Combining Software-Defined Networking with Internet of Things: Survey on Security and Performance Aspects

Muneer Bani Yassein, Qusai Abuein, Sanaa Abu Alasal

Jordan University of Science and Technology
Irbid, Jordan
masadeh@just.edu.jo, qabuein@just.edu.jo, sanaasal11@gmail.com

***Abstract*— With increasing number of devices that are used in the paradigm of the Internet of Things, a need for managing these devices and its applications has popped up. In addition to the need for protecting it appropriately and effectively to ensure its full functional performance. Software-Defined Networking has been suggested so as to overcome the problems of Traditional Networking. The SDN became able to separate the logical and control operations in the distributed devices located in the network layer, by moving it to the central control layer that carries out administrative matters of this network. This survey paper contains a comparison between the Internet of Things that is enabled by Traditional Networking and Software-Defined Networking. Starting with defining different architectures, crossing to the main issues for both and ending with performance measuring to evaluate the SDN concluding to the possibility of applying it.**

Keywords—Internet of Things (IoT); Software-Defined Networking (SDN); Traditional Networking; Big Data; SDNIoT.

I. INTRODUCTION

Day after day the use of the Internet of Things (IoT) is expanding to include a large number of applications being used in our daily life. Also, numerous numbers of devices are used to deliver these applications, including sensors, mobile devices and actuators. These devices communicate with each other through an integrated network that might be wired or wireless. What makes the IoT more important topic is its existence in daily practices that deals with the circumstances and the surrounding events such as business administration, health care, automation, factories, weather forecasting and many more areas. These sensitive events impose some kind of dread, whether IoT is going to be really safe and robust enough to provide services fully and without any concern of private information leaks or errors occur during the automation processes. Users of this type of ongoing dynamic services needs to be in constant connection with portability to anywhere using a large storage capacity to store the vast amount of data that will be picked up by the sensors. A good example is introduced in [1]. Smart connectivity and computation capability are important aspects to be conceded, so that the Traditional Networking would be weak to be involved in such connection. For those reasons the need for a complete connectivity between IoT elements is urgent. Being sure about the existence of a secure connection without the presence of any performance-related or privacy issues is a must [2]. In IoT, networking is not only about connecting two sides, it is also about supporting high quality streaming for huge amount of confidential data. Hence, this kind of connection requires an infrastructure that can deal with the "agility gap". The Traditional Networking structure allows many security-related challenges to occur, so it is inflexible environment for IoT applications. The current base structure needs months to be altered, or even a long time to amend a new version of hardware. Consequently, the idea of combing the Software Defined Networking (SDN) with the IoT infrastructure is shined. SDN promises to enable more manageable and secured connections, by implementing the concept of programmable networks. This paper discusses the ability of handling IoT needs and issues by changing the Traditional Networking architecture.

A. Internet of Things IoT

The Internet of Things IoT is a novel model that is swiftly earned a reputation in the scenario of up-to-date wireless telecommunications. The basic idea of this concept is the widespread participation of a variety of things or objects around us, such as cards, sensors, actuators, Radio-Frequency Identifications (RFID), control systems and PDAs, which are being able to communicate through unified schemes to achieve a common goal [3]. These physical objects must behave together to spread information and make decisions, so IoT converted them from being traditional thingummy to intelligent tools that are capable to act with tact. The most interesting thing about IoT is that it is not just confined for the industrial automation and arithmetic operations; it is also engaged with the way we live. For instance, smart-homes would enable people to open the garage as soon as they arrive home [4]. So, there would be some demands to achieve this ambition summarized as follow [2]:

978-1-5090-6778-7/17/$31.00 ©2017 IEEE

- Sharing the understanding between user and application.
- Full architecture and distributed network infrastructure to process and transmit relevant information.
- The existence of intelligent analysis tools that is capable to deal with things and convert data and information to knowledge.

The IoT elements, applications and services are essential components of an IoT architecture.

1) IoT elements: in a high level of abstraction, there are three main elements in IoT that makes the vision closer to the real. Firstly, hardware that made up the model of the connection, such as sensors, actuators, cameras, etc. Secondly, Middleware; which is the storage and computation tools that used to analyze the data. And finally, Presentation; which is about the smartness of easily visualizing the knowledge by the interpretation tools [3] [4].

