A Softwarization Architecture for UAVs and WSNs as Part of the Cloud Environment

Sara Mahmoud, Imad Jawhar
College of Information Technology, UAEU
P.O. Box 15551, Al Ain, UAE
{201370014, ijawhar}@uaeu.ac.ae

Nader Mohamed
Middleware Technologies Lab.
P.O. Box 33186, Isa Town, Bahrain
nader@middleware-tech.net

***Abstract*—The development of Unmanned Aerial Vehicles (UAV) and wireless sensor network (WSN) applications depends on the availability of resources in UAVs and sensor nodes. In traditional applications, UAVs obtain the required data from specific sensors. However, this tightly coupled architecture restricts the usage of the infrastructure for specific applications. This paper proposes a softwarization architecture for UAVs and WSNs. In addition, the higher layers are proposed to be part of the cloud to benefit from the various cloud opportunities. The proposed architecture is based on the network softwarization concept and the Software Defined Networks (SDN) concept along with the Network Functional Virtualization (NFV) technologies. These concepts are associated with decoupling the hardware infrastructure from the control layer that virtualizes the infrastructure resources for the higher layers. The architecture is illustrated with an agricultural example for a cooperative system of a WSN and UAVs. We implement a prototype system that consists of a sensing node, UAV, WSN controller, UAV controller, and orchestration layer to provide a proof of concept for the proposed architecture.**

Keywords—WSN; UAVs; Softwarization; SDN; NFV; IoT; Cloud Computing

I. Introduction

UAVs and WSNs have applications in agriculture, monitoring, surveillance, and many areas. Usually, systems are developed for particular applications in a pre-determined network of UAVs and WSNs. However, these restricted systems limit the usage of the resources for other applications. In closed network applications, resources are tightly coupled and depend on specific devices. Thus, modifying the available resources becomes a nightmare for the system developers and may require shutting down the system during any maintenance. In addition, the system is fallible, which reduces the system reliability.

On the other hand, the recent paradigms of SDN and NFV represent the future of networks and services. SDN is associated with decoupling the control functions from the hardware using open interfaces, while NFV virtualizes the underlying functions and infrastructure to provide an abstract view of the network. These paradigms motivate the initiation of new research for utilizing the concept of softwarization to change the network without reinventing the network architecture.

This paper continues the work of our previous research. In [1], we discussed the opportunities and challenges of integrating UAVs with the cloud. Then, we presented the integration protocol of UAVs to the cloud as an IoT technology in [2]. Next, in [3] we proposed an architecture for accessing UAVs through a third party. This was followed by [4], where we proposed a cloud platform for developing UAV applications for UAVs.

This paper improves upon the previous research by presenting network softwarization for UAVs as well as other heterogeneous systems such as WSNs by decoupling the infrastructure of UAVs and WSNs from the control in a controller layer that virtualizes the infrastructure to the higher layers and provides an abstract view to the higher layer. Then, the orchestration layer manages the mission. This layer deals with abstract APIs to access UAVs and WSNs. As a result, the orchestration layer deals with the abstract services rather that the details of the devices' protocols. This supports the scalability of the system.

We will use an agriculture application as an example to illustrate the proposed architecture. In this application, a WSN is responsible for sensing the soil humidity and, as a result of this reading; the UAVs spray the agricultural land.

The paper is organized as follows: Section II is a literature review of motivation, background, and related work. Next, Section III compares the tightly coupled architecture and the loosely coupled architecture to show the advantages of the loosely coupled architecture. Section IV describes the considerations and opportunities gained from the SDN, NFV, and softwarization concepts. A detailed description of the architecture is presented in Section V. An implementation is provided to evaluate the proposed architecture in Section VI. Section VII concludes the paper.

II. Literature Review and Related Work

A. UAV and WSN Applications

There are many applications for UAVs such as the example presented by Varela et al. [5], where UAVs are used for environmental monitoring including collecting data on air quality in different layers of the atmosphere, as some information cannot be collected by ground systems due to gases or smoke from fires. In another example, Fausto et al. in [6] proposed an architecture for using UAVs and a Wireless Sensor Network (WSN) in agriculture applications. The researchers developed a system of collaborative UAVs to efficiently spray pesticides and fertilizers in agricultural areas that can be reached only with difficulty by humans without missing some areas in the spraying process, duplicating spraying areas, or spraying outside boundaries.

