

SDNs in Distributed Environments: Past, Current and Future Trends

Ahmad Salman Mansoor¹, Maruf Pasha²

¹ MS Research Fellow/Network Engineer, Department of Information Technology, Bahauddin Zakariya University Multan

² Assistant Professor, Department of Information Technology, Bahauddin Zakariya University, Multan.

ABSTRACT

Presently, there are various objects that influence the Internet architecture and engrave it in stone, which is called calcification of the Internet. The deployment of Internet Protocol version 6 IPv6 and the difficulties of IP multicast services are large obstacles. The inner and outer domains for the high-speed availability of routing provide end-to-end user connectivity and allow the dynamic management of QoS in upcoming applications of network resources. For example, virtualization of the network, cloud computing, and data center are required by the future-generation architecture of the Internet, which is called the Software-Defined Network (SDN). The separation of the control plane from the forwarding device plane is difficult given the distributed control plane in the present network. According to the prediction of researchers, more than 50 billion objects will connect to the Internet in 2020. This concept is called IoT (Internet of Things). In this scenario, SDN plays a vital role in decreasing the physical network compared to the virtual network, both performance- and efficiency-wise.

In this paper, we compare different framework architectures of SDN that are being used in distributed cloud-based environments on the basis of multiple parameters and we determine which framework architecture is best for the exponential growth in the number of Internet devices from the perspective of IoE in cloud- and grid-based environments.

Keywords: Software-Defined Network, Quality of Service, IoE, IoT, IPV6

Author's Contribution

¹ Manuscript writing and Interpretation,
Data collection and Analysis

² Conception, synthesis, planning of
research, discussion

Address of Correspondence

Maruf Pasha

Email: maruf.pasha@gmail.com

Article info.

Received: July 11, 2017

Accepted: June 26, 2017

Cite this article: Mansoor AS, Pasha M. SDNs in Distributed Environments: Past, Current and Future Trends. *J. inf. commun. technol. robot. appl.* 2017; 8(1): 8-14.

Funding Source: Nil

Conflict of Interest: Nil

INTRODUCTION

In the current era, networks provide sophisticated links among IoT objects. The number of links among objects grows according to the current situation. Furthermore, IoT objects are connected to the digital world and are growing in number over time due to the increasing use of smart environments. The extension of IoT into IoE is a future perspective, as mentioned in [1]. These sensors pass data over the Internet towards storage devices because they are continuously generating data in real time. Therefore, the management of this large amount of data is an important issue. In

academia and industry, big data is a very hot topic because managing, controlling and securing these sensor data in a heterogeneous environment is very complex and critical, as traditional networks do not fully support heterogeneity. However, from the data management point of view, NoSql databases come close to solving this issue.

Secure communication among IoT devices is of great concern. Data are transferred among these objects from one location to another, such as towards the storage area. With the immense development of Internet

technology, a large number of these objects are incorporated, with end users employing both wired and wireless schemes. A specific security method must be considered for the IoT, as it integrates each entity into the system efficiently. In this environment, each device has a particular identification number and the capability of moving data over the network without human-to-human or even human-to-machine communication. Additionally, on-demand services and the fast progress that is required by users are still problems on legacy networks. These networks are slow due to the nature of their compatible devices. Therefore, diversity in these networks and devices is necessary for the benefits of IoT networks to be realized. A new, game-changing technology in this domain is the Software-Defined Network (SDN), which has captivated many users from the industrial and education sectors. SDN provides the perfect platform for services that are demanded by users and fast development in IoT networks. Moreover, this new innovation also facilitates future IoT deployment, which is known as the Internet of Everything (IoE). This technology fully supports the distributed environment of IoT processing with different planes.

SDN is a developing technology that can meet the demands of IoT heterogeneity and malleability. The control functionality is decoupled by the SDN into a forwarding plane and a network service plane, which sits above the control plane. Similarly, the optimization of central management and other configurations of the network in an automated way provides interoperability between the IoT network and corresponding devices. Furthermore, the basic purpose of SDN is as a communication network, in which data move across the multiple stations in a distributed environment efficiently and effectively, as discussed in [2]. This network paradigm also provides security and privacy in IoT because the current IoT devices are very vulnerable. In this context, various types of SDN frameworks have been designed for processing, storage and security for management and privacy control of IoT devices in a distributed computing environment. One of the most important protocols is named Open Flow, as discussed in [3], which operates between the SDN and network devices. It reads the present state and provides forwarding instructions to establish a secure connection between the SDN controller

and network devices. Moreover, this protocol facilitates dynamic security policy deployment and traffic filtering among the communications of IoT objects in a distributed computing environment.