2) IoT applications: there are many application domains that export the functionalities of the IoT systems to the end users. These applications classified upon the availability, scale and user involvement as follows [3]:

a) Homes: where the user is the owner of the network, which is usually a WiFi network with high bandwidth for both sounds and videos. Health Care systems are the most obvious example, where grey-haired people health is monitored to collect information about their physiological parameters. This leads to hospital's cost reduction, by early mediations and treatments by doctors. Also, controlling home utilities such as conditioners, machines and lighting would reduce wastage of energy consumption.

b) Projects: which is the first field that IoT started its implementation, where the data selected and released by the owners selectively. It is about managing utilities of building's lights for example, or monitoring the environment for security issues, automation of factories even climate controlling.

c) Services: the information gained from this domain is used to improve the quality of services in order to optimize the assets vs. revenues to gain the best resources management. Water and electricity monitoring is a good example, by constantly collecting data about the consumption to alter the rebalancing of the loads in cities or areas.

d) Mobiles: Bluetooth Devices, GPS sensors, Light sensors and many other technologies in the smart phones could empower the IoT in the mobility area. In this domain the security and privacy is a big challenge to be considered. [2]

3) IoT Challenges: IoT in its first virtually version is a machine to machine (M2M) communicating network, with its own characteristics, elements and state using wireless networks (WLANs). By the time it expanded to include the interactions between Human and Machines too. This new virtual enhancement caused problems concerning with security and privacy of information through [1]:

a) Front-end equipment and sensors: those are receiving data through sensors, and then transmit it using M2M devices. This method requires secured connection with trusted connectivity. Since that nodes are distributed among the network, meddler can easily breakthrough the connection. So that, many possible threats might occur; such as prohibited access to data and DoS attacks [23-28].

b) Network: Represents a cornerstone in IoT, by providing inclusive and efficient connectivity between huge amounts of devices with Quality of Service (QoS) that is satisfactory enough. These numerous number of devices is a weakness exposed for DoS attacks [5].

c) Back-end of IoT systems: means the middleware, which needs a high quality securing. There are seven major criteria: privacy safeguard, access control, user hallmark, communication layer security, data solidity, data exclusiveness and availability at any time.

II. CURRENT SOLUTION: IOT WITH TRADITIONAL NETWORKING

A. IoT Architecture

Many IoT architectures have been suggested, all of them is focusing on the variety of application's manifestation. The most popular model divided to: sensing layer, network layer and service layer [6].

Starting from lower to upper, sensing layer consists of sensing devices or what mentioned before as RFID. Ordinarily these devices are smart enough to read and recognize the environment then read data from different sources, after that they store this data in small sized memory. Subsequently, the data is transmitted to the ports for wireless transmission. Those ports are generally use HTTP or MQTT protocols [6]. Data compressing, gathering and integration are used to deal with big data situations, to efficiently transmit data. Network layer contains the ports and routers that will transmit the data to different applications. The final layer of services concerned with supporting tremendous data centers, which offers analytical services such as Data Mining, processing and many

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors).*

applications.

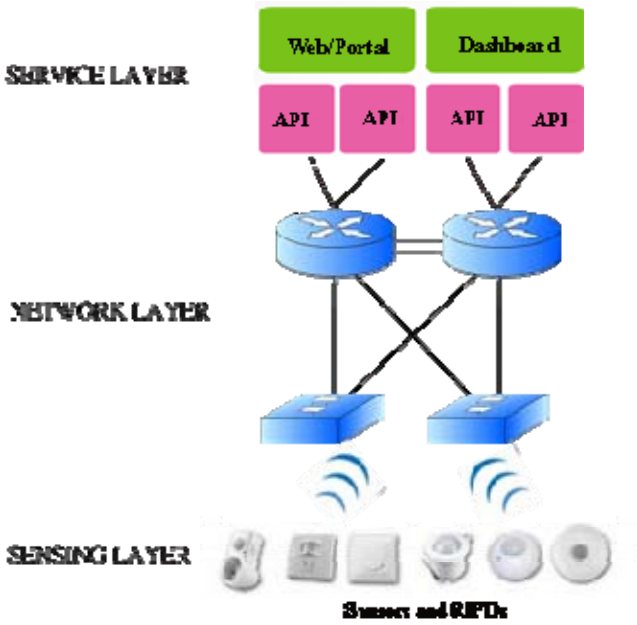

Fig. 1. IoT Architecture.

B. *Traditional Networking*

In traditional networks the data plane and control plane are combined into a single node Fig. 2.

