RESEARCH ARTICLE OPEN ACCESS

Network Traffic Control Using Distributed Control Plane of Software Defined Networks

Mr. Sachin Ashok Vanjari [1], Dr. R. B. Ingle [2]
P.G. Student [1], Dean Second Shift [2]
Department of Computer Science and Engineering
Pune Inst.of Comp.Technology, Pune
Maharashtra - India

ABSTRACT

The SDN layer generally works as a virtual software switch or router in place of (or in conjunction with) the physical network devices. Traditionally job of managing the traffic is performed by software embedded in the routers and switches, in SDN it is performed by software from outside the devices. For large scale network environment, centralized control of Software Defined Network (SDN) having a single controller has several issues associated with it such as single point failure, computational complexity growth, reliability and scalability. To solve these issues multi-controller implementation of Software Defined Network has been introduced. In proposed system SDN controllers are deployed in (distributed) tree like structure such that lower level controller is controlled by upper level controller so that all the network is controlled by single SDN controller centrally. Making possible to centralized control over the global view of network with improved architecture will be considered. Distributed control plane architecture equally distributes overall loads among all controllers arranged in distributed manner. It solves single point failure problem.

Keywords:- SDN, QoS, BG

I. INTRODUCTION

Now a day's it is important to have centralize control over the resources rather than distributed one. In current network architectures, the network devices are bundled with a specialized control plane and various specialized features. This essentially binds us with the features shipped with the device. SDN breaks these pieces apart; with hardware switches at the lowest level responsible simply for forwarding packets and on top of that a control plane that communicates with the data plane using well-defined interfaces. The need for SDN arose from the buggy and unpredictable nature of the distributed configuration of network. Hence, the idea of having a centralized control plane to control the entire network was proposed through SDN.

Software Defined Networking (SDN) [1] is an emerging new networking paradigm, which aims to introduce a new approach to the control and design of networks of various kinds, which makes network control plane directly programmable and the underlying infrastructure to be abstracted for applications and network services. It provides an abstracted centralized view of the distributed network state.

OpenFlow: OpenFlow allows the realization and implementation of re-configurable networks architectures named SDN which presents an attempt to solve the bottlenecks of traditional routing based networking deployments. Thus, OpenFlow [2] is the most advanced communication protocol between a control plan (one or more controllers) and a data plan (network devices such as switches and routers). OpenFlow is used as a basic protocol in centralized routing where transfer tables inside switches are programmed and controlled remotely. Having this central approach reduces the need for N number of intelligent nodes in a N-nodes topology.

With the explosive growth of data traffic over the past few years, the bottlenecks of traditional data networks have been exposed. An emerging technology, OpenFlow [3], has been developed at Stanford University, it is currently gaining more and more support from companies such as Cisco, Juniper, Microsoft, Google and Facebook.

An SDN comprises of a logically centralized controller which has a global view of the network and is responsible for all control decisions, this makes the control plane. These centralized controllers communicate with network-wide distributed forwarding elements via standardized interfaces. The data forwarding elements simply forward the packets; they form the network data plane. Briefly, the two components are [4]:

1. SDN Controller (SDN-C) - Control Plane node which determines forwarding path for each flow in the network and update the routing tables at data plane nodes.

2. SDN Forwarding Element (SDN-FE) - Data Plane node which simply forward the data packets based on the routing tables.

II. MOTIVATION

Web is a discriminating foundation throughout today's reality simply like transportation and power. The large deployment base of internet has made it quite difficult to evolve in terms of physical infrastructure, protocols and performance. With current demands on an exponential increase there is an urgent need for renovation of the infrastructure. In today's network architecture, the network devices and middle boxes are vertically integrated i.e. the equipment and programming is given by the producer and can't be redone voluntarily. New programming is not introduced due to contrary equipment, or at present accessible programming is not influences all the equipment capacities. Software Defined Networking (SDN) presented a centralized network control, where a controller manages a network from a global view of the network [5].

