*International Journal of Advance Research in Engineering, Science & Technology(IJAREST),*
*ISSN(O):2393-9877, ISSN(P): 2394-2444,*
*Volume 2,Issue 4, April- 2015*

# A Survey on Software-Defined Network (SDN): From Concept to Implementation

**Monica Bhambere[1], Shilpa Bharam[2], Sujata Bhange[3], Prof. Ujwala Wasankar[4]**

***Abstract***

*Software-defined network (SDN) has become one of the most important architectures for the management of large scale, complex networks which may require re-policing or reconfigurations from time to time. SDN achieves easy re-policing by decoupling the control plane from data plane. Thus the network routers/switches just simply forward packets by following the flow table rules set by the control plane. SDN is a relatively new area, it has attracted much attentions from both academia and industry. In this paper we will conduct a comprehensive survey of the important topics in SDN implementation, including the basic concept, language abstraction ,controller, virtualization, Quality of service (QoS), security, as well as its integration with optical networks. We will discuss the future research trends in this exciting area. This survey can help both industry and academia R&D people to understand the latest progress of SDN designs.*

***Keywords-*** *Software-Defined Network (SDN), Language Abstraction ,Network Virtualization, QoS, Security*

## I. INTRODUCTION

Conventional Networks utilize special algorithms implemented on dedicated devices (hardware components) to control and monitor the data flow in the network, managing routing paths and determining how different devices are interconnected in the network. In general these routing algorithms and sets of rules are implemented in dedicated hardware components such as Application Specific Integrated Circuits (ASICs) [1].

Current network devices lack the flexibility to deal with different packet types with various contents because of the underlying hardwired implementation of routing rules [2]. Traditional network operations cannot be easily re-reprogrammed or re-tasked [3]. A possible solution to this problem is the implementation of the data handling rules as software modules rather than embedding them in hardware. This method enables the network administrators to have more control over the network traffic and therefore has a great potential to greatly improve the performance of the network. Such an idea is defined in an innovative technology, called Software-Defined Networking (SDN) [4].

**SDN contains three layers:**
**1.Application Layer.**
**2.Control Layer.**
**3.Infrastructure Layer.**

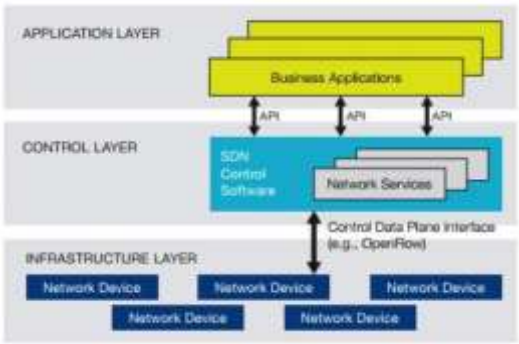


Fig 1.Architecture of SDN

## II.ARCHITECTURE OF SDN:

### 1. Language Abstraction:

In SDN the control function consists of two parts, i.e the controller with the program and the set of rules implemented on the routing/switching devices (Fig.1). This has an implication of making the programmer not worry about the low-level details in the switch hardware. The SDN programmers can just write the specification that captures the intended forwarding behavior of the network instead of writing programs dealing with the low-level details such as the events and the forwarding rules of the network. This enables the interactions between the controllers and switches

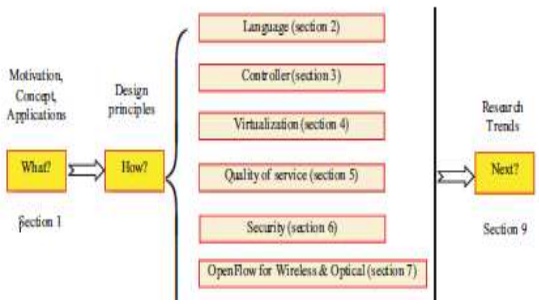


Fig 2: Organization of this survey

### 2.Controller:

The control plane can be managed by a central controller or multiple ones. It gives a global view of the SDN status to upper application layer. In this section, we look into the architecture and performance of controller in software defined networks. While SDN is suitable for some deployment environments and the enterprise the data plane should be made simple. Controller manage routing tasks for flow setup. The low-level switches have to communicate with the controller very frequently in order to obtain the instructions on how to handle incoming packets. This strategy can consume the controller processing power. In most SDN designs the central controller(s) can perform all the programming tasks. This model certainly brings the scalability issue to the control plane. A better control plane should be able to make the packet handling rate scalable

with the number of CPUs. It is better to always have the network status in packet level available to the controller. It can allocate /deallocate memory units, schedule different event handlers, and reduce the interrupts or system calls in order to decrease the runtime system load.