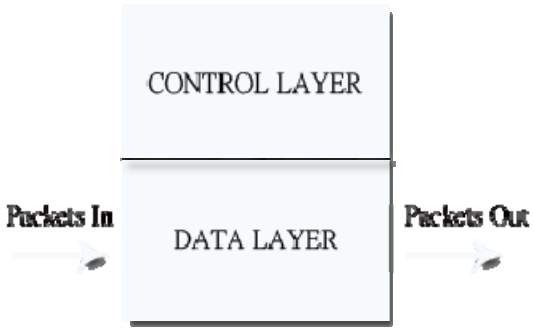

Fig. 2. Traditional Networking Architecture.

Each of these planes has individual tasks to supply the routing or switching missions. Control plane is responsible about configuring devices and programming pathways needed for data flows. For example if a Hello World Message received to the control, it will be processed to check the availability of its path. If the message path is available, then it will be pushed into the data plane and kept in hardware. The data plane uses the recently provided information by the control plane to decide the path inside the hardware.

This traditional hardware process is an effective method that can treat heavy processing limits and configuration exigency, so it can reduce the passivity and gain a fast processing speed. On the other hand, ordinary network kits (routers or switches) have obstacle: a) software and hardware are hold tightly or vertically integrated [7] that cause a reduction in the flexibility; b) network protocols that incorporated in the devices are too knotted; c) devices have manufacturer-proprietary, that makes it difficult to update or change the implemented functionalities [8]. Also, IP addressing in Traditional Networking is complex and robust to be managed. Each individual network device must be configured separately via dealer-specified commands. So the automated configuration and reaction technique does not exist in the present-day IP networking [7]. Another insuperable challenge of Traditional Networking is the Internet Ossification, caused by the high deployment of Internet by the human society. Internet has difficulties in the physical development of infrastructure, performance and protocols. Considering management points such as ports of each switch or router inside the network, we can illustrate that each one must be configured individually which is too lazy and time consuming. As a result, management and performance accuracy are very difficult problems that must be handled [7]. To sum up those issues, the idea of changing the underlying structure of the network has been raised. This change involves the separation between two physical layers, data layer and control layer. This new paradigm called Software Defined Networking (SDN).

III. Proposed Solution: IoT with SDN

A. *Software-Defined Networking (SDN)*

Software Defined Networking shatters the "Vertical Integration" by splitting up the control layer from the routers or switches. This could qualify the centralization of the controlling tasks, with the evolution of programmable interface. This new confrontation aims to terminate the limitations of Traditional Networking, by transforming the routers into normal forwarding devices and grabbing the brilliant piece outside it by implementing the controller to be a centralized and programmable unit administered by an Operating System [7]. So, SDN centering on four dimensions: Separation, centralization, open interfaces and programmability [9].

1) SDN Architecture Overview: SDN architecture consists of three main layers: data layer in the lower level, control layer in the middle, application layer on the top as shown in Fig. 3.

Data forwarding layer consists of unbounded number of SDN switches connected by wire or wireless media. This soft switch contains a forwarding table "Flow Table" that includes a huge amount of rules compelling the forwarding expedition [10]. Each rule exists inside the flow table consists of three fields: action, counter and pattern. The pattern is a header value that defines the packet. As a packet arrived to the port of the switch, it will search for a rule that matches the fields of the packet. If such a rule is found, counter field will increase and the action related to the rule will be invoked. On the other

978-1-5090-6778-7/17/$31.00 ©2017 IEEE

hand, if the rule is not found; the switch will request for a help from the controller or it may drop the packet [7]. This explicit communication between control layer and data layer happens over OpenFlow, which is a "well-defined application programming interface (API)" [7] [11]. As mentioned earlier, the brain of SDN architecture is an extraordinary node called controller, this brain controls and manages the network communications. It uses several routing protocols such as: BGP and OSPF, so all actions happen in the data layer are administrated by instructions generated from the controller. Simultaneously, the controller has high level of vision of the forwarding layer topology. Some recent implementation architectures supports a set of multi-distributed controllers; this could excess the scale of the network resources, since that each individual controller node is taking charge of for controlling separate chunk of switches. These distributed controllers can contact through east-west-bound APIs [11]. Crossing up to the upper level of the SDN architecture where the application layer situated, this layer is responsible for guiding the network operators what the new in the business requirements field. The communication between applications and control layer take place through a north-bound APIs, for example REST API. Controller supplies to application layer the vision of the physical resources exists inside the network. This means that this architecture grants network operators with open programmable interface that can determine data paths of the packets, without a need to configure distributed switches individually [7] [11].

