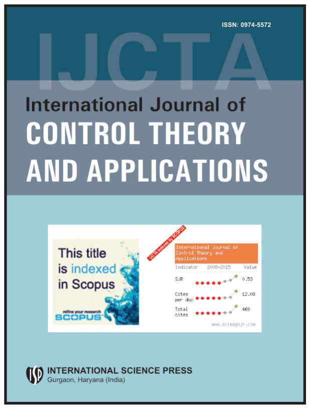

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 16 • 2017

Software Defined Networking: Issues and Applications

G. Sulthana Begam[a] and M. Sangeetha[b]

[a]*Research Scholar, Anna University/Assistant Professor, Information Technology, St. Peter's College of Engineering and Technology*
[b]*Associate Professor, CSE, Coimbatore Institute of Technology*

Abstract: Software Defined Networking (SDN) is a new paradigm in networking technology which separates the data plane as simple forwarding devices and the control plane as logically centralized controller program to monitor and control the status and behavior of the network. SDN has emerged as an efficient network technology capable of supporting the dynamic nature of future network functions and applications, lowering operating costs through simplified hardware and network management, effectively maximizing resource utilization. We outline the ONF defined SDN architecture, discuss innovative SDN applications, and explore possible research directions in the SDN paradigm.

Keywords: Software Defined Networking, data plane, control plane, OpenFlow, SDN Issues, SDN Applications.

1. INTRODUCTION

In traditional networked systems the control plane and data plane or forwarding plane are collectively exist together in a network element. The control plane responsibilities include exchanging routing information to neighbors and constructing routing table. The data plane is a switching element which process and forwards the data packet based on the forwarding table. Internet traffic level is approximately 150 Ebytes in 2012 and it is expected to be 5,021 Ebytes in 2017 [1]. As data traffic level increases costs of carrying the data generated also increases. SDN architecture decouples control plane and data plane simplifies the complexity and estimated to reduce the capital and operational expenditure for carrying data which makes control plane programmable. In traditional approach to modifying the forwarding policy is not easy as the devices are proprietary and it needs to change the configuration setting [2].

SDN is described by Open Networking Foundation (ONF) as the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications [3]. The dynamic factor driving the programmable networking research is the user demand for faster development and testing of new network services. The networking research community can test their new network ideas like routing protocol, security mechanisms, addressing schemes, and Network management techniques in real network setting and use network data traffic for research flow. This facilitates more elastic deployment of new network services than proprietary control systems typically closed vertically integrated systems whose functions are pre-defined through a long standardization process.

SDNs facilitates innovations in network architecture and protocol designs which introduces,

1. Separation of the control plane and data plane or forwarding Plane.
2. Centralized implementation control plane functions.
3. Programmable controller.
4. Standardization of the north and south bound interfaces.

SDN and OpenFlow benefits and influences cloud computing applications. Overlay Virtual Networks (OVN) provides simpler solution to manage virtualized networks and automated provisioning of VMs, and live migration of VMs in multitenant datacenter network [4].

IETF (Internet Engineering Task Force) and IRTF (Internet Research Task Force) are major regulatory bodies producing standards for protocols, APIs aiming centralized network management and control aims generalization of network interconnecting devices. The ONF follows the Standardization the OpenFlow protocol as they focus on standardizing interfaces between Control and Data plane. The Software-Defined Networking Research Group (SDNRG) is part of the Internet Research Task Force focuses some of the major areas like providing controller scalability solutions, programming languages used in an SDN context [5].

The paper is organized as follows: Section II provides an overview of an ONF defined SDN architecture. It also describes the different types of northbound and southbound application programming interfaces (API). Section III describes existing emulation and simulation tools for testing and evaluating SDN solutions. Section IV discusses issues in SDN and mechanisms to resolve the issues. Finally, Section V, we discuss some of the applications in the important research directions of data centers and mobile networks.

2. SDN ARCHITECTURE

SDN architecture is described in Figure 1 and it has Infrastructure, Control and Application layers [6]. The APIs used to communicate between the layers are grouped based on their functions as Southbound, Northbound and East/West bound APIs.