In this study, we have compared and observed different types of SDN frameworks and the internal techniques that are involved in a distributed computing environment. In the future, IoT devices will produce large amounts of data, which will create many issues in management, storage, and processing. Many researchers and scientists have invented different types of techniques and frameworks for processing the large quantities of information that are being produced by IoT entities. By evaluating these techniques according to future needs, we discuss how IoT objects will process data in distributed machines under the umbrella of SDN.

BACKGROUND

Computing devices and technologies are being developed very quickly due to the decline in cost and size of software and hardware. Various organizations and vendors are exploring new domains in which computing methods are very flexible and communication among devices is transparent. The communication and network domains of IoT and SDN are being merged to exploit their benefits for users and create smart environments. IoT overcomes the limitations of legacy networks such as VPN, while SDN better satisfies the new service demands compared to traditional networks. Cloud and grid computing are rapidly creating a virtualized smart environment for IoT, new services and user demands are of interest to the researchers in organizations. Physical connectivity among the data traveler devices creates many issues, such as delay and broadcast storm. In contrast, the software-defined architecture provides a virtual platform on which these issues are almost resolved by using the infrastructure layer (data plane).

This layer facilitates data processing by distributed IoT objects in a cloud or grid environment. As the IoT grows, network computation, analytics and storage will be carried out on different clouds, such as SaaS, PaaS and IaaS. In the near future, the combination of IoT and these clouds will be called IoE. SDN provides the baseline platform for the birth of the IoE. Many frameworks have

been designed in this context, which work in a distributed manner under the SDN architecture.

2.1. Distributed Smart Firewall Framework for Grid

The IoT objects are very vulnerable and static firewalls are used to prevent attacks. In this context, a dynamic firewall approach uses the SDN-based Controller functionality, which is called the Distributed Smart Firewall. This framework is based on the concept of grid-based security in a distributed computing environment as a middleware for decentralized security policy for network services via the firewall. A confidential zone is used for all connected network elements, as described in [3].

2.2. SofAir Framework [4]

This framework was designed for the classification of distributed traffic for network management, especially for routing with Open Flow and CPRI API in 5G networks, as mentioned in [5]. The purpose of this framework is cloudification of network functions and virtualization of the network. It is a very flexible platform for fully centralized structures, but security issues have still not been addressed.

2.3. SDWISE Framework [6]

This framework is being used for distributed localized sensor and energy management. Dump sensor nodes are used with a flow table, such as Open Flow, which has preinstalled flow rules with a dump data plane and centralized controller, as described in [7]. The basic purpose of this framework is to reduce the information exchange and mobility reconfiguration and localization in a distributed environment. However, architectural details and reliability are limitations.

2.4. DISFIRE Framework [8]

SDN-based security plays a vital role in IoT. Objects are working and transferring the data from one location to another by using authentication and authorization in a grid network environment. In this framework, the dynamic firewall verifies the authorization, which is implemented by multiple SDN controllers in a hierarchical network cluster. However, the evaluation is lacking and the OpFlex protocol is not functional, as illustrated in [7].

2.5. Cell SDN Framework [9]

Cellular networks are currently being widely used. Almost all humans are connected via the cellular network. The Cell SDN framework is being used for mobility and policy control management with NOS and support for

virtualization, in which the control plane performs local control as a centralized agent. It is used for MPLS traffic engineering. Mobility and grain control management are performed with the help of virtualization and a local agent, but there is no confirmation of proof concept, and its evaluation scheme and knob traffic for MPLS and VLAN have only been described in [7].

2.6. SOF Framework [10]

This framework is used for flow management in the data plane and packet forwarding. It decouples the distributed data plane and centralizes the controller with the Open Flow Protocol for sensor data. The main benefit is its handling of the compatibility and address classification. However, this is only theoretical and not experimental, as mentioned in [7].