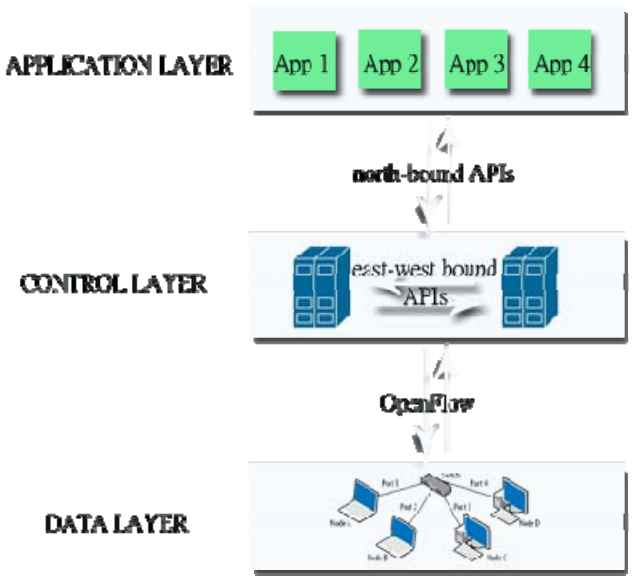

Fig. 3. SDN Architecture.

B. SDNIoT

The traditional IP addressing used in the traditional IoT creates challenges in handling the large number of things that connected together by the Internet. So, the use of IPv6 has become a need to cover the huge number of devices. But, "Heterogeneity" -which means that IoT objects are differing in their characteristics and properties-; is another problem that must be considered. The primary step of the combination between SDN and IoT is to compile and resolve the distinct types of workloads that IoT objects will propagate in the network. Here is the first step in designing the IoT Controller that will structure and modulate the controlling level. IoT Controller is a high level module that interacts with the SDN Controller to plan for the attitude of the network and react to the IoT functions. Fig. 4 shows the overview of the integration, with the minimum number of operational blocks, such as: objects, network element, and control or data layer.

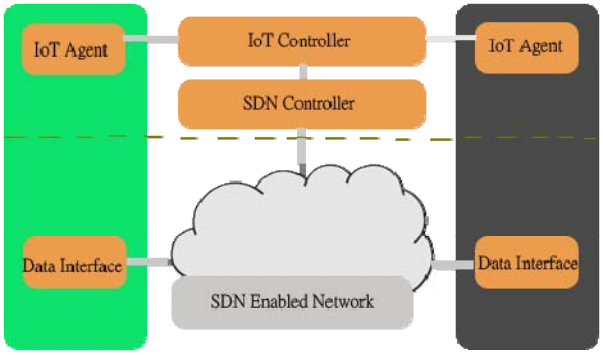

Fig. 4. SDNIoT Architecture [12].

The figure shows that if there are two objects that wish to communicate through SDN enabled network, they must comply with the IoT Controller via the use of the internal agents of IoT. The aim of this step is to give a vision to the controller to make the required decisions and then mirror them to the network. Despite the IoT Controller is shown as a rigid module, it is "Moldable" internally; so that new functionalities may be added to it without the need to affect the other element or changing the relations with SDN controller.

In any Traditional Networking architecture, ordinary communication protocols impose the sender object to ask the network to send the packet of data to another object. The Sender must know the address of the Receiver, and it should specify it in the request. Now SDN works to authorize "Heterogeneous" objects to interact with each other. So, SDN would be used to set up the path that will connect both objects. This path is called "forwarding path", used to build up the needed forwarding rules to all network elements exists in the path. Right now the IoT Controller turns on its functionality. It receives the packet from the sender, then search for the receiver path in the network graph. After that, it uses the routing algorithms to calculate the path to be built. Next, it builds the forwarding rules upon the protocols used in internal objects. Finally, it sends the rules to the SDN Controller which will set them at the forwarding elements. Packets are sent to the IoT Controller by the sender object to find the required details of the communication to be established, such as address of receiver. These steps are a preparation steps used to initialize the network connection between objects. When the IoT

Controller receives the packet with the destination address, it must find the path in the graph easily; this because all the objects are preregistered in the IoT Controller so it can rapidly provide their correspond path. Consequently, controller uses the network topology information –collected from both IoT Controller and SDN Controller- to run the routing algorithms that will build the connection path. This step ensures that the path resulted is suitable and constrained by the rules of the administrators. The following step depends on the IoT Controller, which will check the protocols used by the communicated objects. If the protocols are different, they will have different matching fields for example; and so they will have distinct reactions. Now, the protocols are differ and is not harmonious, specific rules would fire the forwarders to adjust or interpret it; therefore packets received and receiver understood. At the end, rules are sent to the SDN Controller, which uses the suitable protocol to attach them to the forwarders suggested by the controller. This time, the path is build up and a communication is begins. This process is introduced only for the first packet, in order that once the rule is generated to the forwarders they will run effectively as any router of switch.