2.1. Infrastructure Layer

The bottom tier (data plane) involves the physical devices like Ethernet switches, routers and other middle boxes programmable through open interfaces.

2.2. Control Layer

The central tier (control plane) consists of the controllers, software programs responsible for making and manipulating the switch's flow table. controllers makes forwarding decisions, the way to direct network traffic by analyzing traffic statistics and changes them dynamically.

2.3. Application Layer

The top tier consists of functional applications such as intrusion detection system, routing, load balancing, energy-efficient networking, managing network services in a cloud, etc.

2.4. Southbound APIs

It is used to communicate between controllers and infrastructure. It enables a controller to define the behavior of switches/routers, OpenFlow being a prime example of a Southbound Interface protocol.

2.4.1. Netconf

The Network Configuration Protocol (NETCONF) is a network management protocol, which provides mechanisms to install, manipulate, and delete the configuration of network devices. It defines a standardized software APIs in an open and layered fashion. Its functions are implemented on top of a Remote Procedure Call (RPC) layer. The configuration data and protocol messages are encoded using an Extensible Markup Language (XML) and exchanged on top of a secure transport protocol [7].

2.4.2. FoRCES

Forwarding and Control Element Separation (FoRCES) protocol is proposed by IETF working group. It is an open interface used for communication and coordination of control and forwarding plane elements in the distributed router. FoRCES protocol used in the distributed router environments to install forwarding-table entries in the data plane, enabling the complete removal of control functionality from the routers to provide support for different router applications like IP routing, Simple Network Management Protocol (SNMP) and Multiprotocol Label Switching (MPLS) switching [8].

2.4.3. I2RS

The Interface to the Routing System (I2RS) protocol extends the functionalities of Netconf and FoRCES and provides a way for applications to access and control dynamic status information of networks and data flows. Security is one of the major area led the I2RS development [1].

2.4.4. Openflow

OpenFlow is one of the most common southbound SDN interface provides an open and standardized way for the controller to communicate with network devices in SDN architecture. Many vendors, including HP, NEC, NetGear, and IBM, produce OpenFlow-capable network switches which are available in the market. It constructs Flow Table which has entries with three fields they are

- Header describing the flow.
- The Action associated with the data packets are
  - Forward data packet to the given port or controller
  - Drop the packet
- Number of packets and bytes for each flow
- Time since last action.

There are a variety of OpenFlow controllers capable of making changes in the flow table entries for example, NOX, Floodlight, Maestro, Cyan Blue Planet, HP VAN Controller, NEC Programmable Flow, etc. NOX is a framework that allows programmers to develop programs using C++ or Python and a set of APIs to communicate and coordinate with OpenFlow-capable switches. Floodlight is based on Java platform. Maestro focuses on controller scalability and better QOS performance metrics. A Secured Channel connects the open flow switch to a remote controller delivers commands and packets to controller and the open flow switch making use of OpenFlow Protocol. It is designed to support policy-based flow management within a network. These responsibilities includes

- Centralized control
- Traffic monitoring, analysis
- Updating forwarding actions dynamically