2.7. Black SDN Framework [11]

Confidentiality, Integrity, Authentication and Privacy (CIAP), which are known as security building blocks, are very important in generic IoT object communication. This framework performs the encryption on the link layer and uses an SDN controller. However, its main shortcoming in providing complete security is poor scalability, as mentioned in [7].

2.8. CloudNaaS Controller Framework [12]

This framework is a placement optimizer that chooses the dynamic provisioning in a virtual network. This framework has the basic ability to reallocate the resources in a cloud network. It is also used for the evaluation of Open Flow prototyping and has intrusion detection caching as a cloud security feature, as mentioned in [12].

2.9. Architecture of the Software-Defined Network (SDN), Components and Protocol

The network elements require very skillful personnel for configuration and installation since deploying these kinds of objects requires availability and reliability for smooth packet transformation from one point to another. In a traditional architecture, the interactions among network nodes, such as routers, switches, hubs and firewalls, are very complex and enclosing the elements of reproduction is required. There are many limitations in a traditional architecture: The physically distributed devices are generally proprietary, the individual network elements cannot be scheduled due to service layer compliance, and the vertical assimilation in legacy network architecture

creates complications. The exponential growth in the number of connected objects increases the rate of transmission, and online services (e-mail, e-banking, etc.) are also increasing. The concept of SDN is not new; rather, it is the revolutionary result of contributions and development in network research. SDN brings technology in which all the network devices are in administrator premises and have limited numbers of configuration parameters through the low-level (CLI) interface.

From the Operating System emerged technological trends (multi support CPU, and multi-GPU, etc.) and new network requirements (QoS, VLAN, IPv6, etc.) via the protocols. In network fields, the implementation and

design cycles of state-of-the-art solutions could take many years and integrate new devices.

2.9.1 Architecture

SDN has a layered architecture, which contains three basic layers: the application layer, which is also called the management layer; the control layer, which is called the control plane; and the infrastructure layer, which is called the data plane. These three layers communicate with one another through APIs, which are called the Northbound Interface and Southbound Interface APIs, as mentioned in [13]. The architecture is illustrated in Figure 1.