Mohammed et al. [7] referred to UAV applications for smart cities. They also discussed some business applications of UAVs such as in Amazon Prime Air for delivering products and their use for restaurant services [8].

These various application opportunities of UAVs have encouraged researchers and developers to focus more on improving efficient frameworks to develop UAV applications easily, especially for multiple distributed UAVs that cooperate with each other. Therefore, different architectures and communication protocols for collaborative UAVs have been developed.

B. *Background*

Cloud computing is a new paradigm for hosting and delivering services over the Internet. Some research has been carried out to utilize the Cloud for some UAV applications [9]. Video Exploitation Tools is an example of an SOA application for UAVs as implemented by Se et al. [10]. More investigations were conducted to explore smart objects such as sensors, actuators, and embedded devices connected to the Internet through the IoT [11]. The main focus of IoT is establishing network connectivity between smart objects and the Internet, while the WoT builds the application layer on top of the network [12]. Accordingly, the Web tools and protocols can be used for developing and interacting with these objects.

In the IoT field, Guinard et al. [13] proposed the REST architecture by defining an object as a server that provides its resources in an ROA. Guinard et al. used the web tools as a solution for the WoT. These researchers proposed two methods for accessing objects [14]. First, they connected devices to a smart gateway for measuring power consumption. In this approach, objects that have no direct Internet connectivity are connected to the smart gateway through other protocols such as Bluetooth and ZigBee. The second method is a direct access to wireless sensor networks, where each node is considered as a web server that has a uniform interface that the client applications access.

More recently, network research has been associated with SDN that decouples the control plane and data plane from each other [15]. The control plane interacts both with the higher layer and lower layer. With the higher layer interaction, the control plane provides a common abstracted view of the network, while the lower layer interaction direction, the control plane programs the forwarding behavior, using device-level APIs of the physical network equipment distributed around the network. Moreover, Network-Function Virtualization enables the network infrastructure to be virtualized as building block classes and functions managed by the orchestration layer [16].

C. *Related Work*

In the literature, some papers have addressed the SDN for IoT and WSN. Qin et al. [17] designed an SDN architecture for the IoT environment focusing on developing a multi-network controller that serves scheduling algorithms for heterogeneous heterogynous traffic patterns and network links. Caraguay et at. [18] surveyed the opportunities of developing IoT applications using SDN compared with the traditional architectures. Moreover, Jacobsson and Orfanidis [19] proposed an architecture for WSN based on the SDN principles. The SDN was used for building low-cost off-the-shelf hardware to achieve customized development. Jacobsson and Orfanidis discussed the reconfiguration of the network and used the collected information for several applications.

Gante et al. [20] discussed the use of SDN for WSN smart management. They proposed a framework in which a controller resides in a base station to gather information from the distributed nodes to define routing rules.

Although several investigations have adopted the SDN architecture for WSN, they are still at the early stages of developing a mature platform to develop applications on top of it. Our paper proposes an architecture that gains the benefit of SDN as well as NFV and softwarization. In addition, our proposed architecture extends the concept to include several types of physical resources that are managed by multiple controllers, so that the UAVs are separated from the WSNs and each has their specific tasks and services.

III. Tightly coupled vs. loosely coupled Architectures

In distributed systems, components access and interact with others to share and exchange data. System components can either be tightly coupled or loosely coupled. In tightly coupled systems, components and nodes have full knowledge about others with defined interactions and roles. This approach faces many limitations. It is difficult to reconfigure the network by adding or removing nodes from highly coupled systems. Dependencies increase the probability of failures in the tightly coupled systems, where a failure in one of the components affects the others due to interdependencies and data sharing. On the other hand, loosely coupled systems provide more flexibility in configuration. In this approach, components can have limited or no knowledge about other nodes. The component does not have to interact directly with the node that provides the data, since any component that satisfies the conditions may provide the required data or service.