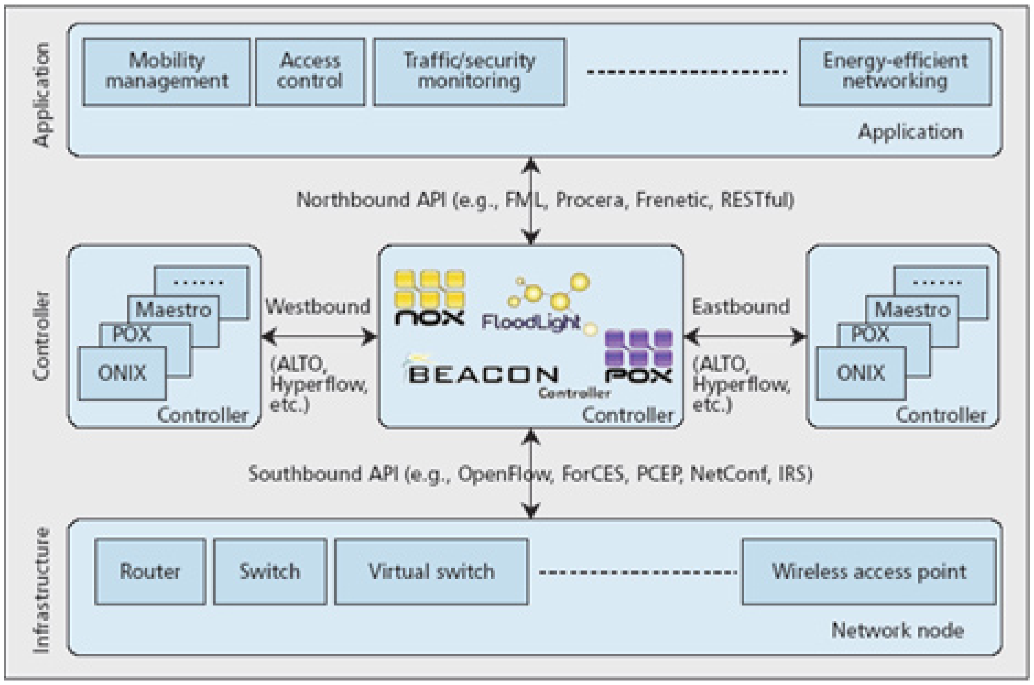

Figure 1: SDN Functional Architecture

OpenFlow-based applications manages to simplifies

- Configuration of a network
- Process of network management,
- Virtualizes networks and data centers
- Deploying mobile systems [2].

2.5. Northbound APIs

The northbound Application Programming Interface (API) lies between SDN controllers and network applications. It provides an interface between network operator and the controller to customize the network applications. Because SDN is relatively new, the industry is still identifying and evaluating the capabilities of northbound APIs.

2.5.1. Frenetic

Frenetic is a network programming language that supports the application development on top of network operating systems for SDN. The Frenetic language offers programmers set of interface for coding controller programs used mainly in the network management functions are:

- Traffic flow Monitoring.
- Constructing packet forwarding policies.
- Modifying policies consistently.

SDN controller programs responds to topology changes, link failures, traffic at specific switches. Frenetic's supports query language extends the possibilities for the developers to specify characteristics to monitor, leaving behind the implementation details of how to collect the appropriate statistics from the network elements. An example query is given below:

Select (bytes) *

Where (inport = 80) *

GroupBy ([10.10.11.1]) *

Every (60) (1)

- Select clause indicates total number of bytes expected
- Where clause limits the query to the traffic destined to input port 1 of the switch.
- Group By clause restricts the program to consolidate traffic from the source IP address 10.10.11.1
- Every clause restricts that the traffic should be counted for every 60 s.

The result set of the query can be used as input to any other controller program [9].

2.5.2. Procera

Procera applies the principles of Functional Reactive Programming (FRP), which provides a declarative, expressive, and extensible framework for describing reactive and temporal behaviors. It implements reactive policies in a simple and declarative language that responds to various types of events such as user authentications, time of day, bandwidth use, or server load. The polices [10] are translated into a set of forwarding rules on the underlying network infrastructure. Many networks uses access control policies which limits access to some devices or services on the network.

2.5.3. Pyretic

Pyretic is Python-based platform that enables system programmers to create sophisticated SDN applications. The primary advantage of Pyretic's is that policies are represented as abstract functions approach which helps to support modular programming. A Pyretic policy is defined for the entire network at once, policies takes packets from the specified switches as input and there are two types of output

A set of single or multiple packets and forwards that to a new location or multicasts the packets to different locations respectively.

An Empty set corresponds to dropping the packet.

