Software-Defined Wireless Network Architectures for the Internet-of-Things

Amr El-Mougy*, *Member, IEEE*, Mohamed Ibnkahla**, *Member, IEEE,* and Lobna Hegazy*
*{Amr.elmougy@guc.edu.eg, Lobna.alaaeldin@guc.edu.eg}
*Faculty of Media Engineering and Technology
*German University in Cairo, Cairo, Egypt
**mohamed.ibnkahla@queensu.ca
**Department of Electrical and Computer Engineering
**Queen's University, Kingston, Ontario, K7L3N6

***Abstract*—The Internet-of-Things (IoT) envisions a world where billions of everyday objects and mobile devices communicate using a large number of interconnected wired and wireless networks. Maximizing the utilization of this paradigm requires fine-grained QoS support for differentiated application requirements, context-aware semantic information retrieval, and quick and easy deployment of resources, among many other objectives. These objectives can only be achieved if components of the IoT can be dynamically managed end-to-end across heterogeneous objects, transmission technologies, and networking architectures. Software-defined Networking (SDN) is a new paradigm that provides powerful tools for addressing some of these challenges. Using a software-based control plane, SDNs introduce significant flexibility for resource management and adaptation of network functions. In this article, we study some promising solutions for the IoT based on SDN architectures. Particularly, we analyze the application of SDN in managing resources of different types of networks such as Wireless Sensor Networks (WSN) and mobile networks, the utilization of SDN for information-centric networking, and how SDN can leverage Sensing-as-a-Service (SaaS) as a key cloud application in the IoT.**

***Index Terms*—SDN, IoT, resource management, heterogeneity, WSN, ICN**

I. Introduction

The world is about to witness another technology revolution. Only a few years from now, the Internet is going to connect tens of billions of communicating "things" [1]. Devices connected to this Internet-of-Things (IoT) will be quite diverse in nature and will provide functions such as sensing, actuating, processing, and storing capabilities among many others. These communicating objects will interact with themselves in a machine-to-machine paradigm as well as with the users, supported by their increasingly sophisticated smartphones and other mobile devices, leading to a more pervasive and immersive Internet. This will foster a wide range of applications in fields such as home and industrial automation, real-time healthcare monitoring, optimization of public services, energy management, etc. Moreover, cloud computing architectures are expected to play a major role in leveraging some of the applications, services, and networks of the IoT.

The components of the IoT can be organized into 4 layers, as shown in Fig. 1. The first layer is the sensing layer, where all sensors, RFIDs, and Wireless Sensor Networks (WSN) exist. Information produced by this layer is collected by the aggregation layer (Layer 2). Different types of aggregators are possible depending on the sensing devices at the first layer. For example, WSNs typically have one or more sink nodes that collect data and upload them to the Internet. On the other hand, independently operating sensors may simply broadcast their data to the population of smartphones or specialized gateways that may exist in their regions. Aggregators either process the data directly or send them to other processing nodes at Layer 3. After data is processed, it can be uploaded to the cloud via an Internet connection (Layer 4), where it can be utilized by a large number of users.

Thus, it is clear that the IoT will be an extremely diverse networking environment in terms of both applications and devices. However, the most challenging problem of the IoT is heterogeneity, which is expected to exist on an unprecedented scope. For example, the sensing layer is expected to utilize different technologies such as Bluetooth Low Energy (BLE) and ZigBee [2]. At Layers 2 and 3, different transmission technologies may be used, such as 3G/4G and WiFi, to ensure persistent connectivity. To support these transmission technologies, network operators often utilize components from multiple vendors, complicating their management and reducing interoperability. Furthermore, operators and service providers are increasingly implementing network and server virtualization solutions in order to maximize the utilization of their resources, which introduces significant management problems. Moreover, challenges at Layer 4 include how to deploy services and components quickly and efficiently and how to optimize information delivery and maximize utilization of the Big Data produced by the IoT. Supporting fine-grained end-to-end Quality of Service (QoS) in such a sophisticated network is indeed a complicated task.

Software-defined networking (SDN) [3] is a paradigm that has emerged in recent years and is expected to hold the key for solving some of the aforementioned issues. The most important advantage of SDN is that it transforms the traditional "black-box" network components into "white-box" components that can be easily controlled and manipulated by

978-1-4673-6773-8/15/$31.00 ©2015 IEEE

the operator. SDN also enables centralized network management that allows efficient optimization and configuration. In short, the programmable features, which are the main drive behind the philosophy of SDN, introduces significant potential for addressing heterogeneity and interoperability in several layers of the IoT.

This article studies possible architectures and solutions for the IoT based on SDN. We discuss SDN solutions for all parts of the network that information might traverse on its end-to-end trip from the sensing source, through different types of mobile networks and the Internet core, until it reaches the user. In particular, we address three fields of IoT research: 1) SDN solutions for managing the sensing layer, 2) SDN architectures for end-to-end resource management over mobile networks and WLANs, and 3) SDN architectures for core network components that support Information-Centric Networking (ICN), which enables Sensing-as-a-Service (SaaS) as a cloud functionality.