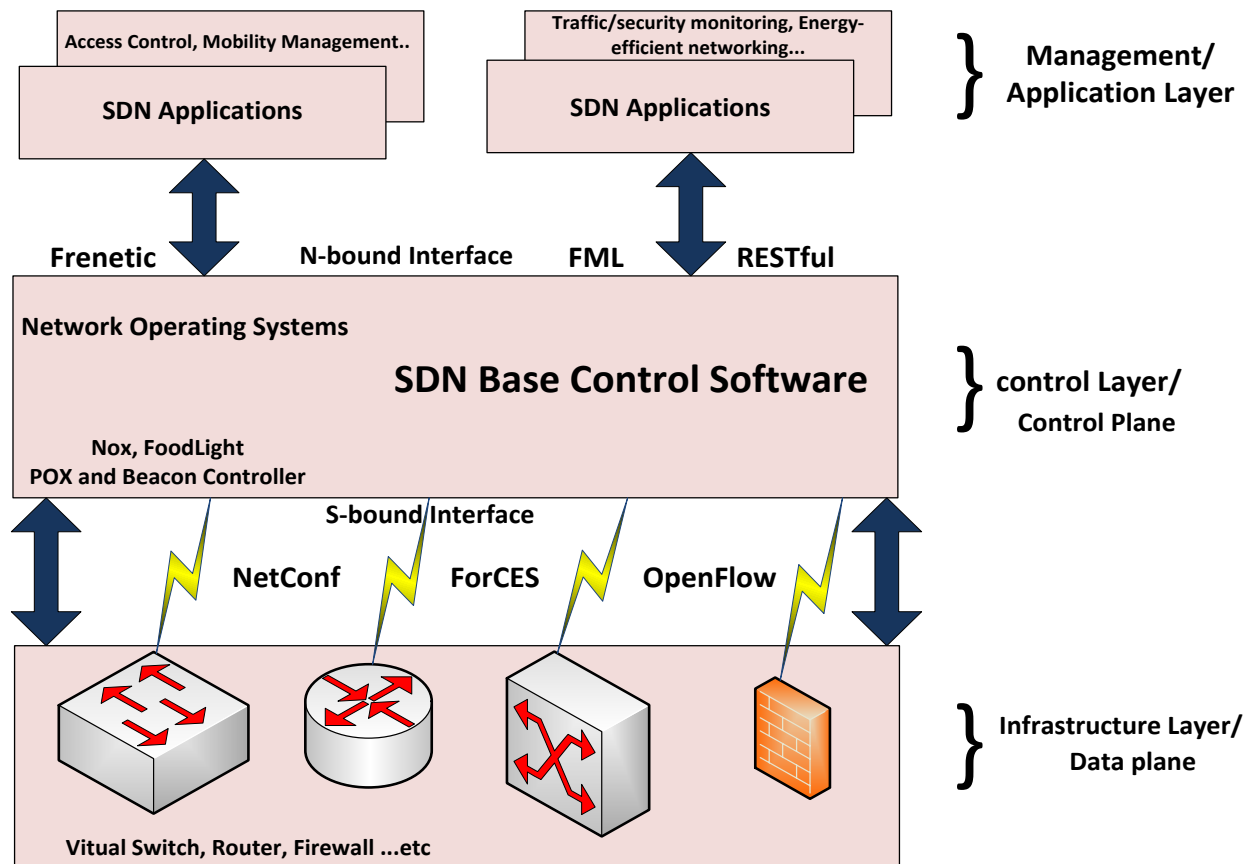


Figure 1. Architecture of Software Define Network

2.9.2 SDN components

This layered engineering structure contains three basic sections, which are represented in the form of rectangles: the management or application service layer, the control plane/layer, and the data plane, which consists of forwarding data traveler devices. These three rectangular sections communicate with one another by

using Southbound (SI) and Northbound (NI) interfaces as APIs, as described in [13].

▪ Application Layer

This layer is also called the management layer. It contains an application that is provided by the northbound interface to the gadget network and operation functions.

The policies are defined by the application layer, which transcribes to the southbound interface that promotes devices.

▪ **Northbound Interface**

NI provides the communication between the management and control planes. The Network Operating System provides the developer applications via northbound APIs. Forwarding of low-level instructions is implemented through the NI API, which is not well-equipped. RESTful is used as an API.

▪ **Control Layer**

This layer is also called the control plane, which decouples objects from scattered forwarding entities that are being stored on the central server machine. Through the northbound interface, the associated programs are forwarded to objects. CP defines the set of rules for forwarding objects. Therefore, CP acts as the “brain” of the network, in which all logic in applications and other controllers available, such as NOX[14] and Open Daylight[13], comes from CP.

▪ **Southbound Interface**

This interface provides the protocol, which is called Open Flow, as mentioned in [15], for communication between the forwarding devices and control plane via the southbound set of instructions. The southbound protocol assists the controller in programming and specifies rules that are used for communication between the control plane (CP) and the data plane (DP). The elements of forwarding and control are For CES [16].

▪ **Forwarding Devices**

Forwarding devices are core objects of the network, which are both software and hardware that complete the constitutional network tasks, according to the literature. These devices act according to the instruction set that is provided by CP on incoming packets. The instructions on the southbound (SI) interface, such as Open Flow[15] or For CES[16], are installed by the SDN controller and implemented by the SI protocols.

RESEARCH METHODOLOGY

The SDN architecture is used in different environments. In a distributed environment, especially a cloud- and grid-based environment, many types of frameworks have been designed. Our aim in this study is to construct a table of frameworks that are used in

distributed environments and the associated values of twelve parameters: Year, Management point of view, Target Network, Protocol, Scalability, Purpose, Security parameters, Implementation Domain, OS Controller, Description, Pros and Cons. Based on these parameters, we can determine which framework is best for a distributed service environment. With the comparative analysis of these frameworks and the given parameters, we can design a new framework that can fully meet the requirements of these parameters and enhance the performance of cloud and grid networks.

3.1 SDN-Based Distributed frameworks

The separation of the data and control planes provides independence of market vendors. The IoT is a layered structure that contains numerous levels of technologies, which can be an advantage in terms of benefits from the SDN. Various types of SDN frameworks have been designed for different components of SDN, especially for managing, controlling and securing the network objects from the IoT perspective. Table 1 lists the different distributed architectural frameworks, along with their internal structures, functions, purposes, pros and cons.

