SDIoT: A Software Defined based Internet of Things framework

Article *in* Journal of Ambient Intelligence and Humanized Computing · August 2015

DOI: 10.1007/s12652-015-0290-y

Yaser Jararweh
Jordan University of Science and Technology
172 PUBLICATIONS **986** CITATIONS
SEE PROFILE

Mahmoud Al-Ayyoub
Jordan University of Science and Technology
157 PUBLICATIONS **1,071** CITATIONS
SEE PROFILE

Ala Darabseh
New York University Abu Dhabi
13 PUBLICATIONS **144** CITATIONS
SEE PROFILE

A.ndrew Rindos
IBM
48 PUBLICATIONS **468** CITATIONS
SEE PROFILE

J Ambient Intell Human Comput
DOI 10.1007/s12652-015-0290-y

ORIGINAL RESEARCH

SDIoT: a software defined based internet of things framework

Yaser Jararweh[1] · Mahmoud Al-Ayyoub[1] · Ala' Darabseh[1] · Elhadj Benkhelifa[2] · Mladen Vouk[3] · Andy Rindos[4]

Received: 15 December 2014 / Accepted: 20 February 2015

Abstract The internet of things (IoT) represent the current and future state of the Internet. The large number of things (objects), which are connected to the Internet, produce a huge amount of data that needs a lot of effort and processing operations to transfer it to useful information. Moreover, the organization and control of this large volume of data requires novel ideas in the design and management of the IoT network to accelerate and enhance its performance. The software defined systems is a new paradigm that appeared recently to hide all complexity in traditional system architecture by abstracting all the controls and management operations from the underling devices (things in the IoT) and setting them inside a middleware layer, a software layer. In this work, a comprehensive software defined based framework model is proposed to simplify the IoT management process and provide a vital solution for the challenges in the traditional IoT architecture to forward, store, and secure the produced data from the IoT objects by integrating the software defined network, software defined storage, and software defined security into one software defined based control model.

Keywords Internet of things (IoT) · Software defined network (SDN) · Software defined systems (SDSys) · Software defined storage (SDStore) · Software defined security (SDSec)

1 Introduction

Nowadays, most people, if not all, complete their needs, work, or even transactions through the Internet. To achieve this, they need to interact with many devices or objects. Moreover, objects might need to communicate with each other. Such communication between humans and objects (things) requires connecting the objects around us with the Internet. The internet of thing (IoT) paradigm, as mentioned by Gubbi et al. (2013), reflects the current and future situation of the world. Indeed, IoT researchers argue that by 2020, IoT will grow significantly to cover all the objects in our environment creating what they call the internet of everything (IoE).

The rapid growth of the IoT produces a lot of data and information collected by the huge number of objects connected to it. Storing, managing, controlling and securing such big data are considered critical issues if we want to connect everything to the Internet in a useful and practical manner. Moreover, in real time transaction or any simple work there is a need to connect these objects with each

✉ Yaser Jararweh
yijararweh@just.edu.jo

Mahmoud Al-Ayyoub
maalshbool@just.edu.jo

Ala' Darabseh
cis.alaa@gmail.com

Elhadj Benkhelifa
e.benkhelifa@staffs.ac.uk

Mladen Vouk
vouk@csc.ncsu.edu

Andy Rindos
rindos@us.ibm.com

[1] Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan

[2] Mobile Fusion Applied Research Centre, Staffordshire University, Stafford, UK

[3] North Carolina State University, Raleigh, NC, USA

[4] IBM Corporation, Research Triangle Park, NC, USA

Published online: 23 June 2015

others to accomplish the desired work. Any delay in response time through the communication will negatively affect on the overall performance and accuracy the system. So finding ways to accelerate the communication process is also considered a hindering point to the IoT acceptance and growth. The software defined systems (SDSys) is considered a vital solution for these challenges. Since, the main goal for SDSys is to hide all the complexities of the management and control functionality of the system resources from the end users. In this work, we build a framework for software defined internet of things (SDIoT) that exploits several SDSys such as software defined network (SDN), software defined storage (SDStore), and software defined security (SDSec).

As the following examples illustrate, researchers have already proposed to use such SDSys in association with IoT networks.

Huang et al. (2014) proposed a framework to manage and control the IoT network by exploiting the benefits of SDN. Their framework focused on the machine to machine (M2M) transactions. Using SDN, the framework eliminated the rigidity in the traditional IoT network by allowing it to respond to any change in the environment dynamically, even if the network gateway broke or failed for some reasons TalebiFard and Leung (2014).