II. Background: SDN Concepts and OpenFlow

One of the main limitations of today's networks is the difficulty of manipulating control functions, which are often hardcoded in the firmware of switches and routers. SDN introduces significant flexibility and manageability in the network by decoupling the control plane from the data plane [3]. All control functions (routing, security, load balancing, traffic isolation, etc.) are implemented in a software-based device called the controller, which sends instructions to one or more SDN-compliant switches or routers telling them how to handle packet forwarding (Fig. 2). This centralizes the control functionality of the network in a programmable device. Several controller implementations have been proposed in the literature such as NOX, POX, Floodlight, Beacon, and OpenDaylight [3]. Each implementation provides a base on top of which any application can be supported.

In addition, SDNs handle traffic as flows of packets. Thus, the controller sends flow policies to the switches and routers, which store them in a flow table. These flow policies specify rules that are applied to all packets between a particular source and destination. Communication between the controller and the switches and routers is based on an SDN protocol such as Forward and Control Element Specification (ForCES), Interface to the Routing System (I2RS), or OpenFlow [3].

OpenFlow is currently the most prominent of all SDN protocols. The latest versions are capable of supporting multiple flow tables at each node as well as advanced network protocols such as MPLS and IPv6 [3]. Using OpenFlow, a switching device receiving a packet checks the IP and MAC addresses to find a match in its flow table. If a match is not found, the packet is forwarded to the controller for further analysis. Using its software, the controller may generate a new policy to specify how to handle this packet and others that may belong to the same flow. This policy will then be forwarded to the switching device, where it will be saved as a new entry in the flow table. The benefits of implementing SDNs include:

- Providing centralized network management that allows easy optimization. This can be performed from remote sites using high speed connections.
- Network protocols can be easily adapted according to the network conditions. For example, load balancing can be easily managed due to the presence of the centralized controller. In another example, medium access can be adapted to network load, for example by switching from carrier-sensing to time division when the load increases.

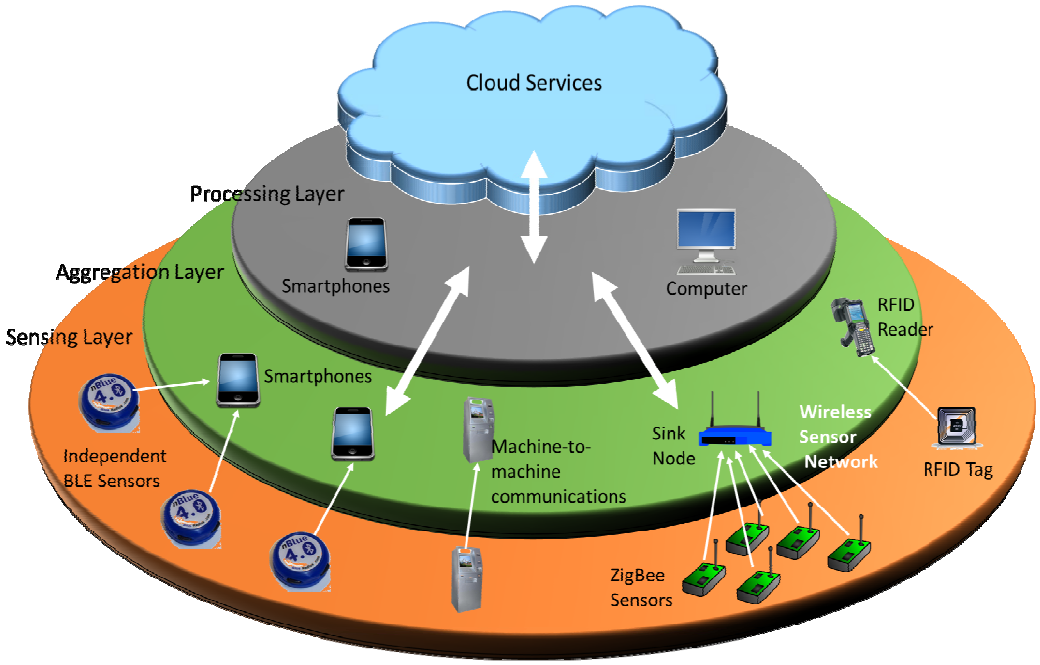

Fig. 1 Components of the IoT

- Providing vendor-independent control, since the protocols can be easily re-configured to change the operation of the switching devices. This means that network devices no longer have to be equipped with an exhaustive arsenal of protocols that suit every possible situation. It also allows management of heterogeneous entities.
- SDNs offer powerful tools for virtualization that allow administrators to maximize the utility of their devices. Using the centralized controller, network resources can be easily re-tasked according to changing conditions. Traffic isolation can be easily enforced by specifying packet flows. This is very advantageous for cloud service providers, since it allows them to manage their resources remotely. These virtualization tools can be easily extended to wireless networks to provide portable virtual machines.

