client. The application programmable interface (API) is implemented on the northbound interface (NBI) so that any rules or flow modification is sent via this interface.

As we see in Fig 2 that SDQ_{main} is installed on the back of the quadcopter which will communicate with the open flow (OF) switch mobile node. Nonetheless, if any waypoints or flow modification is required, it will be done via the NBI interface.

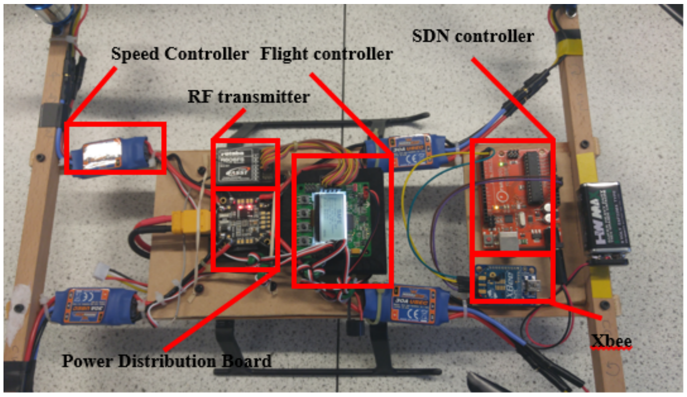


Fig. 1: SDQ_{main} proposed hardware

The power to the board is supplied via the PDB board with voltages range from (3.3v-12v). The PDB is used as well to feed 0.5Amp of current to each brushless motor for continuously managed propeller hovering. The KK2 flight controller board derive its power from the PDB with over-protection circuit installed. The SDQ_{main} have a different 9v power supply.

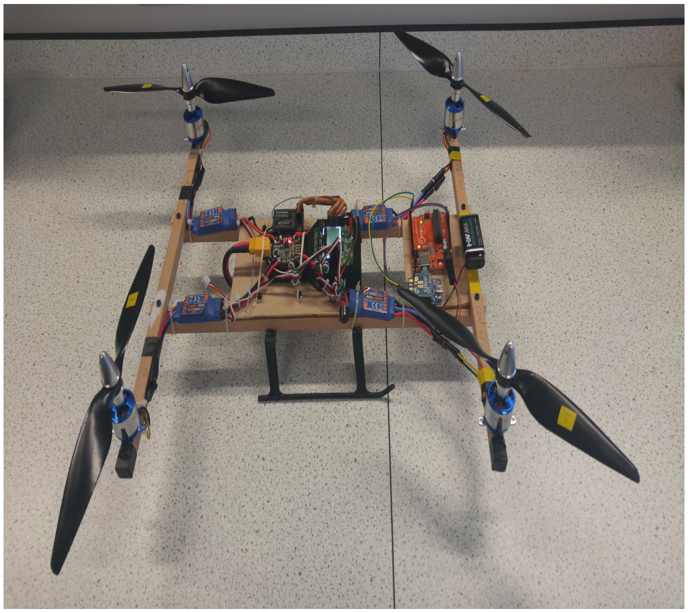


Fig. 2: SDQ_{main} overview

The main power source is installed under the center of the quadcopter with 4 inductive coil circuit install on each arm end of the frame as we see it in Fig 3. The design of the SDQ_{slave} is slightly different from the main controller that it can handle large data communication from the local domains and SD_{cs}. The main controller used is the pixhawck flight controller which is based on 168MHz M4 processor with 256K RAM and 2MB flash memory. It hosts a combination of significant sensors such as 3D ACC(Accelerometer), Mag(Magnetometer), Gyro(Gyroscope) and Baro(Barometer). The pixhawk can provide a homogeneous backup system for in-flight recovery and manual overriding with failover secondary power source.

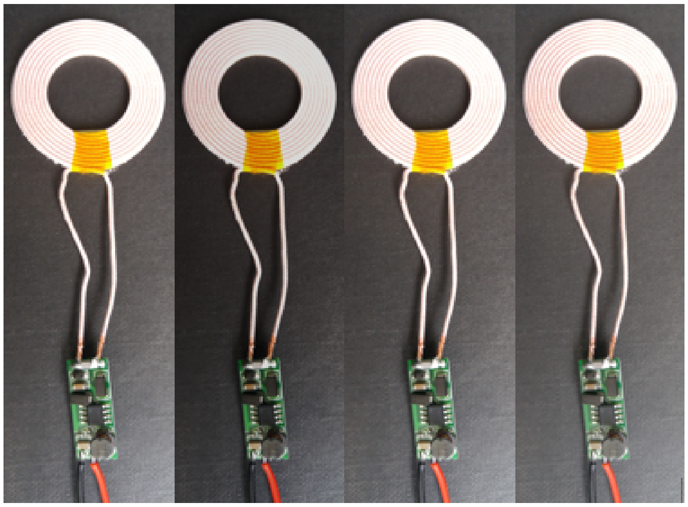


Fig. 3: inductive coils with wireless charging module

The SD_{cs} platform is based on an SDN microcontroller that can handle a vast number of sub-controllers while maintaining a high level of communication. The proposed platform is shown in Fig 4 and Fig 5 as we can notice that the controller is connected to the main power charging source via a voltage divider circuit for simultaneous power level checks and a relay circuit is connected between the main power source and the lipo wireless charger. This acts a circuit breaker in case there is no charging required. The proximity sensor to check if the SDQ is in approach to the transmitter coil, for that we used an ultrasonic sensor as that when the echo signal is interrupted, then the SD_{cs} will detect that charging is required which will turn the relay circuit on to initiate charging.