Stefan Bernbo, the CEO and founder of Compuverde, talked about the need to restructure and redesign the data storage systems to be able to deal with the massive amount of data produced by objects, things, in IoT networks as mentioned by Bernbo (2014). He discussed some problems in traditional storage appliances which are considered critical holes to accommodate this data, and showed how the intelligence in the software defined storage (SDStore) is able to address all of these challenges.

Finally, for many IoT applications, security concerns are one of the major problems hindering their wide acceptance. While there exist solutions derived from classical networking literature, there exist no solutions based on the software defined security (SDSec) principles, to the best of the authors' knowledge.

Despite the above efforts, there exist no prior research on building a comprehensive software defined solution for IoT networks by integrating different SDSys such as SDN, SDStore and SDSec with the IoT technology. In this work, we aim at addressing this issue by building a comprehensive software defined solution for IoT network called the software defined internet of things (SDIoT) framework.

The rest of this paper is structured as follows. In Sect. 2 we talk about SDN and discuss some research works that capture its idea. Section 3 presents SDStore and discuss some real SDStore systems. Whereas, in Sect. 4 the idea of SDSec is explained and some real SDSec systems are discussed. After that, a brief introduction about existing solutions that used the idea of SD paradigm to control the IoT network is given in Sect. 5. Our proposed framework is explained in Sect. 6. Finally, we conclude this paper and present our future plans in Sect. 7.

2 Software defined network (SDN)

SDN is the latest innovation in network environment. It simplifies the network management by separating the control plane from the data plane, where the data plane uses the forwarding tables prepared by the control plane in the controller to forward the messages, flow-packets as mentioned by Jain and Paul (2013).

The controller plays an essential role in network control operations. It resides between the network applications and its forwarding elements. All the applications in the network system need to communicate with forwarding elements. They reside in data layer, and the controller serve as a middleware to transfer and mange the communication. Several APIs which are located between SDN applications and the system controller are used to facilitate the communication process and transfer information between them. On the other hand the controller can interact with the forwarding elements through network protocols such as the OpenFlow protocol.

Several research works have been published on SDN. Dixon et al. (2014) discussed most of SDN aspects and illustrated how this paradigm can support the Software Defined Environments (SDE). In addition, they showed the vision of IBM to consolidate the SDN idea by integrating their IBM SDN virtual environments (SDN-VE) product with the Neutron, OpenStack network platform[1] to extend SDN-VE feathers.

Another comprehensive work was done recently by Hu et al. (2014) of the University of Alabama, USA. It is emerged to cover most of SDN/OpenFlow aspects which range from the concept to SDN solution deployment. The motivation behind this survey was introduced to show how SDN work and how are they built to support several organizations by facilitating the control and management operations and enhancing their performance with lower cost. In addition, the survey shows how the work can be done quickly by distributing and virtualizing the workload across several hardware components. Furthermore, cloud computing providers can exploit the benefits of SDN to manage the heterogeneity in switches/routers infrastructure. Different vendors have different switches with different characteristics. So, instead of managing and customizing each switch separately, SDN provides the ability to manage all switches devices by a single enforcement point, the SDN control layer. Also it gives the

[1] http://www.openstack.org/.

cloud user the ability to use the cloud resources in an efficient way by creating slices/slivers and let the data flow in a transparent way.

Nunes et al. (2014) discussed two different architectures for SDN, the OpenFlow and ForCES. The differences between them revolve around system architecture, design, forwarding models, and interface protocols. Despite the differences between them, both of them follow the same SDN principles of separating the control plane from data plane.

Jarraya et al. (2014) proposed a taxonomy to simplify the understanding of SDN concepts and different related domains. The hierarchy of this taxonomy classifies SDN related problems and their solutions across several layers; application layer, control layer, and forwarding layer. In addition, they considered inter-layers problems and solutions such as application-control, control-infrastructure, and control-application-infrastructure layer. Many research works are presented to study and analyze relevant issues which arose from the emergence of SDN and cover its implemented solutions and propose some modification to enhance these solutions and increase the SDN adaptation. At the end, some issues that are still open were exhibited to draw the attention of the researchers and graduate students to work on them. These issues revolve around compatibility, security, and interoperability of the SDN.

3 Software defined storage (SDStore)

In traditional data storage systems, specially large data storage like data centers, that store a huge amount of data and exploit virtualization to expand the system. The data forwarding, processing and management processes occur at the same place, infrastructure assets, which increases the burden on the underling devices and subsequently reduces the system performance. Software defined storage (SDStor) was proposed to facilitate and simplify such complexity, and at the same time, maintain an acceptable level of QoS Wu and Sun (2013). SDStor takes the responsibility of managing huge data in storage systems by isolating the data control layer from the data storage layer. The control layer refers to the software component that manages and controls the storage resources, whereas the data layer refers to the underling infrastructure of the storage assets Palanivel and Li (2013).