Despite its advantages, SDNs have challenges that need more research. In particular, they were initially proposed for the control plane of wired networks. However, wireless networks have more sophisticated protocols, which complicates their control. Furthermore, scalability is an issue when the number of devices managed by a single controller increases. Moreover, the controller represents a single point of failure in the network.

III. SDN Solutions for the IoT

The most popular use of SDN technologies is in Data Center Networks (DCNs), where optimizing bandwidth consumption and resource utilization are the main goals. However, the IoT will be composed of multi-networks supporting applications with diverse requirements such as minimum delay and reliable delivery. In addition, device and application heterogeneity is much more challenging in the IoT.

In this section we will illustrate how SDN technology can be adapted to support the requirements of the IoT. First, we will discuss how SDNs can be used to manage sensing devices and networks such as WSNs. Then, we will discuss how SDNs can support end-to-end flows over heterogeneous networks. Finally, we analyze how SDN can be used for efficient information retrieval and delivery.

A. SDN Solutions for Managing the Sensing Layer

Typical sensing devices are based on simple hardware platforms that have limited processing and storage capabilities [4]. They also usually operate using small batteries and are required to remain operational for extended periods of time. Sensors implemented in end devices often use BLE technology since it is more suited for direct interaction with users. This is empowered by the popularity of the Bluetooth standard (it comes readily implemented in most smartphones). However, BLE has limited capabilities for forming ad hoc networks and its communication range is quite limited (often ~10m). Thus, in cases where long term monitoring, ad hoc networking, or longer range communication are required, sensors typically use ZigBee technology.

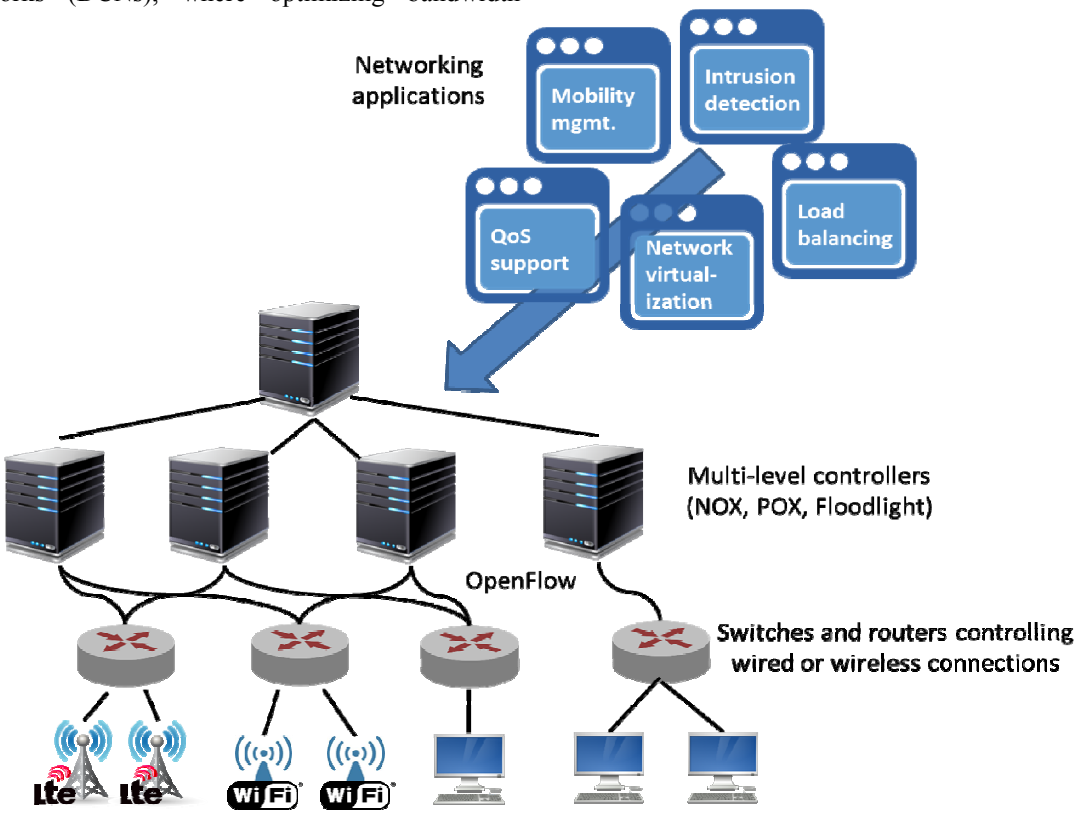

Fig. 2 Components of SDNs

SDNs can play a crucial role in management of independent BLE sensors or ZigBee-based WSNs. The SDN controller can be implemented at the sink node or aggregator points. The centralized position of the controller makes it suitable for optimizing topology control. Here, the controller can optimize the sleep/active cycles of the sensors by choosing the most energy-efficient set in every scheduling cycle. In addition, given a holistic network view, the controller can optimize routing decisions based on differentiated application requirements. Furthermore, if a sensor node has multiple sensors, the SDN controller can choose which ones to activate and which readings to report based on application requirements. This enables query-based communications between users of the IoT and the WSNs. Moreover, it allows WSNs to be used for multiple applications without the need for redesign. Moreover, by activating only the required components, significant energy savings can be achieved.