The loosely coupled system makes it easier to reconfigure the network or modify the nodes with less effort. Adding new components does not require reconfiguring the whole system; instead, only the new component needs to be registered and reconfigured in the control plane. In addition, modifying an existing component or replacing it could be done at run time without affecting the system operation. Moreover, loosely coupled architectures support redundancy for components and nodes, which improves system reaction to failures and increases its robustness and reliability.

In IoT, smart objects are usually designed for a specific application in a coupled architecture where objects are configured together to perform some tasks in certain scenarios. However, this approach faces some difficulties of developing new applications as well as integrating current objects with other nodes.

This paper focuses on WSN as sensors that collect data from the environment, and UAVs perform actions accordingly on the real environment to accomplish a mission. The collected sensing data from the WSN affects the actions of UAVs. According to the running scenario, in a tightly coupled system, sensor nodes interact directly with UAVs as shown in Figure 1; therefore, each sensor node has full knowledge about the UAV address, communication protocol, commands, and other criteria. In addition, to add or remove UAVs, it is necessary to notify the WSN of the new state of the system, which adds more overload on the system besides the mission coordinating communication. Furthermore, the overhead of the decision-making should be considered, where the humidity sensors decide the need of spraying by UAVs, although the sensors' capability is not sufficient to handle decision-making processing.

In a loosely coupled system, the WSN and the UAVs do not interact directly with each other. However, a middle layer takes the control of managing the cooperation between the WSN and the UAVs. This layer separates the WSN layer from the UAVs; therefore, the WSN does not have a detailed knowledge about the UAVs.

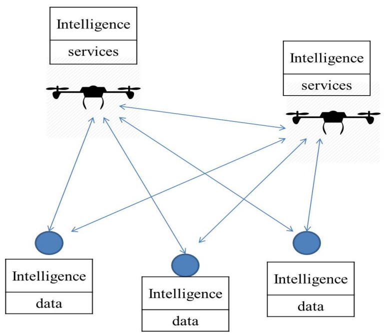

Figure 1 UAVs and WSN in a tightly coupled architecture.

IV. Considerations and Opportunities

Before developing UAVs and a WSN framework, some considerations should be taken into account. These considerations are summarized as follows:

A. *Considerations Related to UAVs and WSNs*

- **Limited capabilities**: Sensing devices in WSN have limited capabilities with respect to memory, processor capacity, and energy. Although UAVs have more powerful resources, they consume their internal resources more rapidly; therefore, they require lightweight software that does not heavily consume their resources.
- **Context perspectives**: The availability of some services depends on some contexts such as the device's location, energy level, or specific sensor readings. Therefore, if an available UAV is currently near the mission location, it is preferable to choose it rather than a similar UAV that is far from the specified location.
- **Real-time management**: UAV task allocation, mission management, and flight control algorithms should be provided for real-time execution and path planning management, which requires reliable communications.
- **Reliable connection**: These devices require continuous connectivity to the cloud so that they can access the cloud and their resources to be invoked when required. The assumption of a reliable connection is valid for operations in city areas such as smart cities where networks are available for 3G/4G/LTE connections.
- **Physical environment** considerations: The services provided by the UAVs are physical world services, thus they sense and affect the physical environment. UAV services that make changes in the environment such as spraying should be managed carefully, e.g. these services should not be duplicated over the same area. In case of a repeated request, there should be approval or acknowledgment before performing the service.

B. *SDN and NFV Opportunities for UAVs and WSNs*

The proposed architecture enables the SDN concept by separating the physical infrastructure layer from the control layer as well as gaining the benefit from the NFV architecture by virtualizing the infrastructure and its functions; this opens numerous opportunities for UAVs and WSN applications and development. In addition, the softwarization concept provides the service modeling.