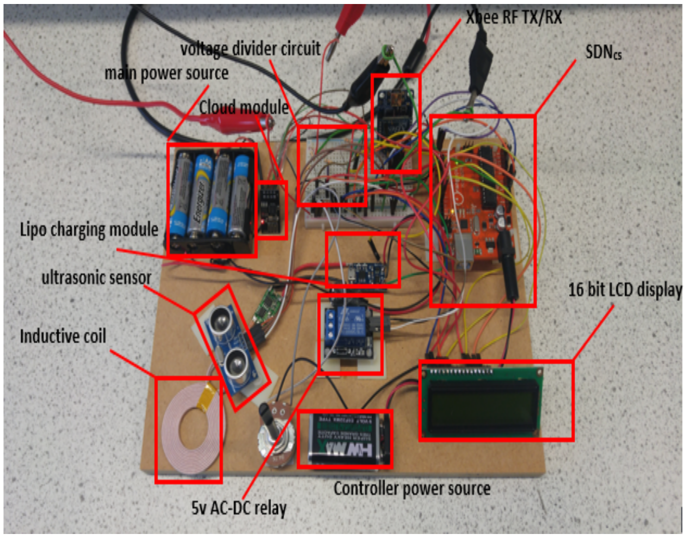


Fig. 4: SD_{cs} proposed testbed

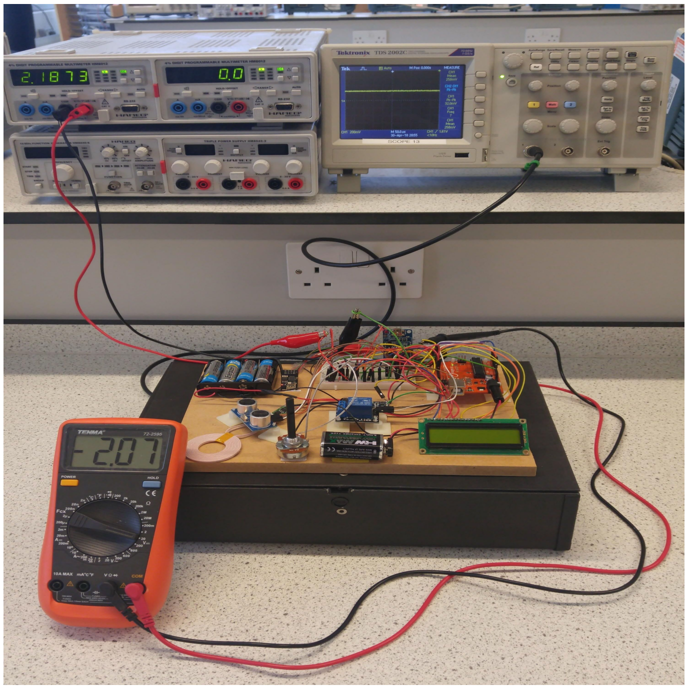


Fig. 5: testbed experimental testing

Additionally, the transmitter inductor coil is a 43mm diameter and 2.3mm in height. The inductance of the coil is 3.7μH.

The coil is connected to a lipo charging module which uses the TP4056 controller with DW01A protection circuit. Both the controller and protection circuit provide over-discharge protection that keeps the main power source from discharging below 2.4 volts. If the Btry for some reason got discharged below the 2.4v, then the module will disconnect the circuit until the charging reaches certain levels above the 2.4v. Additionally, it provides an short circuit protection if occurred. In Fig 7, shows an equivalent circuit model for the lipo charger module.

In Fig 6, we notice that a complete structure of the inter-domain wind speed measurement is designed with multiple software defined quadcopters (SDQs). The system consists of a main SDN controller which is in charge of overseeing the entire network topology with each node status tracking(battery life, node position, data counter). The main controller does not interpolate directly with the domain controller. However, we set a secondary SDN-slave controller that is responsible for gathering all sensor data from the domain controllers with send up the topology info to the main controller. The SDN-slave controller keeps a backup copy of the topology table and all related information of the nodes as in case of SDN-main fail, an easy failover process is implemented via the SDN-slave controller so that interruption does not occur in the system.

VI. Protocol Description and Power Modeling

In our system, we propose a decision-making routing protocol that is based on quadcopter-defined networking (QDN). The objective of our protocol is to expand the network scalability to handle large farmland and to allow distributed continuous power charging platform for each controller in the domains. We consider a set main controllers outside all domains λ = {1,2,.....,N} and a set of multiple quadcopter in various domains as β = {$Q1$, $Q2$, $Q3$, $Q4$,....Q_n}. We formulate the power consumed by each quadcopter based on a given set of parameters such as hardware and type of communications. However, we can define the total power consumed by the SDQ_{main} as Δ_{main} as follows:

$$\Delta_{main}(\alpha_{fc+rotor}, \delta_{sdn}, \psi_{rf}, \nu_{ap}) = \Delta_{fc+esc} + \Delta_{SDN} + \Delta_{rf} + \Delta_{ap} \tag{1}$$

These parameters may vary depending on the hardware capacity added size, as for our proposed model, each rotor draws about 0.5Amp of current and maybe more depending on the KvR capacity of the rotor which eventually will require a heat sink to be installed on each ESC due to elevated heat levels. Furthermore, the next factor in power consumption is the SDQ_{main} hardware, which could be very high with more IoT sensor deployment as more traffic will generate more heat and power utilization. Since our hardware is designed from off-the-shelf components, then future peripheral hardware can be added. The power consumption equation can be described as follows:

$$P_{total} = \Delta_{fc} + \sum_{i=1}^{r} \Delta_{rotor} + \sum_{k=1}^{t} \Delta_{sdn} + \sum_{l=1}^{u} \Delta_{rf} + \sum_{j=1}^{m} \Delta_{preph} \tag{2}$$

The main controller is responsible for keeping the global topology table updated and tracked. The main does not interact with the domain controllers directly. Otherwise, it uses the secondary controller for all update flows. Since the network can be scalable to hundreds of nodes, it is inappropriate that the network is controlled via one controller. Thus, our proposed system uses a redundancy approach to provide dynamic traffic load management. When a request packet reaches the secondary or SDQ_{slave} controller, the secondary checks its flow table, if a match is detected, then it immediately forward the packet and update its secondary topology table. If the packet route was not found in the secondary table, then the request packet will be sent to the SDQ_{main} for further instructions on what to do with the packet. The capacity is considered to be a big factor in minimizing the delay which can be expressed as follows:

$$\psi_{delay} = \frac{\sum \Phi_{pkt}}{\sum_{z=1}^{v} \Omega_{cap}} \tag{3}$$

The trajectory 2D distance between the main controller and the slave controller can be expressed as follows:

$$\sqrt{(x_{main} - x_{slave})^2 + (y_{main} - y_{slave})^2} \tag{4}$$

However, if a DoC (Domain Controller) node has failed or become out of range, the SDQ_{main} sends a broadcast message to all nodes with a request on finding the nearest reliable node in the domain. The other domain nodes start calculating the best link relay node (LRN) and then the LRN value gets shared with all the nodes to determine the best LRN value. The selected node will be the relay node to the SDQ_{main}. The LRN [10] can be expressed as follows:

$$\frac{r \pm d_{n1,nx}}{\sqrt{(v_{n1}\cos\theta_{n1} - v_{nx}\cos\theta_{nx})^2 + (v_{n1}\sin\theta_{n1} - v_{nx}\sin\theta_{nx})^2}} \times S(ch1, ch2, ...ch_n) \tag{5}$$

where v is the velocity, θ is the angle, d is the distance between first node $n1$ and any other node nx, r is the transmission range, and S is the channel that has the best reliability in a set of idle channels. Now, after calculating the best LRN value, the comparison is made to select the min K_{relay} value as shown below:

$$K = min(LRN_1, LRN_2, LRN_3, ..., LRN_N) \tag{6}$$

The multiple controllers that represent the control plane have different processing capacity which can be denoted as $\Psi_{cap} = \{\Psi_1, \Psi_2, \Psi_3, \Psi_4, ..., \Psi_n\}$. The load is a combination of packets that are being forwarded to the controller within time slot which can be noted as ϕ_{load}. We