Many corporations realize the benefits of SDStore and apply it in their storage centers. Examples include EMC Corporation, which launched ViPR software as an implementation for SDStore ViP (2015), IBM with Storwize software IBM Corporation (2014), and many others. All of them define SDStore based on their own perspectives using different terms as discussed below.

IBM launched its own novel virtualized storage, IBM Storwize, as SDStore solution to support and complement their software defined environment Crump (2013). Storwize provides a scalable, flexible, virtualized storage management solution for the cloud environments Systems and Technology Group (2014). It has many functions to support the virtualized environments and help the enterprises to manage their huge data growth in an efficient manner. Storwize family provides several storage solutions that can be deployed easily by different size business storage systems.

In addition to Storwize and ViPR solutions, still, there are many proposed, implemented, or deployed SDStore solutions like Maxta,[2] HITACHI,[3] Datacore,[4] CloudBytes,[5] IBM SmartCloud,[6] etc.

4 Software defined security (SDSec)

It is illogical to follow traditional security mechanisms with the new technology paradigms like SDN and SDStor. For that, the Software defined security (SDSec) which is an example of network function virtualization (NFV) is emerging. The new technology works and provides a new way to design, deploy and manage the security by separating the forwarding and processing plane from security control plane, is similar to the way that SDN abstract the forwarding plane from control and management plane Vizardl (2013). Such separation provides a scalable distributed security solution, which virtualizes the security functions but remains manageable as a single logical system.[7]

SDSec was proposed as a solution to help secure virtualized environment infrastructures, including virtual network, virtual storage and even virtual servers from different threats whether they are traditional such as intrusion detection and denial of service attacks or specific to virtualized environments such as insider threats Yaseen et al. (2013) and Almodawar et al. 2013).

The idea behind the SDSec concept appeared at the cloud security alliance (CSA)[8] as they sought to find a new approach for security with lower costs Vizardl (2013). To transfer their vision into reality they launched the software

[2] http://www.maxta.com/.

[3] http://www.hds.com/solutions/it-strategies/storage-virtualization/.

[4] http://www.datacore.com/.

[5] http://www.cloudbyte.com/.

[6] http://www-03.ibm.com/software/products/en/virtual-storage-center.

[7] https://www.sdxcentral.com/resources/security/security-challenges-sdn-software-defined-networks/.

[8] https://cloudsecurityalliance.org/.

defined perimeter (SDP) project as new security architecture in order to keep secure systems against network attacks SDP (2013). SDP was designed to complement SDN in order to reduce the attacks on the network applications by disconnecting them until the users and devices are authenticated.

Another security company launched its own SDSec solution, called vArmour,[9] for SDN-based and cloud data center systems to fully exploit the benefits of virtualization environments. vArmour addresses the scalability, flexibility, and cost issues facing traditional security techniques in virtualization environments. It provides a dynamic and secure protection for various organizations assets that work with a new paradigms like cloud computing, mobile applications and virtualization systems. vArmour protects distributed data, which are located across several servers in an efficient manner to allow the enterprises to adapt with the new business changes in real-time. Other examples include (VMware 2010; VMware Inc 2013; NetCitadel Inc 2012).

5 Software defined for IoT network

The large number of objects in the IoT network make the traditional IP standards are unable to fit the large number of things connected to the Internet. In addition, these objects may have different characteristics and features, so there is a need to merge another routing protocol to accommodate this growing. Using the IPv6 may considered a good choose to deal with such number of objects, but it does not address the heterogeneity of the underlining objects. In a recent paper Martinez-Julia and Skarmeta (2014), the SDN was used to allow different objects from different networks to communicate with each other by using IPv6 and at the same time simplify the management and control operations of various objects types by adding an additional IoT controller over the SDN controller. In case if there is an object "A" need to communicate with another object "B" located inside the same network or even in another network, the IoT controller gets the information needed to defined different communication rules from the agents of the requested object which located inside this object, and find the receiver object, object B in this case, then uses the routing algorithm and different information from SDN to calculate the path to this object. After defining and establishing the forwarding rules, the IoT controller pushes these rules to the SDN controller which forward it to the forwarding devices. So even these objects, A and B, have non-compatible protocols, the forwarding devices through the path translate it in a proper way to be understandable by the receiver. This let different heterogeneous objects in the network to communicate in efficient manner.

As noted, the control operations are abstracted from the underling hardware objects, things, and set at a software layer, IoT controller. This architecture faces some issues that must be addressed when designing the SDIoT system. Some of these issues briefly discussed in Martinez-Julia and Skarmeta (2014). Selecting the best identification mapping approach that used by the IoT controller is one of these issues. The routing algorithm, the formula of the rules, the northbound interfaces, and the IoT controller model all of them also considered a design issues that must be taken into consideration to establish high level architecture.