## 3.Virtualization:

As technology develops, the modern network becomes larger and more capable of providing all kinds of new services. The cloud computing, and some frameworks such as GENI, FIRE, G-Lab, F-Lab and AKARI, utilize the large-scale experimental facilities from networks. However, resources are always limited and users demands keep increasing as well. The sharing of network hardware resources among users becomes necessary because it could utilize the existing infrastructure more efficiently and satisfy users demands. Network virtualization in SDN is a good way to provide different users with infrastructure sharing capabilities [6]. The term OpenFlow often comes with network virtualization these years.

## 4.QoS:

In past decades, the Internet Engineering Task Force (IETF) has defined two types of Quality of Service (QoS) architectures, IntServ (integrated services) and Diffserv (differentiated services). The IntServ is difficult to implement in todays large networks due to too much operation overhead in different routers.

There are very few works targeting SDN QoS supporting issues. Among the few QoS models in SDN, Open- QoS [7, 8] is one of the most typical solutions. It has a comprehensive controller architecture to support scalable video Streaming in SDNs. We therefore summarize its principle first. A SDN QoS scheme called PolicyCop. PolicyCop uses the control plane of SDNs to monitor the compliances of the QoS policies and can automatically adjusts the control plane rules as well as flow table in the data plane based on the dynamic network traffic statistics.

## 5. Security:

SDN creates some new targets for potential security attacks, such as the SDN controller and the virtual infrastructure [9]. Besides all the traditional networks' attacking places (such as routers , servers, etc.), SDN has some new target points such as:
(1) SDN controller;
(2) Virtual infrastructure.
on the virtual switch and VM (virtual machine); The routing loops make packets never reach the final destination. In [10] it presents a dynamic algorithm which is built on header space analysis, and allows the detection of loops in SDNs. There the network model has been illustrated as a directed graph. Hence concepts of header space analysis have been translated into the language of graph theory. Rule graphs and the dynamic loop detection problem are studied. [10].

They have shown how to model a network as a directed graph. By analyzing the reachability and connectivity of the topology graph, a node-to-node, no-loop path can always be found. A dynamic strongly connected component algorithm is proposed in [10] to allow us to keep track of edge insertions and deletions. It can also be used to detect loops in a routing path.

## 6.Optical Connection:

There is a great deal of benefits when adopting SDN for optical network control:
(1) Current optical networks have difficulties
to react independently to requests from client systems distributed at the network edge. SDN provides programmable, abstracted interface for flexible application re-configurations in optical control units.
(2) Existing optical networks cannot easily upgrade the software in each optical switch due to the embedded software nature. SDN could easily upgrade services due to its separation of control and data planes.
networked re-programming and virtualization
(4) The cost of optical hardware is typically high,
especially the photonics and associated electronic components. SDN could reduce those costs due to its 'dump' hardware operations - just simply following the flow table.

# III. CONCLUSION:

In this paper we have comprehensively surveyed the design issues for SDN.
Especially we have covered all important issues in concrete network implementation including language abstraction, controller design, virtualization scheme, QoS support, security issues, and optical network integration.

# REFERENCES:

[1] S. Ortiz, "Software-Defined Networking: On the Verge of a Breakthrough?"
Computer, vol. 46, no. 7, pp. 10-12, 2013.
[2] H. Kim and N. Feamster, "Improving network management with software defined networking," IEEE Communications Mag., Vol. 51, No. 2, pp. 114-119, 2013.
[3] K. Bakshi, "Considerations for Software Defined Networking (SDN): Approaches and use cases," in Proc. of IEEE Aerospace Conf., 2013 , pp. 1-9.
[4] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in Proc. of IEEE INFOCOM, 2013, pp. 2211-2219.
[5] S. Fang, Y. Yu, C. H. Foh, and K. M. M. Aung, "A Loss-Free Multipathing
Solution for Data Center Network Using Software-Defined Networking Approach," IEEE Trans. on Magnetics, vol. 49, no. 6, pp. 2723- 2730, 2013.
[6] D. Drutskoy, Software-defined Network Virtualization, Master Thesis, Princeton University, 2012.
[7] H. E. Egilmez, B. Gorkemli, A. M. Tekalp, and S. Civanlar, "Scalable video streaming over OpenFlow networks: an optimization framework for QoS routing," in Proc. of IEEE Int. Conf. on Image Processing, 2011, pp. 2241-2244.
[8] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with endto-end Quality of Service over software-defined networks," in Proc. of Asia-Pacific Signal & Information Processing Association Annual Summit and Conf., Dec. 2012, pp. 1-8.
[9] ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking,
Aug. 2013.
[10] D. Kordalewski and R. Robere, "A dynamic algorithm for loop detection in software defined networks", 2012. [online] http://www.cs.toronto.edu/ robere/paper/networkgraph-1214.pdf
[11] "OpenFlow switch specification, version 1.0.0," [Online] http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf
[12] "OpenFlow in Europe  linking infrastructure and applications," [Online] http://www.fp7-ofelia.eu/