A Pyretic program can combine multiple policies together using one of several policy composition operators; the following code uses sequential composition to compose three independent policies.

match(switch=2)>>match(dstip='3.3.3.1')>>fwd(3) (2)

The first two policies [11] are filters to specify that packets located at switch 2 and forwarded IP address 3.3.3.1 and the third policy is a forwarding policy which specifies packets should be forwarded through out port 1.

2.6. East-Westbound APIs

East-West bound APIs Communicates between groups or federations of controllers to synchronize state for high availability. It allows different controllers from neighboring or in the same network to communicate and coordinate with each other.

3. SIMULATION TOOLS

The functions implemented by the controller programs can be deployed, tested over an experimental test bed like Emulab and PlanetLab. The simulation tools ns-3, EstiNet and Mininet able to evaluate its performance also.

3.1. NS-3

Ns-3 is a network simulator that implements the OpenFlow switches with some limitations are

- The real OpenFlow controller program cannot be compiled and linked together to form a single executable program without modification.
- STP and the multiprotocol label switching (MPLS) functions are not supported.
- The simulated results deviates from correct results since a simulated openFlow switches and its controller are not connected through TCP.

3.2. Estinet

EstiNet is a network simulator and emulator based on Openflow [12] combine the advantages of both the simulation and emulation approaches and manages to overcome their disadvantages. It generates time-related OpenFlow performance metrics correctly and the results can be repeated many times. It uses a kernel re-entering approach to test the functionality and performances of OpenFlow controllers.

3.3. Mininet

Mininet is a network emulator which is capable of setting up a a network consists of systems, connecting devices, controller programs, and communication links. It uses the virtualization technique to create emulated hosts and programmable software OpenFlow switches connected together through links in a particular network topology and can also be controlled by a OpenFlow controller program. The application code developed and tested on Mininet, for an OpenFlow controllers can be deployed and tested on a real networked system with minimal changes and the performance can be evaluated.

Mininet supports research development, innovations, learning, prototyping, testing by setting up a entire experimental network on a laptop or other PC. Mininet combines together some of the best features of emulators, testbeds, and simulators [13]. Compared to hardware testbeds, Mininet is

- Inexpensive
- Easily Reconfigurable

Compared to simulators, Mininet

- Executes Open vSwitch, OpenFlow controller and SDN application codes
- Application codes can be shifted to Real Networks easily and without modification.
- Offers Interactive Performance.

It also has some limitations.

- The packet forwarding rate is not predicable and it is based on processor speed and capacity of main memory.
- The results mostly vary for every run.
- The main issue in practice is it can (currently) run only Linux-compatible OpenFlow switches or application.
- It cannot be used evaluate application performance of time related network.

4. ISSUES IN SDN

The ongoing research in SDN mechanisms and protocols is broadly categorized as

- SDN models
- Controller Scalability
- Security
- Interoperability

4.1. SDN Models

Decoupling the data and the control plane leads to interesting properties like number of controllers needed, the placement of controller in different network topology and whether controller resides distributed or Centralized. A central controller put forwards scalability, availability concerns and resilience to failures. An algorithm and a mechanism were implemented [14] to dynamically change number of controllers used and to change control of switches from one controller to another as required.

4.2. Controller Scalability

As the network expands centralized controller can be replaced by distributed controllers and the placement of controller is based on the performance metrics like latency, resilience to failure and balancing load. The controller is responsible for installing forwarding state on switches on a per-flow basis provides flexibility it introduces a flow setup delay also. DIFANE [15] focuses on various mechanisms to resolve controller scalability issue and its performance measure.

4.3. Security

The Characteristic features of SDN exposed to threats and attacks. SDN security issues and the corresponding research solutions [16] are classified into:

- Unauthorized Access of controller and applications resolved by secured connection, authentication and access control mechanism.
- Denial of Service attack (DoS) reduced with dynamic Flow table and Distributed controller.
- Data Leakage like intercepting flow rules and forwarding policy and
- Data Modification is modifying flow rules with no prominent solution for both.
- Configuration issues like lack of support for Transport Layer Security

The SDN architecture supports and also enhances some network security mechanisms like:

- Managing denial of service(DoS) and DDos attack
- Intelligent Intrusion Detection and Prevention Systems
- Securing Middle boxes,
- Authentication, Authorization and Accounting,
- Secure, Scalable Multi-Tenancy

Qiao Yan et. al., in this paper [17] consolidated all the methods to overcome DDos attack using SDN in cloud computing environment and to prevent SDN from DDos attack because of its characteristics.