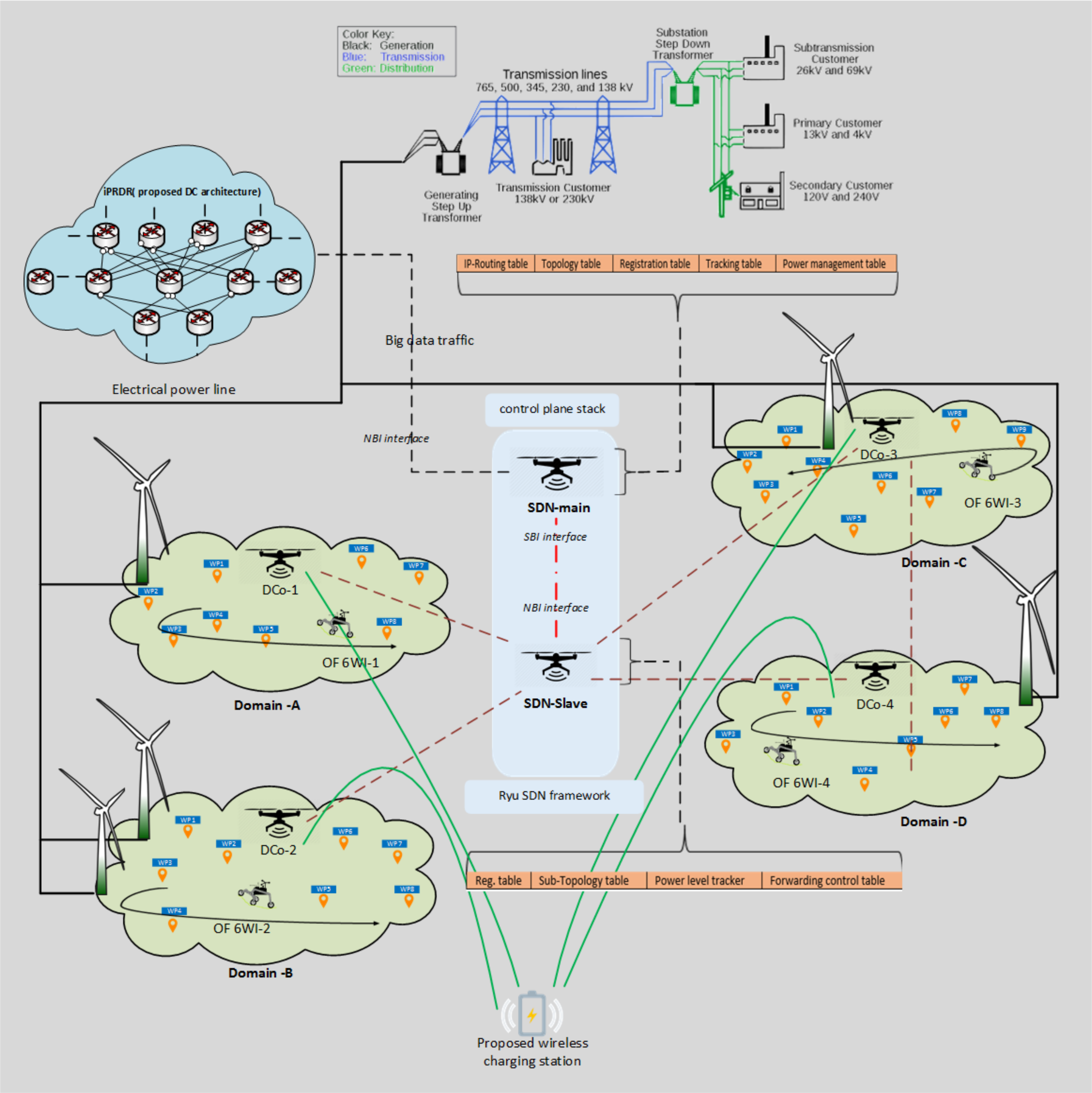


Fig. 6: SDQIM proposed architecture

can express the load as follows:

$$\phi_{load} = \sum_{q=1}^{c} \omega_{1,2,...n} \times \alpha_{qrate} \tag{7}$$

where $\omega_{1,2,...n}$ is the node density. We can calculate the main controller response time based on the above formula:

$$\eta(t) = \frac{\xi_{node} \times \Psi_n \times \sigma_t}{\sum_{j=1}^{\xi} \phi_{load}(t)} \tag{8}$$

where $\sigma(t)$ is the queuing time request and α_{qrate} is the queuing time slot per process. The equivalent circuit model [17] of the lipo charging is shown in Fig 7.

For our proposed circuit in Fig 8, the V_{in} is the AC power source generated from the charging station. w is the phase frequency and can be expressed by $w = 2\pi f$. R_t is the inner resistor of the main primary coil and L_t is the main primary transmitting coil respectively. The impedance of the L_t can be written as $Z_{L_t} = R_t + jwL_t$. The L_f and C_f are the main components that comprise the LCC circuit with their equivalent impedance as Z_{L_f} = jwL_f, Z_{C_f} = 1 / jwC_f and Z_{C_t} = 1 / jwC_t. The receiver circuit consist of 4 x coils

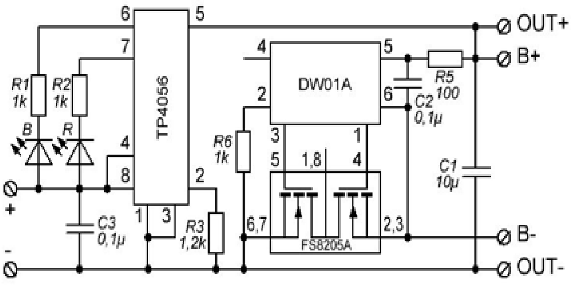


Fig. 7: Lipo charging circuit schematics

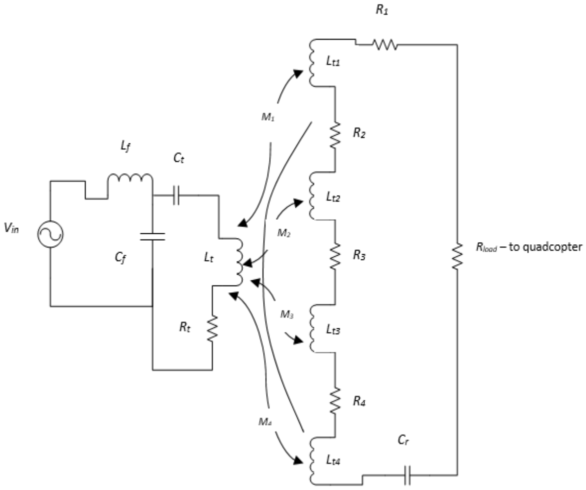


Fig. 8: proposed distributed quad-coil schematics

with their respective inner resistors. The mutual inductance of the L_t with the secondary coil are L_1, L_2, L_3 and L_4 respectively. We can specify the total mutual inductance between the main coil and the quad coils as L_{total}= L_1 + L_2 + L_3 + L_4. When the secondary 4 coils are in resonant state, we can express C_r as follows:

$$C_r = \frac{1}{w^2.L_r} \tag{9}$$

$$C_r = \frac{1}{w^2.(L_{r1} + L_{r2} + L_{r3} + L_{r4} + L_{total})} \tag{10}$$

Assuming that inductance on secondary receiving circuit have all same values.