The framework proposed by Huang et al. (2014) combined four key components; a set of nodes apply M2M protocol, a gateway to handle the devices which are not support M2M protocol, a set of another nodes and a controller to manage all of these types of devices. Ones the routing information changed the controller transfer the new version to the agent in the objects to update the routing information on each one. In this manner, the durability of the IoT network will be enhanced.

The example of the SDN/OpenFlow for WSN by Hu et al. (2014) which we mentioned it in Sect. 2 is also mimic the idea of the SDIoT system. Since in WSN there are many sensors which have different characteristics and features. Maintain and mange these sensors require a lot of work and take a lot of time specially if the network has an extreme number of sensors. It makes no sense to go to each one and make the updates required whenever the environment has changed Zhan and Kuroda (2014). For this reason the SDN/OpenFlow can solve this issue by abstract the control form these sensors and sit it on a control layer and keep the sensors only responsible forward the data without any control operations. Such abstraction simplifies the management and control operations for these sensors and at the same time increases the efficiency, scalability and elasticity of the WSN. The same thing can be applied in smart environments where the objects interact with each other to make the decision by abstracting the control services to IoT controller and link all of these objects to this controller.

Stefan Bernbo, the CEO and founder of Compuverde, talked about the need to restructure and redesign the data storage systems to be able to deal with the massive amount of data produced by objects, things, in IoT networks Bernbo (2014). Thus, he discussed some problems in traditional storage appliances which considered critical holes to accommodate this data, and then he showed how the intelligence in the SDStore is able to address all of these challenges. The strong dependency between the hardware and its software is one of these problems. The hardware

[9] https://www.varmour.com/.

and software provided as a single package, since the software is implemented and designed for the hardware. Such dependence becomes useful when the environment does not change frequently, but when the environment is constantly changing this solution effects on the efficiency of the system. The highly cost which derived from the redundant storage appliances to recovery from failures also considered one of these problems. Further, the overhead on a single entry point in the traditional appliance prevents the system from expansion to provide the required capacity for IoT data growth. Fortunately, the SDStore is able to address all of these bottlenecks by decoupling and moving all the control services to a separated software layer and provide a scalable, portable, undependable, inexpensive, agile and horizontal, distributed data streamline storage solution.

These challenges and other ones that exist in classical storage solutions have been studied by the authors in Cecchinel et al. (2014) and motivated them to propose a new software-based architecture to handle the Big Data which generated from the sensors and other objects in IoT network. This architecture based on the cloud computing to store this data instead of storing it in the physical appliances. Before they start to build their solution, they set up four design requirements that must be carried over by any storage solution architecture for IoT-based network. The new solution must be able to support different types and platforms of sensors, data and protocols, and heterogeneous hardware. Building a scalable solution either vertically to add an extra storage space, or horizontally to provide a good load balancing is also considered a mandatory requirement for any solution. In addition, a remotely reconfiguration for the underling devises should be provided by that solution. Finally, it should have fine-grained user applications to let the end users to access and query the gathering data in a smooth way.

The architecture of their solution is divided into three layers: Physical Infrastructure layer, Cloud Infrastructure layer, and Data as a Service (DaaS), application, layer. The first layer combines various sensors networks; each of one consists of a set of sensor groups where every group is connected to one board sensor. The sensors on each group collect the data from the environment and then transfer it to the sensor board which aggregates several types of the data from different sensors. After that, all of these sensors board transform its collected data to the bridges which linked to its associated sensors network. All of these boards connect to the bridges by a physical or wireless link. The bridge after receiving the stream of data it broadcast the data to the APIs through the Internet. The middle layer, Cloud Infrastructure layer, working as a mastermind for the network, where it has a three key main components; the Database to store the gathering data for further usage in the future by the end user, a sensor parameter database to store all the information about the sensors configuration, and a Middleware which surrounded by a set of APIs; APIs to receive the collected data as well as to broadcast the new sensors configurations to sensors networks bridges, APIs used by the administrator to set up the new policies and configuration measurements needed and APIs to connect with the Database in this layer, whereas, the application layer is implemented some applications for the end users to interact with the data.

The requirements which they figured out are applied in their design. The heterogeneity in the sensors is handled by the sensors network, since the bridge is responsible to define a unified structure for all platforms types in ordered to keep a consistence view all of these different platforms. In addition by using the cloud infrastructure their scalability requirements will be realized. Further, the mastermind middleware provides an ideal solution to remotely reconfiguration in a transparent way without the need to tell the user about the specific details of the underlying infrastructure, he just run some simple operations as Add, Delete, Route, and frequency operations to add new sensor or delete it, change the endpoint destination and change the frequency needed respectively. And sure, by providing a Data API the DaaS requirement will be satisfied.