The ONF has also identified that the southbound communications between controllers and data-forwarding devices as vulnerable. OpenFlow has fundamental authentication technology that prevents a hacker from spoofing flow commands from a controller to a switch.

4.4. Interoperability

The IETF path computation element (PCE) [18] helps in gradual or partial migration from traditional networking technology to SDN.

OpenFlow switches are the main infrastructural element of SDN architecture follows OpenFlow specification. When the Controllers and the OpenFlow switches developed my different vendor's works together there are two major issues

- Interoperability among the switches
- Compatibility of controllers

The research work in this direction needs lot of attention.

5. SDN APPLICATION AREAS

By putting an API on top of the control tier, applications can be written to control the underlying network. Some of the areas of SDN applications are discussed below.

5.1. Load Balancing

Load balancing distributes a server workload or network traffic to improve response times and resource utilization. At present network load balancing is performed by dedicated hardware appliances and software. The hardware is expensive, lacks flexibility and is easy to become a single point failure. In SDN, a central controller makes the decision for packet traversal and not the switches. Controller dynamically detects the topology by listening to the switches and calculates available path with less load. Controller then directs the switches with forwarding entries needed for the paths thus efficiently balancing the load with every flow. Although SDN is a bit slower than traditional networking, there are added advantages of its own such as.

- Consideration of end-to-end path.
- Congestion control.
- Dynamic adaptation to topological changes.

A new solution for load balancing in the OpenFlow environment is proposed in [19]. The Controller collects the server running status and calculates the aggregated load of the severs according to dynamic load balancing scheduling algorithm. The OpenFlow switch will forward the client's request to the server whose aggregated load is lowest, thus minimizes the response time of the server.

5.2. Traffic Engineering

Traffic Engineering (TE) is a method for optimizing the, performance of routing, minimizing congestion control, improving network resource utilization, achieving QoS by monitoring, managing and scheduling the data flows in a network. Traffic oriented performance includes packet transfer delay, packet loss and throughput. ONF proposed different type of protocol for TE solution based on Data-Controller Plane Interface (D-CPI), the Application-Controller Plane Interface (A-CPI) and the Management Interface (MI) existing in SDN Architecture.

The important research areas applicable to TE are

- Design of integral TE solutions operated by different protocols
- Impact of non-TE applications sharing infrastructure.
- Maintaining consistency of TED (Traffic Engg. Databases)
- Problems in re programming the switches

A virtual intrusion detection system deployed on controller can identify incompetent users and recognizes malware traffic. SDN-based IPS deployment is presented in [20] which supports a unified scheduling of security applications in the whole network and load balancing among IPSs. Relevant

5.3. Wireless Networks

The Next generation wireless networks powered by IoT, Smart devices and Social networks demands

- Different Radio Access Technologies(RATs) to optimize channel estimation and allocation,
- To minimize latency in Handover and
- Self-healing in case of failure in backhaul connection.

Kun Wang et. al., proposed SDN based architecture [21] for NWNs with Virtual RATs and a programmable Controller to overcome the above issues. Modern WLAN controllers can now provide better security and access control to wireless LANs. An SDN application makes LANs and WANs simpler to manage, flexible, more secure and more powerful.

5.4. Cloud Computing

Network Virtualization (NV) is the key to the current and future success of cloud computing. SDN is expected to make the networks easily partitionable and virtualizable. These features are required for cloud computing where the network infrastructure is shared by a number of competing entities. Paim de Jesus et. al., proposed the ProViNet platform [22], which merges the SDN and NV concepts to provide a fast and safe deployment of applications which are built simply by composing control plane services in the existing network infrastructure. Wei-Chu Lin et. al., presented an unified solution[23] combining two strategies, flow migration and VM migration, to maximize throughput and to avoid congestion in SDN-Based Cloud Datacenter. Mechtri. M. et. al., presented a SDN controller, called Cloud Networking Gateway (CNG) Manager [24] that enhances networking of distributed cloud resources and provides authorized customers with the ability to control and configure networks.