In this table, the first framework is Soft Air, which is described in [4]. It was designed in 2015 for 5G network technology. Its basic aim is to manage distributed traffic and perform classification on the 5G network by using the Open Flow and CPRI protocols. This framework refines the virtualization in the 5G network architecture with the help of a centralized software-defined controller and immense scalability. It was designed for collaborative processing and management of data mobility. The centralized flexibility of its platform architecture is its main advantage. However, security issues have still not been described in this framework.

The second framework is SDNWISE, which was described in [6]. It was designed in 2015 for the distribution of localized objects and power management in unknown networks. Open Flow[15] is used for the management of WSNs. It supports duty cycle and data management with three reference points in the wireless sensor network domain with a centralized Operating System controller. The scalability of this framework is at the intermediate level if the security parameters are unknown. This framework is used for WSN management

and data gathering. It translates between sensor object and WISE-Visor. It uses the state-full way, reduced data exchange and mobility localization, which are its main advantages. However, its internal architecture is not defined and it lacks reliability and security.

The third framework is DISFIRE, which is described in [8]. It was designed in 2016 for grid networks with the Open Flow protocol. Its scalability is intermediate-level. The basic purpose is the implementation of grid security by using the two security parameters for authentication and authorization. In a smart grid network, multiple SDN controllers are used in a cluster network environment. It implements the dynamic firewall for authorization. However, it has not been adequately assessed and the OpFlex protocol has not been tested in practice.

The fourth framework is CELLSDN, which is described in [9]. It is designed for mobility and policy management in a cellular network architecture using the NOS APIs. Its basic purpose is virtualization in a cellular network domain with an unknown OS controller. However, its scalability is poor. This framework uses Multi-Protocol Label Switching (MPLS) traffic labeling or Virtual Local Area Network (VLAN) for tagging. The positive aspects of the framework are smooth mobility management and achievement of local control. However, no proof of concept has been achieved.

The fifth framework is SOF, which is described in [10]. It was designed in 2012 and it is used for data flow management in sensor network architectures with the OpenFlow sensor protocol, as mentioned in [7]. It uses a centralized controller for WSN domain implementation with unknown security parameters. The scalability of this framework is very poor. It was designed only for the data plane and forwarding. The peer suitability and analysis are its main advantages. However, the concept of this framework is purely theoretical and has not been demonstrated in practice.

The sixth framework in this table is Black SDN, which is described in [11]. It was designed in 2016 for universal IoT objects. It provides a security implementation for small objects. It also uses the three security parameters of Confidentiality, Authentication, and Integrity (CIA), as well as other parameters, to secure universal IoT domain devices. However, its scalability is unknown. It has a centralized operating system controller. The main

shortcoming of this framework is its poor scalability for black networks.

The seventh framework is Distributed Smart Firewall for Grid, which is described in [3]. This framework was designed in 2016 for forwarding packets under a set of rules that are predesigned with the controller and it is used for cluster-based networks in virtual environments. The Open Flow protocol is used with unknown scalability. The main purposes are to control the IP header at the application level, minimize the problems of OpFlex in software-defined networks and provide the security for the grid. It is specially designed to ensure the security of cluster and grid network domains. It has a purely SDN-based controller. It enforces the security policies and management of the network objects from the IoT perspective. It performs the monitoring activities very efficiently and provides a security mechanism against internal and external threats. However, it cannot manage the cluster.

The eighth framework is called the CloudNaaS Controller Framework, which is used as an optimizer that selects the fine virtual network placing and provisions the dynamic model to reallocate the resources in a cloud-based network environment. It evaluates the Open Flow-based prototype of CloudNaaS and for a variety of network functions in a cloud-based network. This framework provides Intrusion Detection caching for ensuring the security of the cloud. The NaaS controller is used as an OS controller and within the cloud infrastructure of a very fast speed of programmable network element the transparency. It reduces the cost and complexity due to solution integration. However, the devices are very limited in terms of state.

CONCLUSIONS

The massive growth in the number of Internet objects has created a global electronic village, where objects are communicating with one another and transferring large amounts of data in a real-time environment under the umbrella of the SDN architecture. To enhance the dynamic nature of network functions, the programmability in different SDN frameworks plays an important role in effective and efficient communication among the objects. The benefits of the upcoming generation of SDN networks are that they make the communication among the IoT

objects transparent and simple, and they make updating and maintaining the application simpler and easier.