Some recent papers (Qin et al. 2014; Zarko et al. 2014; Orphanoudakis et al. 2014; Nastic et al. 2014) mimic the idea of software defined systems for IoT network but within a close range. PatRICIA (Nastic et al. 2013) is a programmable model which provides a simple and efficient solution to develop and deploy the IoT application in the cloud by abstracting the implementation of knowledge from its representation. This solution gives the developer an easy way to implement different IoT applications on the cloud without the need to know details information about underling devices. But, it lack to a monitoring and management programmable solution to control these devices.

6 The proposed SDIoT model

6.1 The proposed model architecture

Finding a good architecture design that tackle all the challenges and issues which founded in IoT network and limit from the full exploitation of its benefits is considered a competitive advantage to any IoT provider. Thus, in this section and according to design principles for the SDStore, SDN, SDSec, or any software defined system we will propose a comprehensive Software Defined IoT (SDIoT) system architecture solution to accelerate and facilitate IoT control and management operations and at the same time cover and tackle the problems that exist in classical design.

Figure 1 depicts the general view of our proposed SDIoT system architecture prototype.

As shown in the Fig. 1 the proposed architecture has three main components:

1. The physical layer: In this layer all the assets and hardware devices in the system are reside. All of these physical devices interact with the data and forward different messages without interfering in the management and control operations. This layer classified into several clusters; Sensor Network (SN) cluster, DataBase pool Cluster and maybe other types of clusters like switches/routers cluster and security appliances cluster.

    (a) Sensor Network Cluster: Each sensor network consists of a set of sensors. The sensors are responsible to gather the information from the surrounded environments in order to use it in different applications. Where the agent in every sensor is responsible to communicate with and transfer the data to the associated board sensor. The board sensor combines different types of data from different types of sensors and pass it to the bridge which is located between the board sensor and the Southbound APIs (S-APIs) in the middleware layer. The bridge after combining the data from different board sensors it carries the data to the Middleware layer through the S-APIs. A unified platform is implemented inside the bridge to translate and map the different types to a single, known and suitable format.

    (b) Database pool cluster: This cluster is responsible to store the data with different types. A dedicated Database is created to each type of information. This database pool is possible to be a data warehouse or any database type. The configuration information Database (config.Info DB) keeps the mandatory information about each sensor and board sensor in the SN like the sensor ID, protocol type, end point destination, frequency and others. Whereas, all the collected data form sensors stores in the raw data Database. Other types of data from the system can be stored in other dedicated DB.

2. Control (Middleware) layer: This layer is considered the core of our prototype. Since, several software defined controllers are located and integrated inside this middleware layer; IoT controller, SDN controller, SDStore controller, and SDSec controller. All of them are entirely software-based controllers which abstract the control and management operations from the underling physical layer. The administrator can easily reconfigure the devices through a standard East APIs (E-APIs). In wide range system where the system is