5.5. Data Centers (DC)

DCs are becoming increasingly complex due to the new emerging virtualization technologies. The multitenant Data Centers are moves towards a dynamic infrastructure which consists of optically interconnected top-of-rack (ToR) switches interconnects group of physical servers each running many VMs. Data center network (DCN) virtualization benefits service providers and organizations with optimized resource utilization, Energy Efficiency, Low operational cost, rapid service delivery, flexibility of using VMs with Guest OS and new network services, and automated provisioning and migration of VMs within DCs. Shuo Fang et. al., proposed a Lossless dynamic load balancing multipathing Solution for Data Center Network Using SDN Approach [25] resulting in better bandwidth utilization, improved network throughput and with integrated congestion control.

5.6. Software Defined Mobile Networks

SDN can simplify the design and management of cellular data networks. SDN for Cellular Networks offers a logically centralized control plane which will lead to simpler and effective radio resource management (e.g., inter-cell interference management). It enables common control protocol across diverse wireless technologies 4G LTE, 3G UMTS, WiFi that simplifies mobility support and also it allows distributing traffic monitoring at switches [26]. Cellular network functions that could benefit from SDN are

- Inter-cell interference coordination
- Mobility support
- Adaptive and opportunistic spectrum use
- Distributed QoS and Access Control Policies enforcement
- Directing Traffic Through middle boxes
- Monitoring for Network Control and Billing

5.7. Information Centric Networking

The separation between information processing and forwarding in ICN is aligned with the decoupling of the data plane and control plane in SDN. SDN can be employed as a key enabling technology for ICNs OpenFlow expands to support customized header matching.

6. CONCLUSION

This paper provides an overview of the emerging field of Software Defined Networking (SDN) and described the SDN architecture as well as the OpenFlow standard in detail. The Languages for writing controller programs for Software-Defined Networks and simulation tools used for implementing SDN were discussed. Also most prominent issues in SDN, some of the potential research solutions to resolve these problems and different application areas which have been making use of SDN paradigm were examined.

REFERENCES

[1] Susan Hares, and Russ White. Software-Defined Networks and the Interface to the Routing System (I2RS). Published by the IEEE Computer Society, IEEE Internet Comp. Mag., 84-88, Jul/Aug 2013.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson,J. Rexford, S. Shenker, and J.Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., Vol. 38, No. 2, pp. 69–74, 2008.

[3] ONF, “Software-Defined Networking: The New Norm for Networks,”white paper,https://www.opennetworking. org

[4] “Software Defined Networking A new paradigm for virtual, dynamic, flexible networking,” IBM Systems and Technology White Paper, Oct 2012.

[5] David Meyer. The Software Defined Networking research group. IEEE Inter. Comp. Mag., 84–87, Nov./Dec 2013.

[6] Sakir Sezer, Barbara Fraser, and David Lake. Are We Ready for SDN? Implementation Challenges for Software-Defined Networks. IEEE Commun. Mag., 36–43,July 2013.

[7] Rensink Enns, Ed., “NETCONF Configuration Protocol”, RFC 4741, December 2006.

[8] ForCES IETF Working group, [Online]. Available http://www.ietf.org/html.charters/forces charter.html.

[9] N. Foster, M. J. Freedman, A. Guha, R. Harrison, N. P. Katta, C. Monsanto, J. Reich, M. Reitblatt, J. Rexford, C. Schlesinger, A. Story, and D. Walker, “Languages for software-defined networks,”IEEE Communications, Vol. 51, pp. 128–134, Feb 2013.

[10] Hyojoon Kim and N. Feamster. Improving networ management with software defined networking. IEEE Commun. Mag., 51(2):114–119, February.