Nevertheless, the integration of SDN and sensor technology is far from straightforward. Neither BLE nor ZigBee are compatible with SDN standards such as OpenFlow. In addition, SDN standards mainly focus on specifying traffic flows based on controller software. They have no capability to control sensor hardware such as sleep cycle or the activation of individual sensors within a sensing device. This means that hardware modifications are needed to enable SDN control over sensing devices.

There has been increasing interest in this subject in recent years. For example, the authors in [5] propose a reconfigurable sensor node based on ultra-low power field programmable gate arrays (FPGAs) and microcontroller units (MCUs), as shown in Fig. 3. FPGAs are used for sensor interfacing as well as data aggregation, thus allowing full control and manipulation of sensors and transmitted data. Moreover, the MCU can offload heavy processing tasks to FPGAs such as encryption/decryption. In this reconfigurable WSN, the user defines scenarios and uploads them to the controller. Using these scenarios, the controller generates "roles" for individual sensors and sends them using over-the-air programming. The main drawback here is that communication between the base station and sensor nodes is not based on any well-known SDN standard such as OpenFlow, which might impose challenges when integrating this network with other SDNs.

Another solution proposes an SDN protocol for WSN management [6] that is based on flow tables, making it similar to OpenFlow. The proposed protocol is able to control functions such as medium access, localization, and tracking, making it more suitable for WSNs. The idea of extending OpenFlow for use in WSNs is further studied in [7]. Even though the proposed extensions could control the flow of data in WSNs, there is limited analysis of how the new versions of the OpenFlow protocols can be used to control sensor hardware.

B. SDN Architectures for End-to-End Resource Management

The previous section outlined how SDNs can be used to manage resources of a single WSN. However, when a user requests information from the IoT, it is likely going to be fetched from several WSNs and will probably traverse networks using different technologies such as WiFi and 3G/4G. Thus, resource management and provisioning across these heterogeneous networks is critical to ensure end-to-end QoS [8]. In addition, communications in the IoT might be opportunistic, as in the case of a smartphone coming within range of a BLE sensor; point-to-point, as in any client-server application; or multi-point to point, as in monitoring applications collecting information from a number of resources [9].

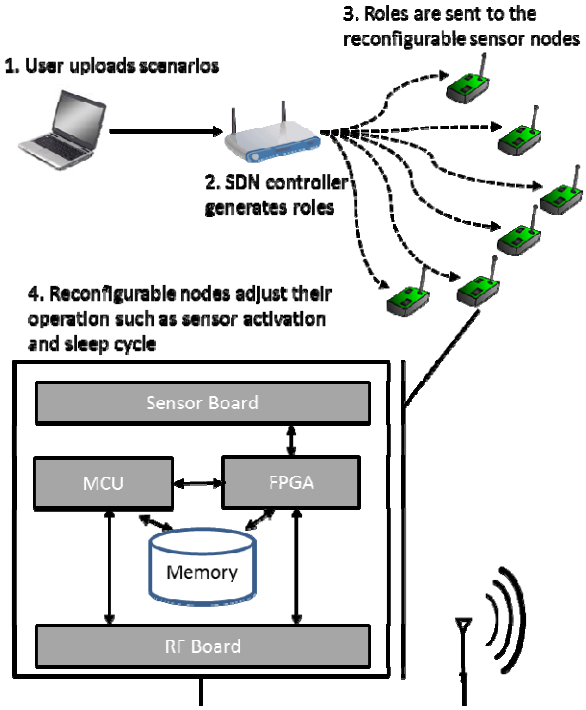

Fig. 3 Reconfigurable sensor node

The work in [10] proposes MobileFlow, which is an SDN architecture for carrier networks. Two main nodes are introduced into the architecture of LTE networks, namely MobileFlow Controller (MFC) and MobileFlow Forwarding Engine (MFFE). The MFC relies on network level abstraction to perform functions such as topology auto-discovery and resource viewing, and network resource monitoring and virtualization. It manages the resources of MFFEs, which are in turn responsible for data plane operations such as managing radio bearers and tunnel processing. Furthermore, they are responsible for forwarding data from the transport network. The MFFEs are envisioned to replace SGW/PGW in the future. This architecture allows operators to extend their resources and services easily and gives them the opportunity to utilize the services of multiple vendors. In addition, it allows efficient management of available network resources.

SDN approaches have also been adopted in managing access points in WLANs. For example, the work in [11] proposes an architecture for open campus WLAN based on software access points (SAE). These SAEs act as controllers of other access points in order to manage associations of users. In [11], this architecture was used for seamless mobility management and load balancing.