SDN, NFV, and softwarization provide several opportunities for the architecture model. To demonstrate that, a description of the system is discussed below without each of them following the agriculture example:

- **Separating responsibilities**: Without the SDN, the control plane and the service plane reside on the same component. In this case, the WSN and UAVs have the intelligence systems on them. Hence, the sensor collects the humidity of the soil and decides the need for spraying, which overloads the sensors processing. As a result, by separating the control plane from the sensing devices, each component has separate and specific responsibility.
- **Virtualization:** This allows the higher layer to view the infrastructure as a single block system. WSN and UAVs can be homogeneous or heterogeneous in their operating systems, commands, communication, acting, sensing, storage, and processing capabilities as well as their energy levels. While applications that rely on homogeneous UAVs are easier to develop, heterogeneous UAVs can offer significant opportunities for providing cost-effective solutions for complex applications that require different capabilities for the various tasks involved. However, developing such applications for UAVs with heterogeneous devices, different energy levels, and varying storage, communication, sensing, and processing capabilities is a complex task without virtualization [21].
- **Abstraction**: Without virtualization, the higher layers interact with the components by specifying the device explicitly. Hence, the higher layers should have the knowledge of each device and its address with its communication protocol. On the other hand, with virtualization, the higher layers view the WSN as a network of sensors regardless of their real distribution.
  The virtualization allows the abstraction of heterogeneous components in the system to be viewed as homogenous components.
- **Modularity:** Softwarization utilizes the SDN and NFV and then provides the functions of the components as services so that the higher layer is re-programmed easily

to modify the network mission according to predeveloped modules.

- **Configurability:** Without softwarization, the control system is difficult to be re-configured. In the case of adding other types of components such as ground vehicles to the WSN and UAVs system, the softwarization architecture provides the ground vehicles' services and registers their descriptions to be added to the system easily. Then, the higher layers deal with the components as services.
- **Cloud advantages:** Because the control plane resides in huge servers or in the cloud, it utilizes the powerful processing and storage available. Allocating the control plane on the cloud provides several advantages such as 1) **Ubiquity**: The ubiquitous property of cloud computing that allows users to access the system from anywhere at any time. 2) **Elasticity**: The cloud has a huge and scalable infrastructure of processing power; the controller plane computations could be made on the cloud. Therefore, the reserved processing and storage of the higher layer are increased according to the current usage rather than reserving fixed processing servers. 3) **Cloud services**: Cloud computing provides ubiquitous services such as Google Earth 3D maps and computations that can be integrated with the system services to develop efficient applications. 4) **Cloud-level reliability**: As the controller resides on the cloud, it allows duplication on multiple servers, so that in case of controller failure on one server, it switches to another working server. 5) **Mobility**: Devices are required to connect to the cloud through any access point. This allows the devices to move to different places as long as they are connected to the cloud. This can be compared with the central station where devices are connected to a single point, which restricts the area of movements. 6) **Standardized communication protocols:** The cloud uses standardized communication protocols to request services and exchange data such as HTTP. Therefore, versatile nodes can use these standards regardless of their operating systems, programming languages, and commands. The standardized protocols make the application development easier on top of the platform.
- **Scalability**: Adding more UAVs or resources becomes easier by registering these devices to the platform as plug-and-play without affecting the application layers, so that devices are attached to the mission in the run time of the operation.
- **Component fault-tolerance**: In case of a component failure in the lower layer, it may be replaced with another similar component to perform the task without affecting the architecture or the flow of the mission.
- **Reusability**: The web service architectures support reusability so that the device resources are used for different applications according to their availability. For example, the agriculture humidity sensors could be used simultaneously for the spraying mission as well as for a soil quality mission that measures ability of the soil to reserve its water for a period of time.
- **Cost efficiency**: The reusability of the same component in different applications reduces the cost of owning the same component for each application, the CAPital EXpenditure (CAPEX) by reducing the hardware resources of multiple systems by reusability. In this case, it can be used easily without interfering between applications. In addition, the orchestration and management of the resources reduces the OPerational EXpenditures (OPEX) during the process.

V. Softwarization Architecture Layers

In the near future, cloud computing will extend the cloud infrastructure to include terminals in the real world. The new research of the Software Defined Network evolves the client side to act as servers and provide services in the physical world through well-defined APIs [22]. The softwarization architecture is shown in Figure 2.