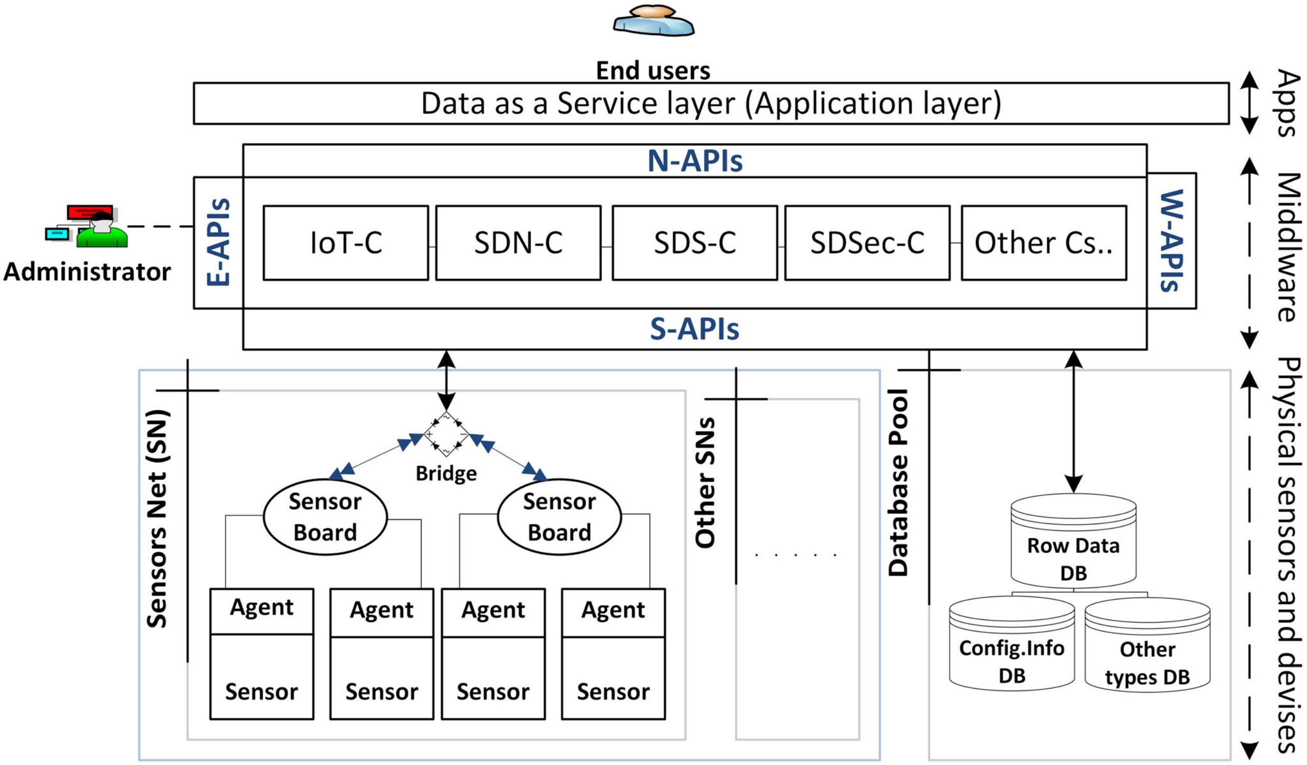

Fig. 1 The proposed SDIoT architecture design

too large or when it physically distributed then an additional controllers set nodes will be added to the system in the middleware layer to handle all the requests from the users concurrently, this maintains a high level of load balancing and consistence, and at the same time provides a fast responses time for many requests. The West APIs (W-APIs) is used to communicate with these other sets of controller nodes in case the system needs to be scale. More details about the workflow inside this middleware will be discussed later in this section. Furthermore, the integration of the sensors in any environment to capture the data to assist in decision-making process means building a smart environments which required an autonomous system (AS) to control it without the need to involve the end user or even the administrator.

3. Application (DaaS) layer: Lastly, the application layer is simply combined many fain-grained user applications which facilitate the accessing and acting with the stored data by the end users through the Northbound APIs (N-APIs).

6.2 The overall workflow for the proposed model

Figure 2 presents the overall workflow inside the Middleware layer. The starter point begins when the data received from different bridges in the data plane. A group of collectors take this data and apply some appropriated operations to organize the data into different packages according to their sender IP address. This process tries to reduce the further processing time when it links all the data for a specific network to the same package.

After that, the role of SDSec-C will be started to address the identity for every object. It asks the checker to look at the Authentication database and check if the object is authenticated or not. Then, the checker forwards the results to the SDSec-C. Other security techniques are applied to find if there are vulnerabilities on the data. If the SDSec-C discover that everything is going fine then it will assign a positive (P) flag value for this object. Else it will assign a negative (N) value. The Authentication process looks for the value of the flag. If it P then the message enter into the message queue and wait until one of message processors becomes free. The message processor is responsible to filtering and generating useful, meaningful information from the row data.

Using more than one collector, message queue, and message processor allows more than one request to be treated at the same time. This distribution is designed to accelerate the response time of system at overall. After the message processing operation has been finished, the IoT-C role will be started. In IoT-C, different operations applied on the derived information to defined different communication rules and find the receiver object. After that, it uses