However, enabling SDN in WLANs has bigger advantages than just optimizing campus services. Future wireless networks will witness the deployment of a large number of small-sized cells such as femtocells and WiFi hotspots. Managing policies for data offloading and vertical handovers leads to efficient resource utilization. This approach was adopted in [12], which introduces an SDN architecture in the mobile backhaul that derives policies for data offloading to smaller cells. The SDN controller receives monitoring reports about the wireless conditions observed by the users and the applications they are running and makes handover decisions accordingly. This gives wireless operators powerful tools for improving QoS support.

Therefore, it is obvious that SDNs can provide solutions in different wireless networking paradigms. These solutions have great potential in the IoT as they give wireless operators significant flexibility in managing their resources and introducing new resources and services. SDNs also provides solutions for multi-vendor interoperability and technology heterogeneity in the IoT. For example, SDNs can be involved in designing protocols for reliable delivery of sensor data via smartphones and aggregator points that may be utilizing WiFi or mobile Internet access.

Nevertheless, the above approaches do not provide an integrated vision that includes service providers and core transport backbones. If these parts of the network are not considered, then an SDN architecture might simply be shifting the bottleneck from one part of the network to the other. In a wide area multi-network such as the IoT, where billions of devices are expected to be connected, this cannot be afforded.

Such an integrated vision is proposed in [13]. In this architecture several heterogeneous radio access networks (RANs) coexist and are used by multiple operators (see Fig. 4). The RANs are assumed to be equipped with programmable devices that can understand SDN instructions (ex: based on OpenFlow). In addition, the core network is assumed to be composed of SDN-compliant switches and routers.

The SDN controller (which may have several instances in the network) is logically centralized between all network entities, namely service providers, backbone, and RANs. The controller communicates with the operators (some of them may be virtual) in order to generate policies for resource sharing according to network load. This will result in more efficient service level agreements (SLAs). Moreover, the controller communicates with service providers to allow them to influence how their traffic is being handled by the network. These two interfaces are called northbound interfaces.

In addition, the controller interacts with the backbone to implement policies for resource sharing and adaptation according to network conditions. It also interfaces with the wireless RANs in order to allow virtualization of access networks and enforce SLAs that operators may have with their users. The final interface is with the users themselves to allow more programmability, for example to improve mobility management. These are called southbound interfaces.

This architecture allows joint optimization of RANs and backbone networks in order to improve QoS. Furthermore, it extends programmability from service providers down to the users, allowing quick and simple service deployment and adaptation as well as fine-grained mobility management. In the scope of the IoT, this can mean detailed service discovery, better connectivity, and optimized QoS support for differentiated services.