[11] [Online]. Available: http://www.frenetic-lang.org/pyretic, 2013.

[12] Shie-Yuan Wang, Chih-Liang Chou and Chun-Ming Yang. EstiNet OpenFlow Network Simulator and Emulator. IEEE Commun. Mag., 110–117, September 2013.

[13] [Online]. Available: http://mininet.org/

[14] Advait Dixit, Fang Hao, Sarit Mukherjee, T.V. Lakshman, and Ramana Kompella. Towards an elastic distributed networking, HotSDN ’13, pages 7–12, New York, NY, USA, 2013.

[15] M. Yu, J. Rexford, M.J. Freedman, and J. Wang. Scalable flow-based networking with difane. In Proc. ACM SIGCOMM 2010 conf. on SIGCOMM, pages 351–362. ACM, 2010.

[16] Sandra Scott-Hayward, *Member, IEEE*, Sriram Natarajan, and Sakir Sezer, *Member, IEEE* “A Survey of Security in Software Defined Networks ” Ieee Communication Surveys & Tutorials, Vol. 18, No. 1, First Quarter 2016 Pages 623-654

[17] Qiao Yan, F. Richard Yu, *Senior Member, IEEE*, Qingxiang Gong, and Jianqiang Li “Software-Defined Networking (SDN) and Distributed

[18] Denial of Service (DDoS) Attacks in Cloud Computing Environments:

[19] A Survey, Some Research Issues, and Challenges” IEEE Comm. Surveys

[20] & Tutorials, Vol. 18, No. 1, First Quarter 2016 Pages 602–622

[21] [Online]. Available: http://tools.ietf.org/html/rfc4655

[22] Zhihao Shang, Lanzhou Univ., Lanzhou,Wenbo Chen ; Qiang Ma ; Bin Wu “Design and implementation of server cluster dynamic load balancing based on OpenFlow” in International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA), 2013 .

[23] Lei Zhang Guochu Shou ; Yihong Hu ; Zhigang Guo, “Deployment of Intrusion Prevention System based on Software Defined Networking “, Communication Technology (ICCT), 2013 15th IEEE International Conference, Nov 2013

[24] Kun Wang, Yihui Wang, Deze Zeng, and Song Guo, “An SDN-Bas ed Architecture for Next-Generation Wireless Networks” IEEE Wireless Communications, February 2017 pp 25-31

[25] Paim de Jesus, W. ; Fed. Univ. of Rio Grande do Sul, Porto Alegre, Brazil ; Araujo Wickboldt, J. ; Zambenedetti Granville, L., “ ProViNet -- An Open Platform for Programmable Virtual Network Management”, IEEE 37th Annual Conference on Computer Software and applications (COMPSAC), pp. 329–338, July 2013

[26] Wei-Chu Lin ; Dept. of Electr. & Comput. Eng., Nat. Chiao Tung Univ., Hsinchu, Taiwan; Chien-Hui Liao ; Kuan-Tsen Kuo ; Wen, C.H.-P. "Flow-and-VM Migration for Optimizing Throughput and Energy in SDN-Based Cloud Datacenter ", Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference, Vol. 1, pp. 206-211, Dec. 2013.

[27] Mechtri, M. ; Inst. Mines-Telecom, Telecom SudParis, Evry, France Houidi, I. ; Louati, W. ; Zeghlache, D., "SDN for Inter Cloud Networking ", Future Networks and Services(SDN4FNS),2013 IEEE SDN, pp. 1-7, Nov. 2013.

[28] Shuo Fang, YangYu, Chuan Heng Foh, and Khin Mi Mi Aung, "A Loss-Free Multipathing Solution for Data Center Network Using Software-Defined Networking Approach", IEEE TRANSACTIONS ON MAGNETICS, Vol. 49, No. 6, pp. 2723- 2730, JUNE 2013.

[29] Li Erran Li, Z. Morley Mao and Jennifer Rexford "Toward Software-Defined Cellular Networks".