$$C_r = \frac{1}{w^2.(4L + L_{total})} \tag{11}$$

The total impedance for the primary coil circuit can be expressed as:

$$Z_{total} = Z_{L_f} + (Z_{C_t} + Z_{L_t} + R_t)||Z_{C_t} \tag{12}$$

$$Z_{total} = jwL_{L_f} + \frac{(\frac{1}{jwC_r} + jwL_t + R_t).\frac{1}{jwC_f}}{\frac{1}{jwC_f} + \frac{1}{jwC_t} + jwL_t + R_t} \tag{13}$$

$$I_{in} = V_{in}.(\frac{1}{jwC_f} + \frac{1}{jwC_t} + jwL_t + R_t) + (\frac{1}{jwC_r} + jwL_t + R_t).\frac{1}{jwC_f} \tag{14}$$

The current of the secondary circuit can be calculated as follows:

$$I_q = \frac{jwL_{total}.I_{in}}{R_{load} + R_{inner}} \tag{15}$$

the power at the quadcopter load can be derived as:

$$P_q = I_q^2 \times R_{load} = [\frac{jwL_{total}.I_{in}}{R_{load} + R_{inner}}]^2.R_{load} \tag{16}$$

For our current protocol to work efficiently, a set of messages has to be communicated in order for the domain to be managed. Fig 9 shows the sequencing device to device communications.

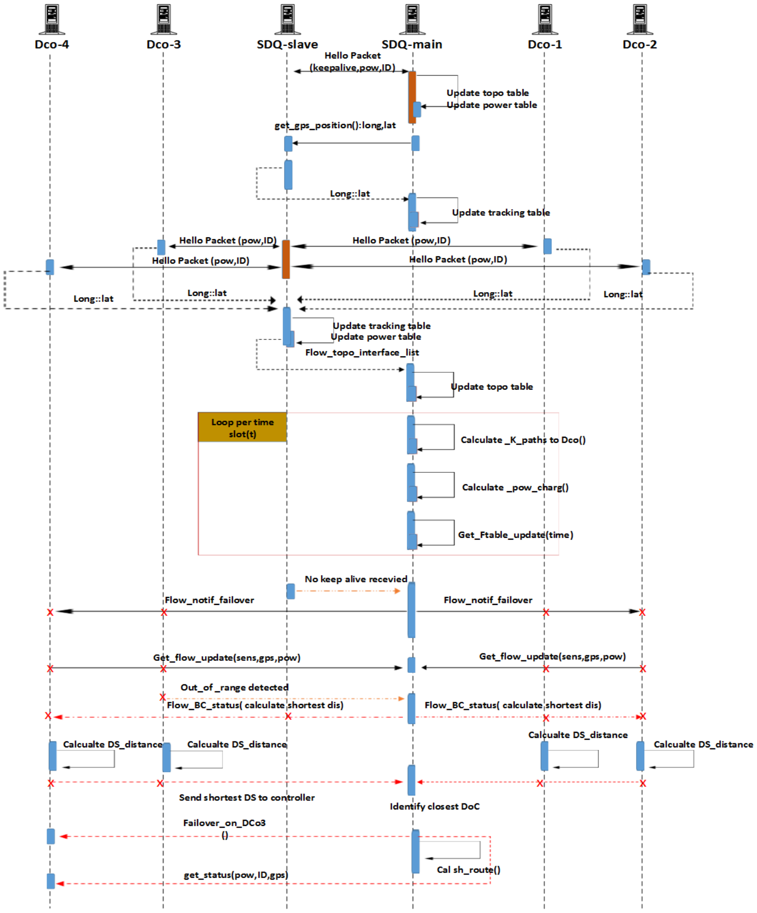
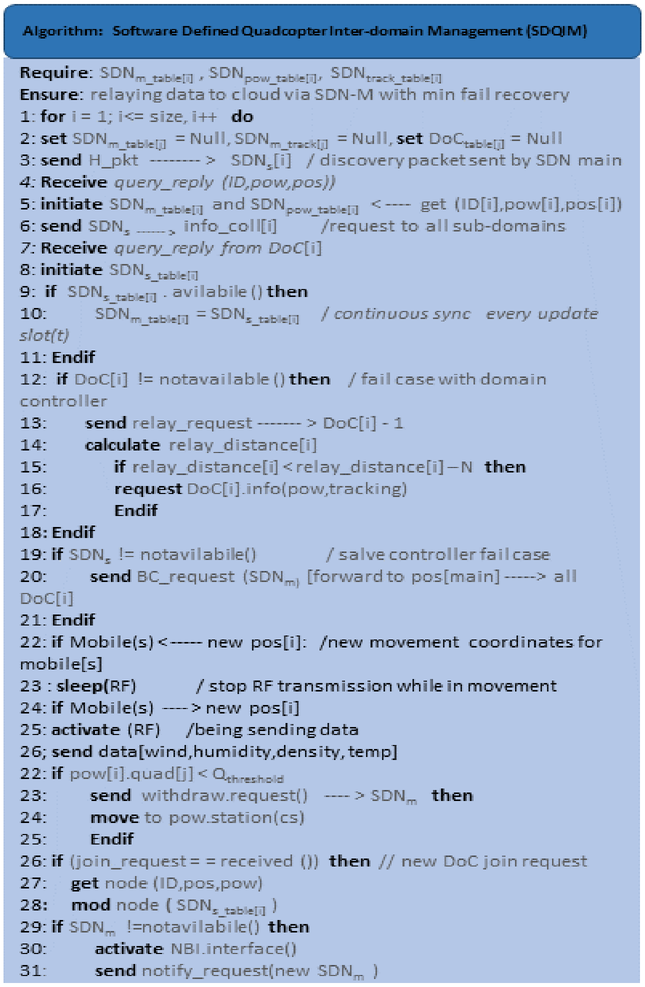


Fig. 9: SDQIM protocol sequence diagram

Fig 10 below shows the SDQIM algorithm process starting from initiation and covering all failure scenarios.



```
Algorithm: Software Defined Quadcopter Inter-domain Management (SDQIM)
Require: $SDN_{m\_table[i]}$ , $SDN_{pow\_table[i]}$, $SDN_{track\_table[i]}$
Ensure: relaying data to cloud via SDN-M with min fail recovery
for i = 1; i<= size, i++ do
set $SDN_{m\_table[i]}$ = Null, $SDN_{m\_track[i]}$ = Null, set $DoC_{table[i]}$ = Null
send H_pkt --------> $SDN_s$[i]  / discovery packet sent by SDN main
Receive query_reply (ID,pow,pos))
initiate $SDN_{m\_table[i]}$ and $SDN_{pow\_table[i]}$ <---- get (ID[i],pow[i],pos[i])
send $SDN_s$ ------> info_coll[i]    /request to all sub-domains
Receive query_reply from DoC[i]
initiate $SDN_{s\_table[i]}$
if $SDN_{s\_table[i]}$ . avilabile () then
        $SDN_{m\_table[i]}$ = $SDN_{s\_table[i]}$   / continuous sync every update slot(t)
Endif
if DoC[i] != notavailable () then   / fail case with domain controller
      send relay_request -------> DoC[i] - 1
      calculate relay_distance[i]
          if relay_distance[i] < relay_distance[i] − N then
          request DoC[i].info(pow,tracking)
          Endif
Endif
if $SDN_s$ != notavilabile()        / salve controller fail case
      send BC_request ($SDN_m$) [forward to pos[main] -----> all DoC[i]
Endif
if Mobile(s) <----- new pos[i]:  /new movement coordinates for mobile[s]
sleep(RF)        / stop RF transmission while in movement
if Mobile(s) ----> new pos[i]
activate (RF)    /being sending data
send data[wind,humidity,density, temp]
if pow[i].quad[j] < $Q_{threshold}$
      send withdraw.request()  ----> $SDN_m$  then
      move to pow.station(cs)
      Endif
if (join_request = = received ()) then // new DoC join request
    get node (ID,pos,pow)
    mod node ( $SDN_{s\_table[i]}$ )
if $SDN_m$ !=notavilabile() then
       activate NBI.interface()
       send notify_request(new $SDN_m$ )
```