As said before, every object must be preregistered into the IoT Controller. This mission is assigned to the IoT Agent that related to the object. Necessary information is provided to the IoT Controller such as: object's network, its protocol and address [12] [13].

IV. LITERATURE REVIEW

Jararweh et al. [13] overviewed the architecture such that the operation starts from the Middleware layer. Data is organized into several different packages depending on their senders IP addresses; this process tries to reduce the wasted time when linking the data of specific network to the same package. The next step is to check the authentication of the objects using SDSec [14]. If the authenticated of the object is successful, then a new field is added to the message after it is assigned a value (P). After that the message enters the queue for processing. On the other hand; if the authentication fails, then the field is assigned an (N) value and the request is denied. After that IoT Controller will fire the routing algorithm to determine the path of the destination object. When the path is determined then the IoT controller sends the information of the forwarding to the SDN Controller; which is responsible for changing the state in the IoT Controller when a modification occurs. IoT Controller is also responsible for broadcasting the routing/forwarding information to other elements inside the SDNIoT cluster. DevoFlow [15] for example, splits the coupling exists between the controller and its global vision of network, the idea tested through simulations; and the results showed that flow table entries and controller messages are decreased. Shu et al. [11] summarize the security threats into three groups according to the architecture level targeted. At data forwarding level, Man-in-middle attack may occur between switch and controller. Some researchers proposed applying some modifications on the data layer for some applications. Such proposal is not possible because it requires new hardware that does not support all kinds of security applications. Open Flow Extension Framework (OFX) [16] proposed the utilization of existing Open Flow switch, by allowing applications to load software modules for the security processes executions dynamically inside the data forwarding layer. Kandoo [17] suggested shifting the processing of frequent events into data forwarding layer, as an application of distributed control plane concept. Since this relocation needs modifications in the hardware of Open Flow switches, Kando proposed dividing the control layer into two sub-layers. The lower one contains a set of controllers that are neither connected to nor have knowledge about other network elements. These controllers handle the most frequent actions to support the upper level that contains the center-brain that drives the network. DIFANE [18] thought about the method of keeping the packets inside the data layer, and directing the selected ones by the use of a middle switch that saves the needed rules. Here the controller task became simpler since it partitions the rules among the switches. Hyper Flow [19] separates the decision making unite into individual controller by coinciding Open Flow controller vision of the whole network. This would decrease the control layer response time; which is the time needed to answer the data layer requests. The idea was an application of NOX [20]. It is worthy to mention that the Mininet [21] was the simulation tool used in previous solutions. This emulator can test wide networks with view resources through the use of a PC or virtual machine.

V. RESULTS AND COMPARISONS

From the security analysis point of view; the first advantage is to focus on the high level of monitoring of traffics inside the network topology, so it became easier to detect any abnormal attitude. Although the rapid response to vulnerabilities does exist in the network. According to the concept of programmability; once a threat is thrown and detected, network operators can implement an immediate solution without the need to waste time and efforts for updates in the operating system of the highly-integrated devices [11]. Going back to the challenges of separation concept and its disadvantages, the controller became the main part of the network; where information collection, configuration and routing operations are combined inside the controller. So, the network cannot function without this intelligent node. With the emergence of cloud computing that support the attackers with large ability of implementing several attacks; if a malicious happened successfully then all the services covered by the controller will be under this attack. Furthermore, the three divisions will be spread in different locations across the network; this would increase the communication lines; which will open the door for possible attacking points. Some examples of such points are:

- The SDN switch.
- Links between SDN switches.

978-1-5090-6778-7/17/$31.00 ©2017 IEEE

- SDN controller.
- Links between controller and switches.
- Links between controllers.
- The application software.