A. Physical Resource Layer

1) WSN Infrastructure Layer

This layer is composed of the WSN and UAVs. First, the WSN includes tiny sensing nodes that are distributed in one or more regions of the area. A node consists of several components of a local controller that is a tiny processor responsible of controlling the inner processing of the node, wireless communication system, sensors, and energy source. Each node is registered to the WSN controller, then identified and allocated. They accumulate the required data from the physical environment and send it to the WSN controller. The WSN may be divided into groups according to the network area.

The WSN provides sensing services to the WSN controller in a loosely coupled design, as each node provides its sensing data from the physical environment as services accessed through APIs, regardless of other nodes or components. In this scenario, WSN nodes interact with the WSN controller when needed, so that it sends the reading data when the WSN controller requests it.

2) UAVs Infrastructure Layer

Second, UAVs are flying vehicles; their components include their payloads, internal memory, processor, and other resources. In this research, they are considered as actuators that respond to a certain action according to the WSN data. Each UAV provides several services that physically affect the real world. UAVs do not directly interact with WSN nodes; however, they respond to the interaction with the UAV controller through standardized protocols and APIs.

B. Controller Layer

The controller layer is responsible for monitoring the physical components and collecting data from them. This layer provides the abstraction view to the higher layer so that the WSN and UAVs are virtually accessed without specifying a certain component. This layer is composed of the WSN controller and the UAV controller. Newly added devices register themselves to their controller layer. Moreover, changes and re-configurations are centrally managed through the controller layer. This layer is also responsible for managing the heterogeneity of the components and communication protocol of the registered devices.

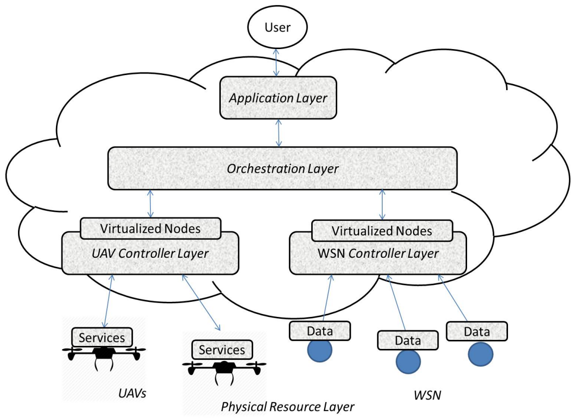

Figure 2 Softwarization of UAVs and WSN architecture.

1) WSN Controller

The WSN controller resides in the cloud side, and it is responsible for registering the sensors in its database to distinguish the available sensors from the ones in use. In addition, it deals with the low-level communication protocols where different sensors may have different protocols. Subsequently, the WSN controller provides virtualized sensing services to the higher layer in abstract APIs.The importance of this layer is to hide the heterogeneity of the sensor interfaces from the higher layers to be viewed as sensing services without being concerned about the sensors' configurations.

2) UAVs Controller

Similarly, the UAV controller resides in the cloud side and could be in the same or a different server. It provides virtualized services of UAVs in abstract APIs to the higher layer. It is also responsible for allocating the suitable available UAV for the required task requested by the higher layer.

C. Orchestration Layer

This layer exists on the cloud side; it is modeled as middleware to isolate the Physical Resource Infrastructure layer from the Application layer. It offers resources as services to the application layer. It allows integrating the cloud platform with the Physical Resource Infrastructure layer to implement powerful applications. The orchestration layer is responsible for organizing the flow of the tasks and services. This layer interacts with the Controller Layer that virtualizes the services of the physical world. As a result, the orchestration layer does not directly interact with the components.

D. Application Layer

The higher layer is the application layer, which usually has a user-friendly interface that allows the user to apply for a mission and receive the results and notifications of the workflow during the mission.

VI. Evaluation

The system was implemented using several layers as shown in Figure 2 and discussed in Section V. First, the infrastructure layer is composed of sensor nodes and UAVs. The sensors were implemented as shown in Figure 3 and Figure 4 using Arduinos as a processor along with a DHT sensor for humidity measurements and Adafruit CC3000 for wireless connectivity. A 9 volt battery was used as the power source.