Fig. 10: SDQIM algorithm

Parameter	Description
Δ_{SDN}	power consumed in the controller
Δ_{rf}	power consumed in the xbee module
Δ_{ap}	power consumed in the esp8266
Ψ_{cap}	processing capacity of SDN
LRN	link relay node
$d_{n1,nx}$	distance between mobile nodes
v_{nx}	speed of a mobile node
Ψ_{cap}	SDQ processing capacity
ϕ_{load}	incoming traffic load
$\eta(t)$	controller response time
$\sigma(t)$	queuing time request
Δ_{fc}	power consumed in flight controller
Δ_{preph}	power consumed by peripheral hardware

TABLE I: NOTATION TABLE

In Table 1, we have listed the main important notation that we have used in the problem formulation. Furthermore, we implemented extensive testing in an open farmland area with testing parameters that are listed in Table II.

Parameter	Values
test area map	1.5km x 2.0 km
traffic links	6
packet size	256 bytes
no. of channels	8
channel frequency	2.4GHz
transmission range	400m (LoS)
transport protocl	UDP with no authentication
no. of runs	350
Baud rate	9600
Transmission rate	250kbit/s

TABLE II: EXPERIMENTAL TESTING PARAMETERS

VII. SYSTEM EXPERIMENTAL RESULTS

To test the effectiveness of our proposed hardware and algorithm, we conducted extensive testing in the designated test area. The SDQ_{main} and slave were deployed to manage four inter-domains. Fig 11 shows the node distribution points in the designated test zone.

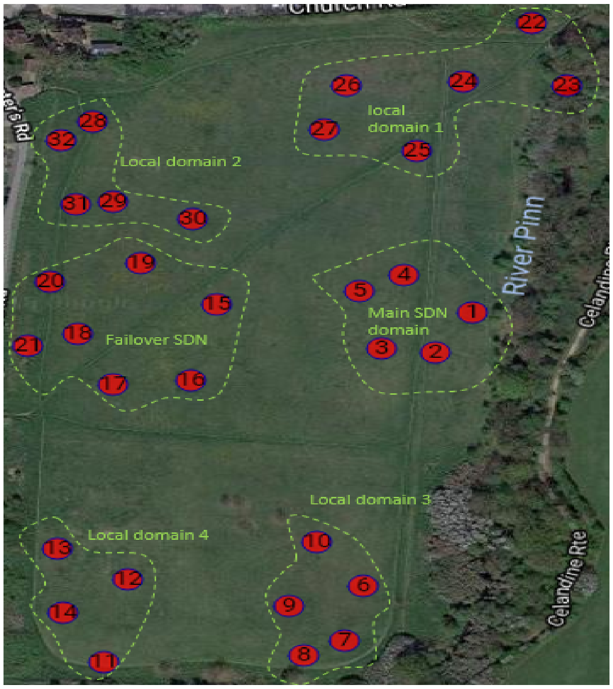
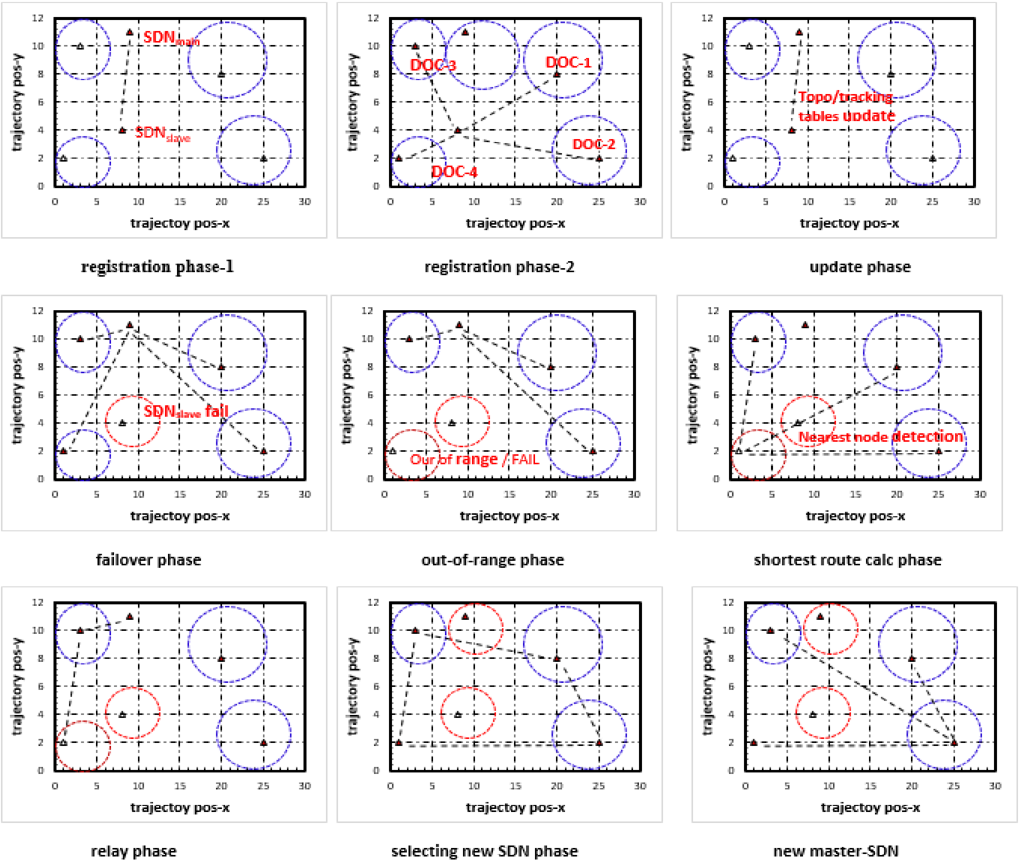


Fig. 11: node range distribution points

We can notice in the Fig 11 above that each quadcopter is located within its domain and the red dots represent the new points that the SDQ passed through. The location of the dotted points may change depending on varying testing conditions.



registration phase-1

registration phase-2

update phase

failover phase

out-of-range phase

shortest route calc phase

relay phase

selecting new SDN phase

new master-SDN

Fig. 12: communication phases

The communication phases covers, the registration phase which is divided into two sections, the SDQ_{slave} connection setup phase which is very crucial for establishing the redundancy process. The second registration phase is to communicate with all sub-domains. The next following stages in Fig 12 covers the failover case, where SDN_{main} or slave may fail which requires new controller takeover from one of the controllers. Additionally, out-of-range nodes can is detected via the controllers where the shortest link relay distance is calculated to establish a new route for the remote or out of range node.