Open-programmable interface may be an emergence point to malicious applications coming from the application layer. Passing to the upper level; which is the application layer, some illegal accesses may occur by the unauthenticated penetrate malicious codes that bypassing into the open interface of the controller. Therefore, the need for security applications that can protect the SDN becomes very necessary. But, this is not applicable because security applications must be able to monitor, collect and process packets traffics in advanced manner. According to the SDN architecture, those tasks are separate inside the controlling unit.

From the Performance Analysis point of view; the main concern with the SDN here is to prove that the new architecture will be able to process the packets efficiently with high security levels. The performance efficiency here refers to the speed of processing the tasks of network elements with considering the throughput and latency [9]. If we think about the case of packet processing story, when a packet is arrived to a port it will be stored in the flow buffering area inside the Open Flow switch; waiting the acceptance from the flow table on packet information correctness. In case of acceptance it will pass, but in the case of uncertainty; flow table will request a help from the controller through the open flow client, then the controller would generate instructions of insertion or deletion of a rule inside the flow table by making some routing calculations. According to this long story we can predict that this process will be very slow. Some researches discussed the ability of enhancing the software or hardware capabilities to increase SDN performance by the use of [7]:

- Static Random-Access Memory (SRAM).
- Dynamic Random-Access Memory (DRAM).
- Reduced-Latency DRAM.
- Graphics Processing Unites (GPU).
- Field-Programmable Gate Array. (FPGA).

After analyzing the previous ideas, we arrived to a dark end by considering the huge cost of this solidly managed network. As a reflection on this high cost; researchers proposed the distribution of tasks between the controllers and Open Flow switches.

Although Kandoo [17] did not modify the switches, it has a problem with its local controller; such that it cannot propagate the OpenFlow events without the controller's subscription to that event. Its design depends on duplicating controllers to reduce the stress on the main controller. DIFANE [18] can be easily implemented using switches with huge amounts of PCs, flows and rules. Also, it could deal with wildcard rules efficiently and responds rapidly to network demands such as policy and topology changes. In addition to achieving higher throughput than NOX, it can rapidly recover from switch failures. HyperFlow [19] organizes surveillance in the data level; this yields to: de-serialize 987 events, serialize and write 233 events per second. These results were much better than waiting the main controller to react. DevoFlow [15] diminished the communications between controller and data plane; it has cut the flow table entries 10-53 times and the controller messages 10-24 times. After all, when analyzing all these different solutions; we wonder whether the application of SDN is approaching. Or are we really ready for this? The purpose of the survey is to find new solutions to provide efficient and complete communication that can handle the new services of IoT. Software-Defined Networking is a solution that is more probably to be established if a huge capital is available. It is remarkable to mention that Microsoft's datacenter network and Googles backbone network are using SDN architecture [11]. Therefore, it currently exists, but not all kinds of multisided organization could enable it.

VI. REMARK AND CONCLUSION

This paper gives an overview of the Internet of Things that is driven by the existing Traditional Networking architecture. Traditional Networking has its pros, cons, security issues, threats and some countermeasures. Then Software Defined Networking was proposed as an entrance to the concept of Programmable Networks and separation of architectures layers. The SDN architecture and its characteristics were introduced, going through the analysis of security and performance perspectives. Finally, the related works that helped in improving the security and performance issues were listed. We conclude that IoT is hard to be managed by Traditional Networking; because of the vertically integration property and vendor specific limitations that made the network more knotted and limited for particular configurations. SDN is proposed to be an opportunity to solve the previously mentioned problems. With SDN, logic applications implemented in a separated level; which makes it easier to be developed and deployed. SDN troth to provide networking with a centralized and open-programmable interface that is flexibly and efficiently managed. In order to achieve this promise, SDN must overcome many challenges. So, the paper presented a discussion of two main challenges, security and performance. There are many academic researches and industry fields trying to help in the shaping of SDN to be a successful and well managed network, thinking about the use of network management concepts and Programmable Networks architectures [22]. In another meaning, SDN does not fix any specific problem, but it extends more flexible ways to upgrade the IoT network management.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications", IEEE Communications Surveys & Tutorials 2015.

[2] J. S. Kumar, D. R. Patel, "A Survey on Internet of Things: Security and Privacy Issues", International Journal of Computer Applications (0975 – 8887) Volume 90 – No 11, March 2014.

[3] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions”, Future Generation Computer Systems 29 (2013) 1645–1660.

[4] L. Atzori , A. Iera, G. Morabito, “The Internet of Things: A survey”, Computer Networks 54 (2010) 2787–2805.