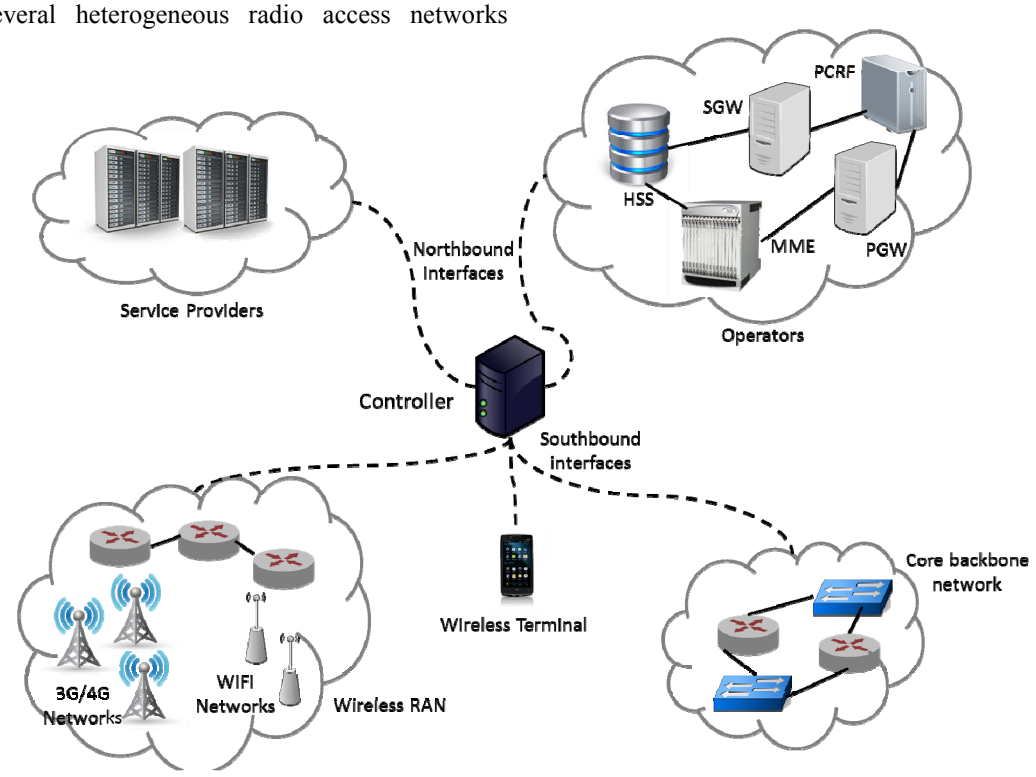

Fig. 4 Architecture for software-defined wireless network

C. *Enabling Sensing-as-a-Service (SaaS)*

This section attempts to answer a fundamental question in the IoT: how will sensor data eventually be delivered to the users? SaaS [1] is a cloud paradigm that is expected to exist on top of the IoT infrastructure in order to facilitate sensing services in a wide area network. SaaS also provides a business model that allows organizations or individuals to sell sensing services. However, this model cannot succeed if the only tool available for mining sensing data is traditional keyword-based search engines. These search techniques are not capable of capturing sensor characteristics or the context of the application. For example, several applications may require readings from temperature sensors. However, one application may require 1000 accurate readings from a specific area, while another may require less accurate samples dispersed over wider geographic locations. Clearly, keyword search techniques will be severely inadequate in satisfying the needs of both applications.

Several middleware solutions exist for facilitating SaaS [1]. For example, GSN is a platform that integrates sensor data and allows users to insert queries to find appropriate lists of sensors. Moreover, OpenIoT is an open source platform for managing IoT cloud resources such as sensors and actuators. However, both these platforms require manual data entry and do not deal efficiently with the problem of sensor heterogeneity. This problem is addressed in the Semantic Sensor Network (SSN) [1] project which uses ontologies [14] to design a linked database of sensors. It is worth noting that there is growing interest in using ontologies in the IoT due to their inference capabilities and their ability to add semantics. The model proposed in [1], known as CASSARAM, adds context-awareness to SSN in order improve the accuracy of sensor search techniques.

However, two key challenges exist in enabling SaaS. The first challenge is how to translate application requirements into sensing services that can be located in the IoT [9]. The second challenge is how to deliver the appropriate content to the users. Given that tens of billions of sensors will be available, the traditional networking model of connecting hosts will not work. Rather, it is more desirable to have loosely coupled user devices and content providers (sensors). Thus, information-centric networking (ICN) [15] will be more appropriate for use in the IoT. ICN is based on the observation that users are usually interested in acquiring particular information rather than who generated this information. This is even truer in the IoT. ICN addresses this issue by separating the control functions that match user interests to actual content from data forwarding functions.

The last statement illustrates how the philosophy of SDN lends itself quite naturally to the requirements of ICN in the IoT. A programmable controller can provide a solution for the first challenge of translating application requirements into existing resources as it will have two interfaces, one to the applications and the other to network resources. Software running on the controller can match the requirements to the existing resources and generate flow rules to forward the information to the appropriate destination, thereby solving the second challenge as well. This architecture can satisfy the requirements of one-time query-based information retrieval, as well as publish/subscribe paradigms, where the user declares interest in a particular type of data and the SDN controller will specify flow rules so that switches and routers constantly forward this data to the user. In addition, this architecture enables dynamic rerouting and efficient multicasting. The architecture is shown in Fig. 5.