In Fig 13, we notice the trajectory path of each node per time slot. The movement is kind of stable in each area. However, as wind speed increase in the area, the quadcopter has to establish some yawing and pitching balancing movements in order to maintain a controllable steady-level quadcopter. The hovering path may change due to network scalability reasons that is required to cover as much as possible of the farmland for efficient wind speed measurements. The inductive charging platform is placed near the farmland for the easy-access charging process.

In Fig 14, we measured the incoming traffic on the SDQ_{main} and we can notice that variant traffic spikes per rounds that reflect changes and updates occurring in the network which requires status and topology update. In Fig 15, the used ZigBee channel consists of 26 x channels. We used each channel and the measured packet delivery ratio was varying based on each channel capacity.

Although, in Fig 16, we notice a gradual drop in the SDQ power consumption. Ones the power reaches 25%, the SDQ will select a new path to the charging station(CS). When the route to charging is initiated, the SDQ will have over its routing responsibility to the slave node which provides efficient uninterrupted service management.

Since the charging is performed via our proposed inductive charging platform, the SD_{cs} will be communicating with all SDQs of sub-domains for the efficient charging process. In Fig 17, the traffic increases on the controller as the distance increase due to the new position and reachability conditions.

In Fig 18, the end-to-end delay may vary depending on the condition of the network. We notice that Dco have a similar steady delay, while the slave is the final delivery point within the sub-domain that collect all the sensor data and send it up to the main controller. The fail delay bundle covers the cases of either Dco fail or SDN main/slave fail that will require network reconfiguration internally which will impose additional delay parameters.

The following figures show the operational status conditions of the SDQ system. Fig 19 shows The SDQ_{main} position within different altitude levels as it is import to have a centralized overview of the network is that to have a top-altitude on all below sub-domains. Furthermore, sample velocity sets of

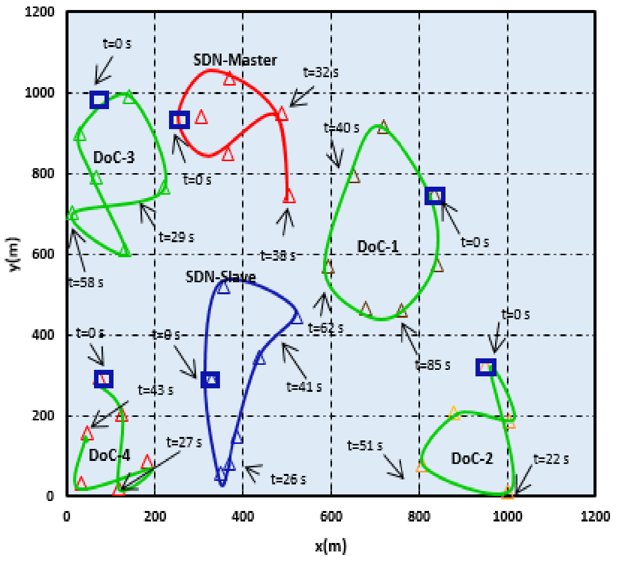


Fig. 13: trajectory path per domain

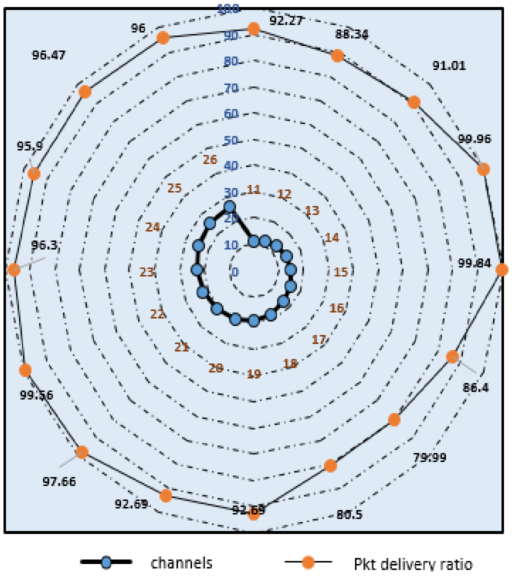


Fig. 15: packet delivery ratio vs xbee channels

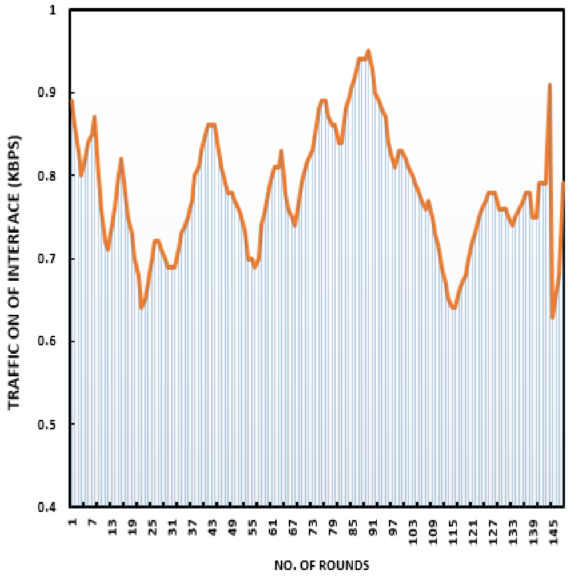


Fig. 14: incoming traffic on interface

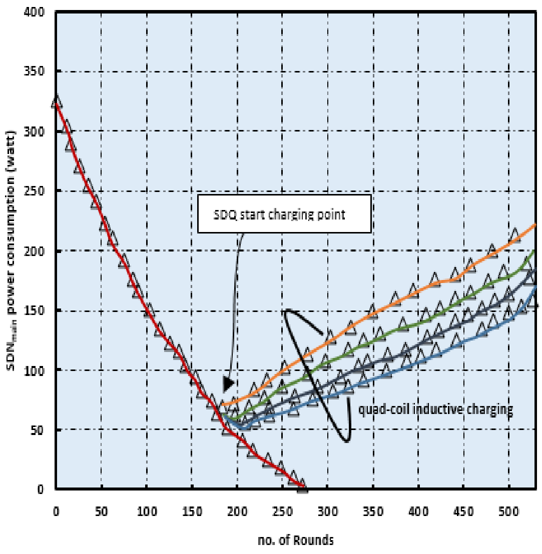


Fig. 16: SDQ charging via wireless charging platform

our proposed quadcopter are shown in Fig 20 which represents the four directions of movement. The negative velocity values indicate that quadcopter is moving down or left. Moreover, Fig 21, illustrates the quadcopter thrust that is required to keep it hovering within the domain. The start and end of the concave-up and concave-down curves are the start and end of quadcopter hovering points, while, the medium steady thrust represents the preferred stable altitude.