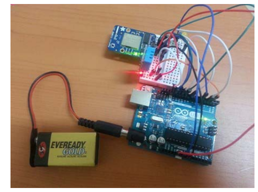

Figure 3 The implemented sensing node using Arduino and humidity sensor.

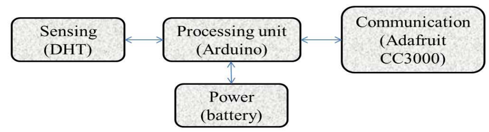

Figure 4 The conceptual view of a sensing node.

Then, an AR drone was used to implement the UAV physical layer. The AR drone has pre-programmed commands for controlling the UAV but has no spraying service. Pre-programmed commands were used instead to show the interaction between the UAV and the UAV controller server.

The controller layers were built in the NodeJS language for the WSN controller and the UAV controller. The controllers were installed in two different computers to simulate two servers. Devices and controllers were implemented in a local network for the purpose of simplicity. They were given local IP addresses. The WSN controller was installed in the first computer, and it provides an HTTP API for providing an abstraction view of the nodes using the GET operation with:

http://WSN_controller_address/service/humidity

This API requests a humidity reading from the WSN controller without directly interacting with the sensing node. Then, the controller communicates with the sensing node according to its APIs and communication protocol to request the humidity and returns the result to the requester. More specifications and requirements could be defined in a JSON object along with the HTTP request.

Next, the UAV controller was installed in the second computer and provides an abstract view of the UAVs' services. For example, a GET operation with the HTTP request:

http://UAV_controller_address/service/spray

This API requests a spraying service from the UAV controller, so that it interacts with the UAV to perform the service and returns the result to the requester. In our implementation, the UAV takes off, hovers around the area, and then lands. Similarly to the WSN controller, in order to define additional parameters, a JSON object could be provided along with the request. The orchestration layer was implemented in a mobile device to request the services from the controller layer. The orchestration layer requests an HTTP request for a humidity reading from the WSN controller, then receives a current value of the humidity level. Accordingly, it sends a spraying service request as an HTTP request to the UAV controller.

Although the HTTP APIs implemented in this prototype are GET requests, PUT requests could be used for security purposes to add authentication property in the request parameter. Therefore, only authorized users are allowed to request the service.

The response time of the HTTP requests from the orchestration to the WSN controller and the UAV controller were measured ten times as shown in Figure 5. It is observed that the response time for the humidity sensor is higher than the response time for the UAV. That is due to the differences in the processing capabilities of the devices, where UAVs have more powerful resources and faster processing than the sensor node.

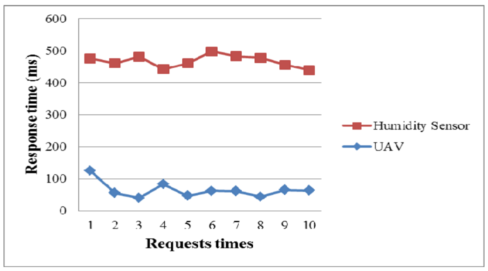

Figure 5 Response time for Humidity sensor and UAV through the WSN controller and the UAV controller respectively.

VII. Conclusion

The proposed architecture uses the recent paradigms of SDN and NFV to adapt a novel architecture of softwarization for UAVs and WSNs. This architecture eliminates the restrictions of tightly coupled architectures and utilizes the benefits of the recent research topic of the softwarization of network resources. This research takes advantages of the SDN and the NFV to propose a platform for the softwarization of UAVs and the WSN. This platform allows the flexibility of developing applications on top of the platform from the available infrastructure and services without reinventing the wheel due to the opportunities provided by these concepts. By decoupling the resources of UAVs and WSNs, the device is responsible for proving its services when invoked by the resource API regardless of the other devices' services. Therefore, the physical infrastructure layer is associated with collecting data and performing actions on the real world without being involved in decision-making or invoking other services. The system was implemented to provide a proof of concept using a sensing device and UAV along with their controllers to abstract the physical layer services. Then, the orchestration layer interacts with the controllers to request the required service.

References

[1] S. Mahmoud and N. Mohamed, "Collaborative UAVs Cloud," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, 2014, pp. 365–373.