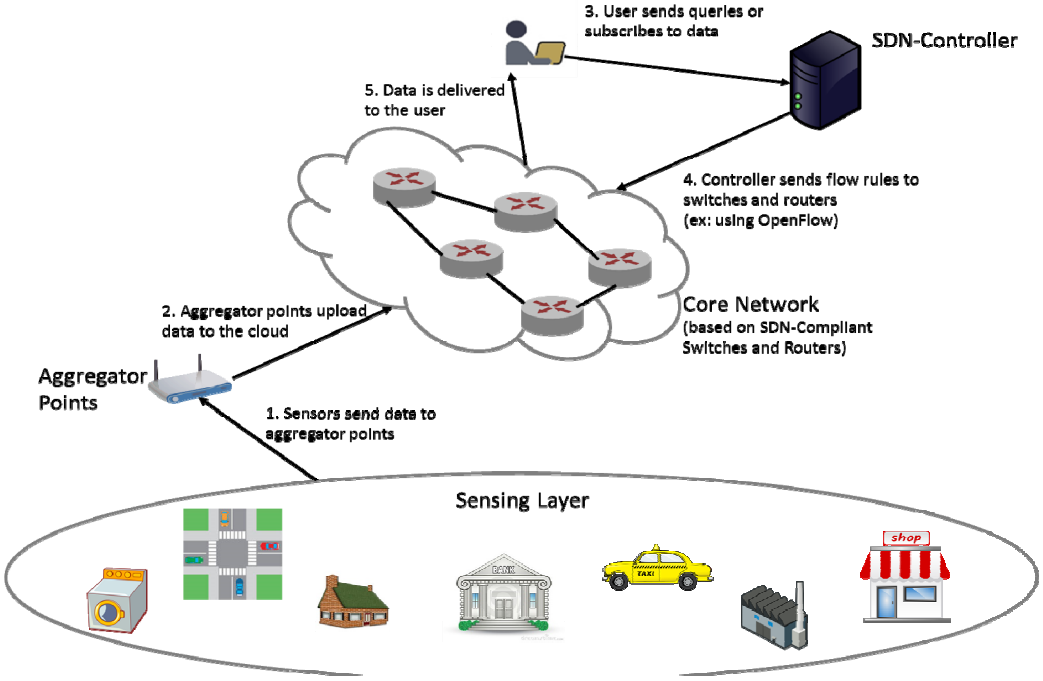

Fig. 5 SDN-based ICN architecture for the IoT

There are several architectures proposed for realizing ICN. An interesting survey can be found in [15]. Some of the ideas include routers that can examine application layer data and content brokers, which act as intermediaries between content providers and actual routers. However, to realize any ICN architecture, the format of data packets must be modified to include names (labels) for data in addition to traditional IP addresses. For example, the Data Oriented Network Architecture (DONA), Network of Information (NetInf) architecture, and the Publish/Subscribe Internet Technologies (PURSUIT) architecture [15] all propose a naming scheme in the form of *P*:*L*, where *P* is a cryptographic hash function of the owner's public key, and *L* is a globally unique label chosen by the content provider. On the other hand, the Named Data Networking (NDN) architecture [15] proposes a hierarchical naming scheme to represent content in human-friendly language.

These naming schemes were proposed for general Internet content. Their main drawback is that they do not include any semantics that can be used to link data. An attempt to address this was made in [16], where a naming scheme based on spatial indexing was proposed for SDN-based publish/ subscribe systems. Here, a bit string was used to allow subscribers to indicate fine-grained requests (ex: temperature readings in a particular range and from a particular location). However, the bit string has to be of fixed length to allow fast filtering at the routers and switches. If we consider, the virtually limitless applications and data types expected in the IoT, the fixed length limitation may be too prohibitive. Thus naming data is an area that requires further research in the IoT. Perhaps a new naming scheme is required that can include semantics describing and linking sensor characteristics. This naming scheme can then be coupled with existing models such as SSN and CASSARAM to include context-awareness.

Another challenge in SDN-based ICNs is responsiveness, which is the time taken for a controller to fully update the network topology after the events of new, modified, or deleted subscriptions. In any of these events, the controller may need to send new flow rules to the switches or modify/delete existing ones to update the forwarding tables. As the number of switching devices supported by the controller increases, the delay in updating the network topology may significantly increase. This may have a drastic impact on QoS and the ability of the network to support dynamic requests. In addition, this latency may lead to inconsistency problems, where some switches have an updated network view while others have not yet received the updates. This may lead to routing loops, black holes, and other problems.

In order to address this scalability issue, a hierarchical SDN architecture was proposed in [17]. In this architecture, the network is divided into areas, where a controller is responsible for each area. Then, all area controllers are managed by a second tier controller, which will then have a fully centralized network view. The area controllers do not communicate with each other. Computer simulations showed that this architecture is capable of managing a larger number of requests than a single centralized controller. Nevertheless, the average delay in updating the network and the consistency problems were not evaluated.

The consistency problem was addressed in [18], a distributed control plane was proposed for SDN publish/ subscribe networks. First, scalability is achieved by utilizing multiple controllers in a multi-tiered architecture. Two approaches are proposed: one with full cooperation between controllers and another with no cooperation at all. In order to ensure consistency, the controllers identify the operations that may lead to conflicts and serialize their execution. Performance results indicate that the latency in updating the network decreases as the number of controllers increase. Simulations also show that the proposed control plane is able to improve throughput.

D. End-to-End SDN Solutions

The previous sections have demonstrated that the SDN philosophy can be applied to different parts of the IoT to improve flexibility and maximize resource utilization. However, due to the diverse nature and requirements of the IoT layers (see Fig. 1); it is unlikely that we will see a unified end-to-end SDN solution across all layers. However, it is quite possible for SDN solutions to extend beyond the boundaries of one layer. For example, extending the ICN philosophy to SDN-controlled sensor networks could lead to improved management of sensing events and better resource utilization. Similarly, using ICN in SDN-controlled mobile networks could lead to more fine-tuned resource assignment.

In addition, the presence of a centralized controller introduces potential for incorporating semantics in sensor networks. Controllers could be responsible for storing detailed sensor attributes and providing a translation service for sensing requests similar to the one provided by the domain Name System (DNS) for host addresses. This can be pivotal for enabling SaaS as a worldwide service. Note that this sensor translation service cannot be supported by DNS since it requires real IP addresses, which the sensor attributes will not have. As mentioned before, the work in [16] has already attempted to utilize the controller for incorporating semantics. However, more work is needed to provide a more generalized translation scheme.

Finally, it is worth noting that SDN has significant potential in solving some of the heterogeneity problems of the IoT. Given a unified protocol for communication between the controller and the host devices (this protocol can be OpenFlow or something else), SDN can provide a common aggregation layer for different types of networks such as sensor networks, WLANs, and mobile networks. However, such a protocol would require standardization. In addition, an aggregation layer of this scale would have to be distributed, which requires more research on responsiveness and consistency in large hierarchical SDN architectures.