In Fig 22, The incoming traffic to the SDQ_{slave} is higher due to continued sensor data flow from all the sub-domains to the controller. However, the traffic to the main controller is limited only from the slave controller. Nonetheless, we notice spike change in the main controller traffic and that is due to the quadcopter request for charging as its power consumption have reached near the 25% value.

When a wind increase in an area, the quadcopter will try to manage the stability and balance using the IMU sensor. Thus, with wind increase, the rotor thrust power increases which will lead to more power consumption. In Fig 23, we measured the power drain levels and we found it increasing with wind speed respectively.

With regards to our proposed charging platform in Fig 24, we found that the proposed testbed charging ability is better than using a single coil system. The four receiving coils can charge the quadcopter with approximately double charging rate within the same period of time than using a single coil system which is prone to fail or not charging as intended.

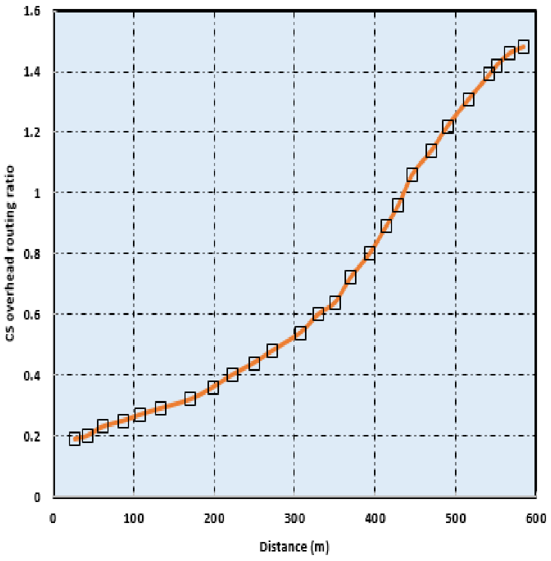


Fig. 17: CS routing traffic vs distance

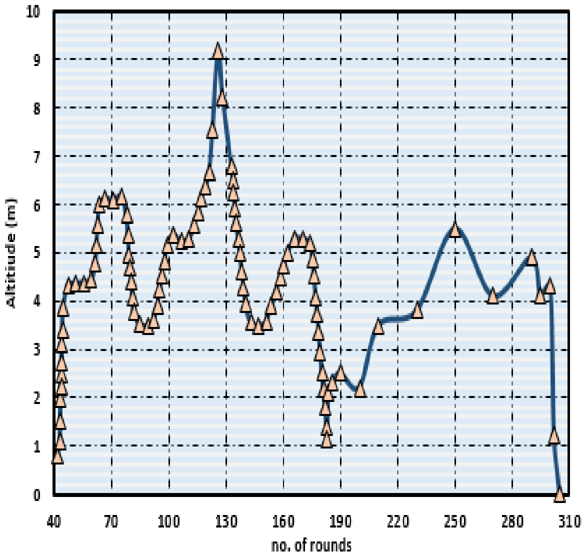


Fig. 19: SDQ$_{main}altitude$

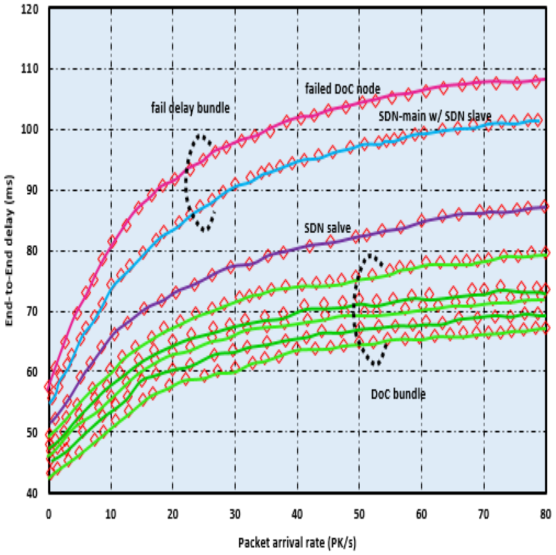


Fig. 18: end-to-end delay

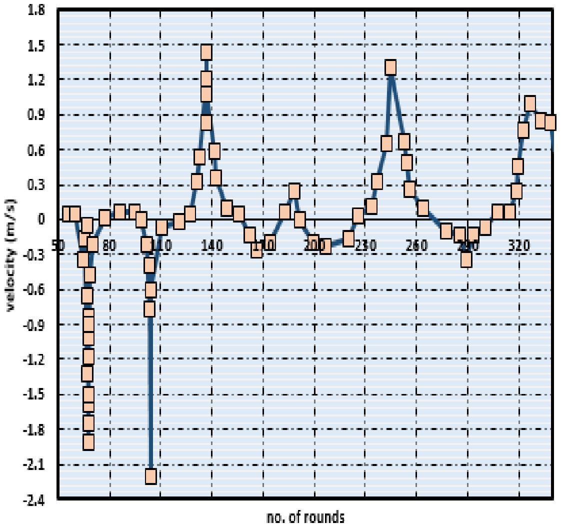


Fig. 20: SDQ$_{main}velocity$

VIII. CONCLUSION

The work in this paper examines the usefulness of developing and implementing our proposed inter-domain cognitive management protocol (SDQIM). The work is based on software-defined networking that is built on our designed prototype quadcopter system. Our SDQ prototype was designed to select the best optimal wind power location for future wind turbine placement with the support of mobile ground stations. However, with implementing such system many scalability and management challenges may occur. Thus, we designed a large-scale inter-domain management protocol that is based on multi-layer SDN controller. The main controller prototype was designed to handle updates and topology management from the slave node. The slave controller was designed differently in terms of hardware and capacity so that it can handle largely distributed sub-domains. Furthermore, in order to provide a continuous uninterrupted service, we designed a prototype charging testbed so that each quadcopter can charge itself autonomously and wirelessly via our proposed testbed using inductive quad-coil charging technique as this will double the charging process for more effective energy feeding platform. Our extensive experimental testing has proved that our system can provide a high reliability with continuous wind sensors readings for very large scale farmland.