[5] P. Thubert, M. Palattella,T.s Engel, “6TiSCH Centralized Scheduling: when SDN Meet IoT”, 2015 IEEE Conference on Standards for Communications and Networking (CSCN).

[6] J. Li1 , E. Altman , C. Touati, “A General SDN‐Based IoT Framework with NVF Implementation”, ZTE COMMUNICATIONS September 2015 Vol.13 No.3.

[7] D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103, no. 1, pp. 14–76,2015.

[8] A. Ahmad, “Type of security threats and it’s prevention,” International Journal of Computer Technology and Applications, vol. 3, no. 2, 2012.

[9] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, “Are we ready for sdn? implementation challenges for software-defined networks,” IEEE Communications Magazine, vol. 51, no. 7, pp. 36–43, July 2013.

[10] H. Kim and N. Feamster, “Improving network management with software defined networking,” Communications Magazine, IEEE, vol. 51,no.2,pp.114–119,2013.

[11] Z. Shu, J. Wan, D. Li, J. Lin, A. V. Vasilakos, and M. Imran, “Security in software-defined networking: Threats and countermeasures,” Mobile Networks and Applications,pp.1–13.

[12] P. Martinez-Julia, A. Skarmeta, “Empowering the Internet of Things with Software Defined Networking”, WHITE PAPER.

[13] Y. Jararweh, M. Al-Ayyoub1, A. Darabseh, E. Benkhelifa, M. Vouk3, A. Rindos, “SDIoT: a software defined based internet of things framework”, February 2015 Springer.

[14] A. Darabseh , M. Al-Ayyoub , Y. Jararweh , E. Benkhelifa , M. Vouk , A. Rindos, “SDSecurity: A Software Defined Security Experimental Framework”, IEEE ICC 2015 - Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA).

[15] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, “Devoflow: scaling flow management for high-performance networks,” in ACM SIGCOMM Computer Communication Review, vol. 41, no. 4. ACM, 2011, pp. 254–265.

[16] J. Sonchack, A. J. Aviv, E. Keller, and J. M. Smith, “Enabling practical software-defined networking security applications with ofx,” 2016.

[17] S. Hassas Yeganeh and Y. Ganjali, “Kandoo: a framework for efficient and scalable offloading of control applications,” in Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012, pp. 19–24.

[18] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, “Scalable flow-based networking with difane,” ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, pp. 351–362, 2011.

[19] A. Tootoonchian and Y. Ganjali, “Hyperflow: A distributed control plane for openflow,” in Proceedings of the 2010 internet network management conference on Research on enterprise networking, 2010, pp. 3–3.

[20] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, “Nox: towards an operating system for networks,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 3, pp. 105–110, 2008.

[21] K. Kaur, J. Singh, and N. S. Ghumman, “Mininet as software defined networking testing platform,” in International Conference on Communication, Computing & Systems (ICCCS. 2014), vol. 20, 2014.

[22] B. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A survey of software-defined networking: Past, present, and future of programmable networks,” Communications Surveys & Tutorials, IEEE, vol. 16, no. 3, pp. 1617–1634, 2014.

[23] Aljawarneh, S. (2011). Cloud Security Engineering: Avoiding Security Threats the Right Way. International Journal of Cloud Applications and Computing (IJCAC), 1(2), 64-70. doi:10.4018/ijcac.2011040105

[24] Aljawarneh, Shadi. "A web engineering security methodology for e-learning systems." Network Security 2011.3 (2011): 12-15.

[25] Aljawarneh, Shadi A., Raja A. Moftah, and Abdelsalam M. Maatuk. "Investigations of automatic methods for detecting the polymorphic worms signatures." Future Generation Computer Systems 60 (2016): 67-77.

[26] Aljawarneh, Shadi A., Ali Alawneh, and Reem Jaradat. "Cloud security engineering: Early stages of SDLC." Future Generation Computer Systems (2016).

[27] Aljawarneh, S.A. and Yassein, M.O.B., 2016. A Conceptual Security Framework for Cloud Computing Issues. International Journal of Intelligent Information Technologies (IJIIT), 12(2), pp.12-24.

[28] Aljawarneh, Shadi, and Muneer Bani Yassein. "A resource-efficient encryption algorithm for multimedia big data." Multimedia Tools and Applications (2017): 1-22.

978-1-5090-6778-7/17/$31.00 ©2017 IEEE