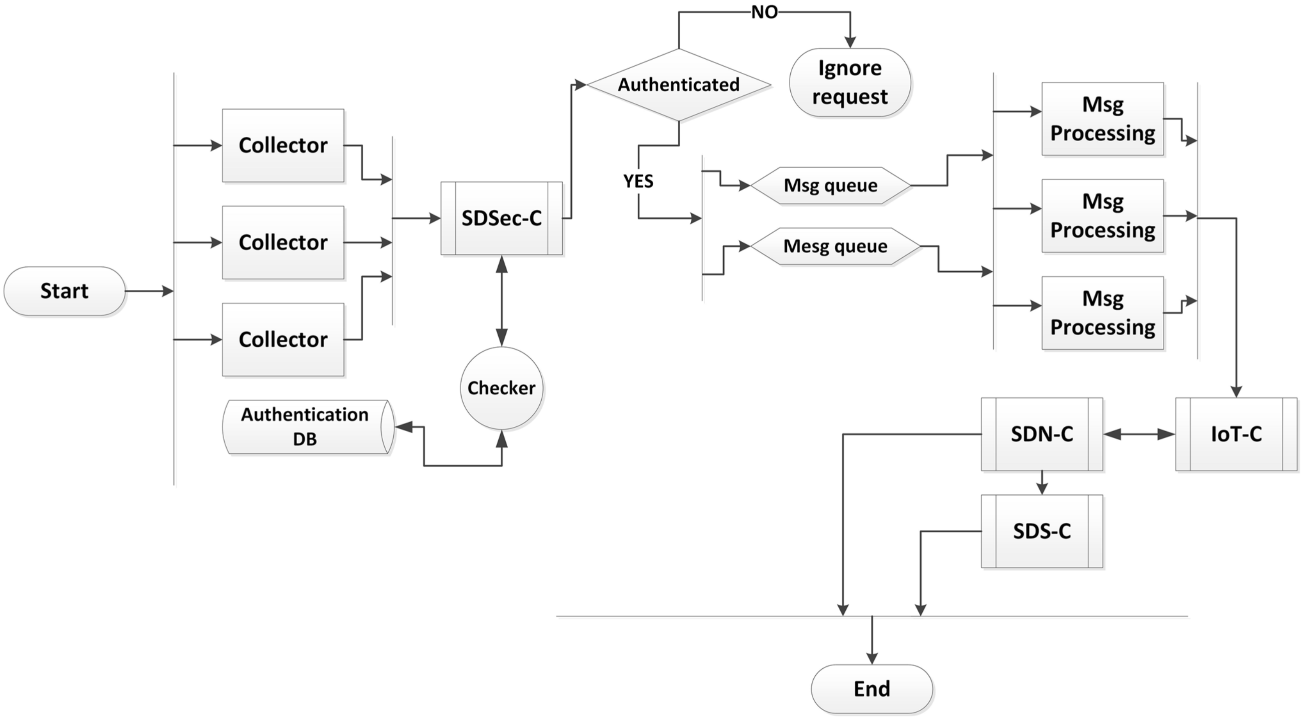

Fig. 2 The workflow inside the middleware layer

the routing algorithm and different information from SDN-C to calculate the path to destination object. When the IoT-C finished its work, it pushes the forwarding information to the SDN-C.

The SDN-C has three responsibilities; the first one is updating the information in the IoT-C in case if there are any changes have been occurred in the underling devices. In addition, it is responsible to broadcast the final routing and forwarding information to the underling devices which located inside the SN cluster. The last one is to inform the SDS-C to begin its work by sending an alert after the routing tables reached the underling devices. The SDS-C stores all the data in the suitable database in the Database pool cluster. At this point the middleware role will be finished. Different topologies and framework can be built to test different aspects related to the SDN, SDStore and SDSec by using our experimental frameworks; SDStorage (Darabseh et al. 2015b) and SDSecurity (Darabseh et al. 2015a).

7 Conclusion

In this paper, a software defined based framework for Internet of Things (SDIoT) is proposed. At the first, we highlighted how the software defined system handle the challenges of traditional system architecture as it provides a centralized, programmable, flexible, simple and scalable solution to control the systems. Then, different forms of SDSys (SDN, SDStore, and SDSec) presented and explained, which are considered the main known ones from the SDSys.

After discussing existing SDN, SDStore and SDSec solutions, we talked about our proposed SDIoT architectural model and showed how we exploit the ideas from SDN, SDStore, and SDSec to build it. Later, we presented its main elements and showed how these elements interact with each other to provide a comprehensive framework to control the IoT network.

The proposed model was built to provide a proof of concept, and we explained how the systems can be built to accommodate large data which produced from the widespread of the IoT. We plan to develop an experimental framework for SDIoT to test different forms and types of the IoT topologies.

Acknowledgments The authors would like to thank the Deanship of Research at the Jordan University of Science and Technology for funding this work, Grant Number 20150050. Also, they would like to thank IBM and IBM Cloud Academy for their support.

References

Almodawar A et al (2013) Security-aware placement and migration algorithm in iaas interclouds. In: The fourth international conference on information and communication systems (ICICS 2013)

Bernbo S (2014) The internet of things demands new data architecture. http://datacenterpost.com/2014/07/internet-things-demands-new-data-architecture/. Accessed Nov 2014

Cloud Security Alliance (2013) Software defined perimeter. White paper, Cloud Security Alliance

Cecchinel C et al (2014) An architecture to support the collection of big data in the internet of things. In: 2014 IEEE world congress on services (SERVICES), pp 442–449. doi:10.1109/SERVICES.2014.83

Crump G (2013) Storage switzerland: software defined storage needs a platform. In: Technical Report TSL03137USEN, IBM Inc