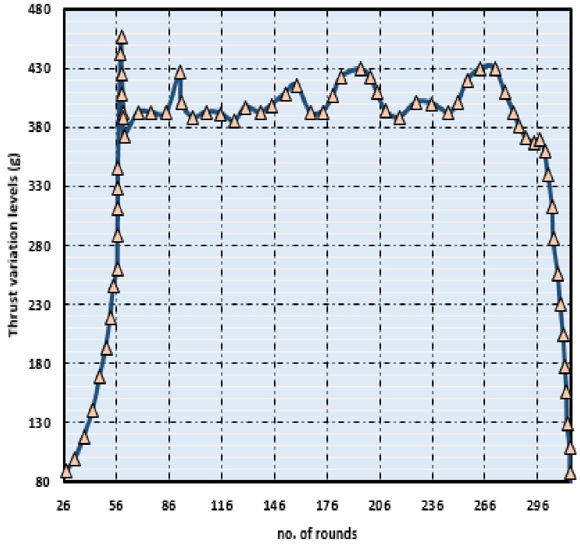


Fig. 21: SDQ$_{main}thrust$

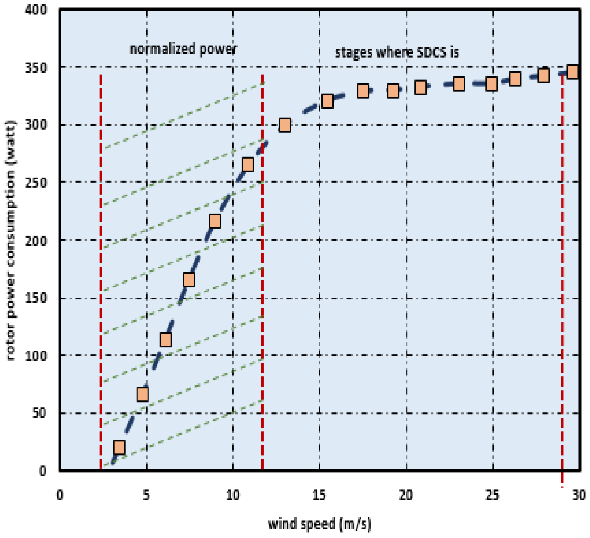


Fig. 23: SDQ power consumption vs. wind speed in the area

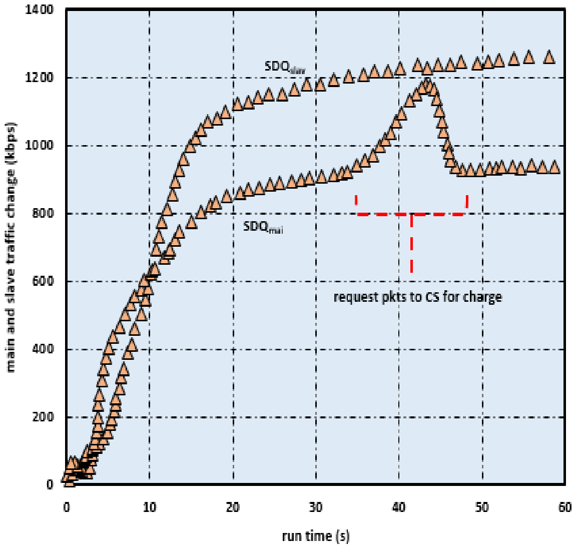


Fig. 22: traffic change due to CS charging request

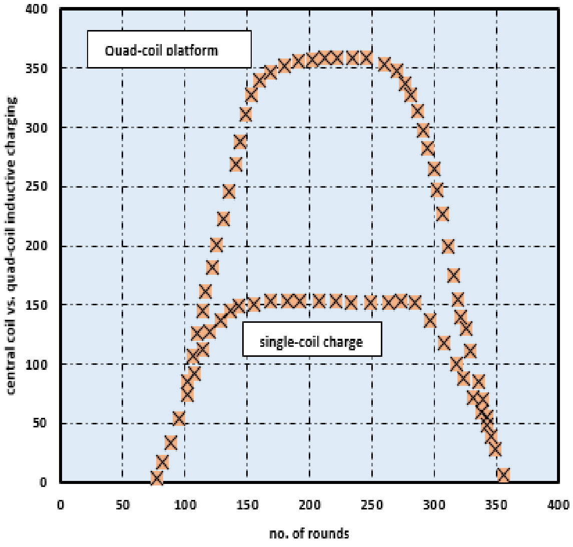


Fig. 24: quad-coil vs. single-coil power efficiency

References

[1] A. Al-Mhdawi and H. Al-Raweshidy, “Sdq-6wi: Software defined quadcopter-six wheeled iot sensor architecture for future wind turbine placement.” IEEE Access Journal, 2018.

[2] F. Hu, *”Network Innovation through OpenFlow and SDN - Principles and Design”*, 1st ed., F. Hu, Ed. 6000 Broken Sound Parkway NW, Suite 300: AUERBACH PUBLICATIONS, 455. [Online]. Available: www.auerbach-publications.com

[3] S. Azodolmolky, *”Software Defined Networking with OpenFlow”*. packt publishing, 2003, vol. 21.

[4] Nic Sharpley, “A few guidelines for selecting wind energy sites,” 2013. [Online]. Available: www.windpowerengineering.com

[5] Renewable First, “How windy does it have to be?” [Online]. Available: www.renewablesfirst.co.uk

[6] S. Park, H. Kim, K. Kim, and H. Kim, “Drone formation algorithm on 3D space for a drone-based network infrastructure,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2016.

[7] 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 Network*, pp. 56–61, Dec, 2017.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[9] Y. L. P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, “A West-East Bridge Based SDN Inter-Domain Testbed,” *IEEE Wireless Communications*, pp. 104–111, Feb, 2010.

[10] H. Ghafoor and I. Koo, “CR-SDVN: A Cognitive Routing Protocol for Software-Defined Vehicular Networks,” *IEEE Sensors Journal*, vol. 18, no. 4, pp. 1761–1772, 2017.

[11] H. Liu, X. Huang, L. Tan, J. Guo, W. Wang, C. Yan, and C. Xu, “Dynamic wireless charging for inspection robots based on decentralized energy pickup structure,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1786–1797, 2018.

[12] C. Wang, L. Zhang, Z. Li, and C. Jiang, “SDCoR: Software Defined

Cognitive Routing for Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 4662, no. c, 2018.

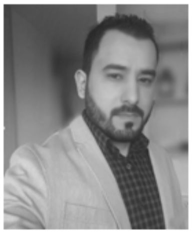
[13] D. K. N. Venkatramana, S. B. Srikantaiah, and J. Moodabidri, "SCGRP: SDN-enabled connectivity-aware geographical routing protocol of VANETs for urban environment," *IET Networks*, vol. 6, no. 5, pp. 102–111, 2017. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-net.2016.0117

[14] J. Shen, A. Wang, C. Wang, P. C. Hung, and C. F. Lai, "An Efficient Centroid-Based Routing Protocol for Energy Management in WSN-Assisted IoT," *IEEE Access*, vol. 5, pp. 18 469–18 479, 2017.

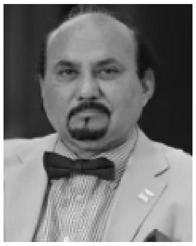
[15] S. Goyal and T. Chand, "Improved Trickle Algorithm for Routing Protocol for Low Power and Lossy Networks," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2178–2183, 2018.

[16] Arduino, "Ardunio board." [Online]. Available: https://www.arduino.cc/

[17] C. diagramz, "TP4056 Micro-USB Battery Charger Circuit Diagram," 2017. [Online]. Available: http://circuit-diagramz.com/tp4056-schematic-circuit-diagram/



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 ten years of experience in network engineering and telecommunications. His research interest include IoT, IIoT, software-defined networks, wireless sensor networks, delayed tolerant networks, renewable energy, and smart grid.



Hamed S. Al-Raweshidy 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.