IV. CONCLUSIONS

This paper presented a study of some promising solutions based on SDN for the IoT. The areas addressed included management of WSNs, end-to-end resource management in wireless networks including core transport backbone networks

and service providers, as well as ICN as an enabler for SaaS. Even though SDNs provide promising solutions to some persistent challenges in the IoT, there are some open problems that require addressing. Particularly, scalability is an important issue since IoT is fundamentally a wide area network. Some of the main advantages of SDN architectures arise from the fact that the controller is logically centralized. However, in reality, multiple instances of the controller will have to be realized. This introduces significant management and synchronization challenges. In addition, security and privacy are important issues that may not be easy to solve due to the limited capabilities of some nodes, which might mean that sophisticated cryptography techniques are not feasible. Furthermore, there is a need to standardize SDN protocols, especially at the sensing layer. Finally, there is great need for a semantic naming architecture for realizing ICN and thereby SaaS.

REFERENCES

[1] C. Perera, A. Zaslavsky, C. Liu, M. Compton, P. Christen and D. Georgakopoulos, "Sensor search techniques for sensing as a service architecture for the Internet of Things," *IEEE Sensors Journal,* vol. 14, no. 2, pp. 406-420, 2014.

[2] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context-aware computing for the Internet of Things: a survey," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 1, pp. 414-454, 2014.

[3] F. Hu, Q. Hao and K. Bao, "A survey on software-defined network and OpenFlow: from concept to implementation," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 4, pp. 2181-2206, 2014.

[4] M. Ibnkahla, Wireless Sensor Networks: A Cognitive Perspective, CRC Press, 2012.

[5] T. Myazaki, S. Yamaguchi, K. Kobayashi, J. Kitamichi, S. Guo, T. Tsukahara and T. Hayashi, "A software defined wireless sensor network," in *International Conference on Computing, Networking and Communications (ICNC)*, pp. 847-852, 2014.

[6] A. D. Gante, M. Aslan and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *IEEE Queens University Biennial Symposium on Communications (QBSC)*, pp. 71-75, 2014.

[7] T. Luo, H.-P. Tan and T. Quek, "Sensor OpenFlow: enabling software-defined wireless sensor networks," *IEEE Communications Letters,* vol. 16, no. 11, pp. 1896-1899, 2012.

[8] A. El-Mougy, M. Ibnkahla, G. Hattab and W. Ejaz, "Reconfigurable Wireless Networks," *To Appear in Proceedings of the IEEE,* pp. 1-34, 2015.

[9] Q. Zhijing, G. Denker, C. Giannelli and P. Bellavista, "A software-defined networking architecture for the Internet of Things," in *IEEE Network Operations and Management Symposium (NOMS)*, pp. 1-9, 2014.

[10] K. Pentokousis, Y. Wang and W. Hu, "Mobileflow: towards software-defined mobile networks," *IEEE Communications Magazine,* vol. 51, no. 7, pp. 44-53, 2013.

[11] T. Lei, Z. Lu, X. Wen, X. Zhao and L. Wang, "SWAN: an SDN-based WLAN campus framework," in *IEEE Conference on Wireless Communications, Vehicular Technology, Information Theory, and Aerospace and Electronic Systems (VITAE)*, pp. 1-5, 2014.

[12] M. Amani, T. Mahmoodi, M. Tatipamula and H. Aghvami, "Programmable policies for data offloading in LTE network," in *IEEE Conference on Communications (ICC)*, pp. 3154-3159, 2014.

[13] C. Bernardos, A. De La Oliva, P. Serrano, A. Banchs, L. Contreras, H. Jin and J. Zuniga, "An architecture for software defined wireless networks," *IEEE Wireless Communications,* vol. 21, no. 3, pp. 52-61, 2014.

[14] A. El-Mougy, A. Kamoun, M. Ibnkahla, S. Tazi and K. Drira, "A context and application-aware framework for resource management in dynamic collaborative wireless M2M networks," *Journal of Network and Computer Applications,* vol. 44, pp. 30-45, 2014.

[15] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros and G. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 2, pp. 1024-1049, 2014.

[16] M. A. Tariq, B. Koldehofe, S. Bhowmik and K. Rothermel, "PLEROMA: a SDN-based high performance publish/subscribe middleware," in *ACM International Middleware Conference*, pp. 217-228, 2014.

[17] S. Gao, Y. Zeng, H. Luo and H. Zhang, "Scalable area-based hierarchical control plane for software defined information centric networking," in *IEEE Conference on Computer Communication and Networks (ICCCN)*, pp. 1-7, 2014.

[18] S. Bhowmik, M. A. Tariq, B. Koldehofe, A. Kutzleb and K. Rothermel, "Distributed control plane for software defined networks: a case study using event-based middleware," in *To appear in the ACM Conference on Distributed Event-Based Systems*, pp. 1-12, 2015.

[19] A. Lara, A. Kolasani and B. Ramamurthy, "Network innovation using OpenFlow: a survey," *IEEE Communications Surveys and Tutorials,* vol. 16, no. 1, pp. 493-512, 2014.