Darabseh A et al (2015a) Sdsecurity: a software defined security experimental framework. In: Third workshop on cloud computing systems, networks, and applications (CCSNA-2015)

Darabseh A et al (2015b) Sdstorage: a software defined storage experimental framework. In: IEEE international conference on cloud engineering (IC2E 2015)

Dixon C et al (2014) Software defined networking to support the software defined environment. IBM J Res Develop 58(2/3):3:1–3:14. doi:10.1147/JRD.2014.2300365

EMC Corporation (2015) Transform your storage for the software defined data center with emc vipr controller. White Paper H11749.4, EMC Corporation

Gubbi J et al (2013) Internet of things (iot): a vision, architectural elements, and future directions. Future Gener Comput Syst 29(7):1645–1660

Huang H et al (2014) An sdn_based management framework for iot devices. In: Irish signals systems conference 2014 and 2014 China-Ireland international conference on information and communications technologies (ISSC 2014/CIICT 2014). 25th IET, pp 175–179. doi:10.1049/cp.2014.0680

Hu F et al (2014) A survey on software defined networking (sdn) and openflow: from concept to implementation. In: Communications surveys tutorials, IEEE PP(99), pp 1–1. doi:10.1109/COMST.2014.2326417

IBM Corporation (2014) Choose a storage platform that can handle big data and analytics. Solution Brief TSS03158-USEN-01, IBM Corporation

Jain R, Paul S (2013) Network virtualization and software defined networking for cloud computing: a survey. IEEE Commun Mag 51(11):24–31. doi:10.1109/MCOM.2013.6658648

Jarraya Y et al (2014) A survey and a layered taxonomy of software-defined networking. In: Communications surveys tutorials, IEEE PP(99), pp 1–1. doi:10.1109/COMST.2014.2320094

Martinez-Julia P, Skarmeta AF (2014) Extending the internet of things to ipv6 with software defined networking. White paper, Euchina-fire. http://www.euchina-fire.eu. Accessed May 2015

Nastic S et al (2013) Patricia—a novel programming model for iot applications on cloud platforms. In: 2013 IEEE 6th international conference on service-oriented computing and applications (SOCA), pp 53–60. doi:10.1109/SOCA.2013.48

NetCitadel Inc (2012) Netcitadel's onecontrol platform the key to intelligent, adaptive network security. White paper, NetCitadel Inc

Nastic S et al (2014) Provisioning software-defined iot cloud systems. In: 2014 international conference on future internet of things and cloud (FiCloud), pp 288–295. doi:10.1109/FiCloud..52

Nunes B et al (2014) A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surv Tutor 16(3):1617–1634. doi:10.1109/SURV.2014.012214.00180

Orphanoudakis T et al (2014) Next generation optical network architecture featuring distributed aggregation, network processing and information routing. In: 2014 European conference on networks and communications (EuCNC), pp 1–5. doi:10.1109/EuCNC.2014.6882669

Palanivel K, Li B (2013) Anatomy of software defined storage challenges and new solutions to handle metadata. In: Report. University of Minnesota, Minneapolis

Qin Z et al (2014) A software defined networking architecture for the internet-of-things. In: 2014 IEEE network operations and management symposium (NOMS), pp 1–9. doi:10.1109/NOMS.2014.6838365

Systems and Technology Group (2014) Transform your business with cloud-ready storage "realize wide-ranging cloud benefits using ibm storwize family systems". Solution Brief TSS03147-USEN-01, IBM Corporation

TalebiFard P, Leung V (2014) Context-aware dissemination of information and services in heterogeneous network environments. J Amb Intell Hum Comput 5(6):775–787. doi:10.1007/s12652-013-0210-y

Vizardl M (2013) What software-defined security could mean for the channel. http://goo.gl/OABC3D. Accessed Oct 2014

VMware (2010) Vmware vshield virtualization-aware security for the cloud. White paper, VMware Inc

VMware Inc (2013) Vmware vcloud networking and security overview. White paper, VMware Inc

Wu F, Sun G (2013) Software-defined storage. In: Report. University of Minnesota, Minneapolis

Yaseen Q et al (2013) Pep-side caching: an insider threat port. In: 2013 IEEE 14th international conference on information reuse and integration (IRI). IEEE, pp 137–144

Zarko I et al (2014) Iot data management methods and optimisation algorithms for mobile publish/subscribe services in cloud environments. In: 2014 European conference on networks and communications (EuCNC), pp 1–5. doi:10.1109/EuCNC.2014.6882657

Zhan Y, Kuroda T (2014) Wearable sensor-based human activity recognition from environmental background sounds. J Amb Intell Hum Comput 5(1):77–89. doi:10.1007/s12652-012-0122-2