[2] S. Mahmoud, N. Mohamed, and J. Al-Jaroodi, "Integrating UAVs into the Cloud Using the Concept of the Web of Things," *J. Robot.*, vol. 2015, 2015.

[3] S. Mahmoud and N. Mohamed, "Broker Architecture for Collaborative UAVs Cloud Computing," presented at the *The 2015 International Conference on Collaboration Technologies and Systems (CTS 2015)*, Atlanta, Georgia, USA, 2015.

[4] S. Mahmoud and N. Mohamed, "Toward a Cloud Platform for UAV Resources and Services," presented at *the IEEE 4th Symposium on Network Cloud Computing and Applications*, Munich, Germany 2015.

[5] G. Varela, P. Caamamo, F. Orjales, A. Deibe, F. López-Peña, and R. J. Duro, "Swarm Intelligence Based Approach For Real Time UAV Team Coordination In Search Operations," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 2011, pp. 365–370.

[6] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório, and P. A. Vargas, "The Use Of Unmanned Aerial Vehicles And Wireless Sensor Network In Agricultural Applications," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, 2012, pp. 5045–5048.

[7] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "UAVs for Smart Cities: Opportunities and challenges," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, 2014, pp. 267–273.

[8] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Opportunities and Challenges of Using UAVs for Dubai Smart City," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, 2014, pp. 1–4.

[9] C. E. Lin, C.-R. Li, and Y.-H. Lai, "UAS Cloud Surveillance System," in *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on*, 2012, pp. 173–178.

[10] S. Se, C. Nadeau, and S. Wood, "Automated UAV-Based Video Exploitation Using Service Oriented Architecture Framework," in *SPIE Defense, Security, and Sensing*, 2011, p. 80200Y–80200Y.

[11] L. Atzori, A. Iera, and G. Morabito, "The Internet Of Things: A Survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[12] S. Gustafson and A. Sheth, "Web of Things," *Comput. Now*, vol. 7, no. 3, 2014.

[13] D. Guinard, V. Trifa, and E. Wilde, "A Resource Oriented Architecture For The Web Of Things," in *Internet of Things (IOT), 2010*, 2010, pp. 1–8.

[14] D. Guinard, V. M. Trifa, and E. Wilde, Architecting A Mashable Open World Wide Web Of Things. *ETH, Department of Computer Science*, 2010.

[15] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, and others, "A Survey Of Software-Defined Networking: Past, Present, And Future Of Programmable Networks," *Commun. Surv. Tutor. IEEE*, vol. 16, no. 3, pp. 1617–1634, 2014.

[16] R. Jain and S. Paul, "Network Virtualization And Software Defined Networking For Cloud Computing: A Survey," *Commun. Mag. IEEE*, vol. 51, no. 11, pp. 24–31, 2013.

[17] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A Software Defined Networking Architecture for the Internet-of-Things," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, 2014, pp. 1–9.

[18] A. L. Valdivieso Caraguay, A. Benito Peral, L. I. Barona Lopez, and L. J. García Villalba, "SDN: Evolution and Opportunities in the Development IoT Applications," *Int. J. Distrib. Sens. Netw.*, vol. 2014, 2014.

[19] M. Jacobsson and C. Orfanidis, "Using Software-Defined Networking Principles For Wireless Sensor Networks," in *11th Swedish National Computer Networking Workshop (SNCNW), May 28-29, 2015, Karlstad, Sweden*, 2015.

[20] A. De Gante, M. Aslan, and A. Matrawy, "Smart Wireless Sensor Network Management Based On Software-Defined Networking," in *Communications (QBSC), 2014 27th Biennial Symposium on*, 2014, pp. 71–75.

[21] S. Hadim, J. Al-Jaroodi, and N. Mohamed, "Middleware Issues And Approaches For Mobile Ad Hoc Networks," in *The IEEE Consumer Communications and Networking Conf.(CCNC 2006)*, 2006, pp. 431–436.

[22] C.-S. Li and W. Liao, "Software Defined Networks [guest editorial]," *Commun. Mag. IEEE*, vol. 51, no. 2, pp. 113–113, 2013.