In the context of the SDN architecture, various types of frameworks have been designed for different purposes, such as to efficiently and smoothly manage the network programmability of services, where the different OS controllers work behind the software-based network infrastructure. Cloud- and grid-based SDN environments, where the control and data planes are separated, eliminates the legacy network structure and creates new horizons for virtualization and physical data transport. Data residency devices are converted into software by high-level programming techniques, which work in the virtualized cloud-based platform, where hardware resources are highly available. Various types of distributed SDN-based frameworks have been developed and are used in different types of activities in distributed cloud infrastructures for smoothing and managing associated plane activities.

In this paper, we have studied various types of frameworks that are used in distributed computing environments. Our main achievement is the comparison of these frameworks on the basis of multiple parameters and the determination of which framework is best according to the given parameters. In future work, we will design a framework on the basis of these parameters that meets our requirements for distributed cloud and grid networking and the predicted future requirements of IoE.

REFERENCES

- Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, 54, 1-31.
- Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., ... & Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7), 36-43.
- Gonzalez, C., Charfadine, S. M., Flauzac, O., & Nolot, F. (2016, July). SDN-based security framework for the IoT in distributed grid. In *Computer and Energy Science (SpliTech), International Multidisciplinary Conference on* (pp. 1-5). IEEE.
- El-Mougy, A., Ibnkahla, M., & Hegazy, L. (2015, October). Software-defined wireless network architectures for the Internet-of-Things. In *Local Computer Networks Conference Workshops (LCN Workshops)*, 2015 IEEE 40th (pp. 804-811). IEEE.
- Tayyaba, S. K., Shah, M. A., Khan, N. S. A., Asim, Y., Naeem, W., & Kamran, M. (2016). Software-defined networks (SDNs) and Internet of Things (IoTs): A qualitative prediction for 2020. *network*, 7(11).
- Galluccio, L., Milardo, S., Morabito, G., & Palazzo, S. (2015, April). SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless SEnsor networks. In *Computer Communications (INFOCOM), 2015 IEEE Conference on* (pp. 513-521). IEEE.
- Tayyaba, S. K., Shah, M. A., Khan, O. A., & Ahmed, A. W. (2017, July). Software Defined Network (SDN) Based Internet of Things (IoT): A Road Ahead. In *Proceedings of the International Conference on Future Networks and Distributed Systems* (p. 10). ACM.
- Dely, P., Kessler, A., & Bayer, N. (2011, July). Openflow for wireless mesh networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on* (pp. 1-6). IEEE.
- Li, L. E., Mao, Z. M., & Rexford, J. (2012, October). Toward software-defined cellular networks. In *Software Defined Networking (EWSN), 2012 European Workshop on* (pp. 7-12). IEEE.
- Luo, T., Tan, H. P., & Quek, T. Q. (2012). Sensor OpenFlow: Enabling software-defined wireless sensor networks. *IEEE Communications letters*, 16(11), 1896-1899.
- Wu, D., Arkhipov, D. I., Asmare, E., Qin, Z., & McCann, J. A. (2015, April). UbiFlow: Mobility management in urban-scale software defined IoT. In *Computer Communications (INFOCOM), 2015 IEEE Conference on* (pp. 208-216). IEEE.
- Benson, T., Akella, A., Shaikh, A., & Sahu, S. (2011, October). CloudNaaS: a cloud networking platform for enterprise applications. In *Proceedings of the 2nd ACM Symposium on Cloud Computing* (p. 8). ACM.
- Lara, A., Kolasani, A., & Ramamurthy, B. (2014). Network innovation using openflow: A survey. *IEEE communications surveys & tutorials*, 16(1), 493-512.
- Lara, A., & Ramamurthy, B. (2014, December). OpenSec: A framework for implementing security policies using OpenFlow. In *Global Communications Conference (GLOBECOM), 2014 IEEE* (pp. 781-786). IEEE.
- Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3), 1617-1634.
- Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4), 2233-2243.
- Perera, C., Liu, C. H., Jayawardena, S., & Chen, M. (2014). A survey on internet of things from industrial market perspective. *IEEE Access*, 2, 1660-1679.