Initially, SDN network consists of single SDN controller. For small networks, single controller SDN network works well compared to traditional networking approaches. But single controller is not enough to control entire network. Deployments rely on a single controller network stop working if used single controller fails. Single controller network has problem related to response time and overload which increases delay. Type and number of controller used in network affect the overall performance of network. That is the reason to choose the problem related to kind of controllers used and placement of used controllers. Software Defined networking relies on directly programming the packet handling mechanisms of the network nodes by a network controller. It is understood that the behavior of the networking equipment is defined by software today. Programmability of the network, separation of the control plane from the data plane and a controller that has a view of entire network motivates to work in software defined networking domain.

A. Literature survey

Agarwal S. , Kodialam M. , Lakshman T.V. et.al.: [6] This paper gives a description about applying effective traffic engineering in scenario where software defined networks are incrementally deployed into existing network. It presents the optimization problem for the centralized SDN controller based on minimizing the maximum link utilization in the network. The results of simulation show improved network performance by the use of SDN in the existing network. This paper shows how controllers apply on different part of existing network. They also formulate the SDN controller optimization problem for traffic engineering with partial deployment and developed Fully Polynomial Time Approximation Scheme. Paper proposes improvement by analysis and ns-2 simulations. Proposed system in this paper is implemented in distributed architecture which is difficult to maintain.

Rihab JMAL and Lamia CHAARI FOURATI [7] This paper shows all the control-level logical decisions are taken at a central way, as compared to traditional networking. This paper present a routing solution based on SDN architecture implemented in OpenFlow environment and providing shortest path routing. Proposed system in this paper is implemented in centralized architecture, but it is not like tree structure which is difficult to maintain.

C. Rothernberg, C. N. A. Correa, R. Raszuk, et.al.[8] This paper discussed on the centralized Routing Control Platform (RCP) in context of SDN. A controller centric hybrid networking model is proposed in the paper.

Rahamatullah Khondoker et al.[9] gives selection criteria for controllers by comparing controller features. SDN is a new networking paradigm that separates the control plane of a networking device (e.g., a switch / router) from its data plane, making it feasible to control, monitor, and manage a network from a centralized node (the SDN controller). A decision making a template is proposed in this paper to help researchers choose the SDN controller that best fits their needs.

Zuhran Khan Khattak et al.[10] described Performance Evaluation of OpenDaylight SDN Controller. The performance analysis of network controllers is generally with the help of benchmarking. Benchmarking of recently developed OpenDaylight SDN controller is not done till now. Results of benchmarking of OpenDaylight SDN controller and Floodlight SDN controller are presented.

Shivaleela Arlimatti et al.[11] described a comprehensive survey on SDN and OpenFlow. Study of infrastructure, southbound, controllers, northbound and network applications is done. Challenges in the field of SDN are discussed which gives ideas to work on.

Deepankar Gupta et al.[12] discussed the Border Gateway Protocol (BGP) protocol useful for inter-SDN controller communication. Multiple SDN controller communication is done with the help of east-west interface, which share control plane parameters like Quality of Service (QoS), policy information, and so on.

Soheil Hassas Yeganeh et al.[13] proposed a Kandoo, a novel distributed control plane that offloads control applications over available resources in the network with minimal developer intervention and Realization of scalable Software Defined Network is done by limiting the overhead events are frequently processed in the data plane. This requires modifying switches and comes at the cost of visibility in the control plane. Kandoo, a framework preserves scalability without changing switches.

III. PROPOSED WORK

Actual flow of proposed architecture is explained by giving routing examples. These examples play important role to understand design of proposed distributed hierarchical control plane architecture. Illustrated how intra-area routing and inter-area routing is carried out. The two routing examples are based on the topology shown in Figure 1.1 In given topology, a domain has two sub-domains and each sub-domain has two areas. At the same time four hosts (host A, B, C and D) are located in the topology.

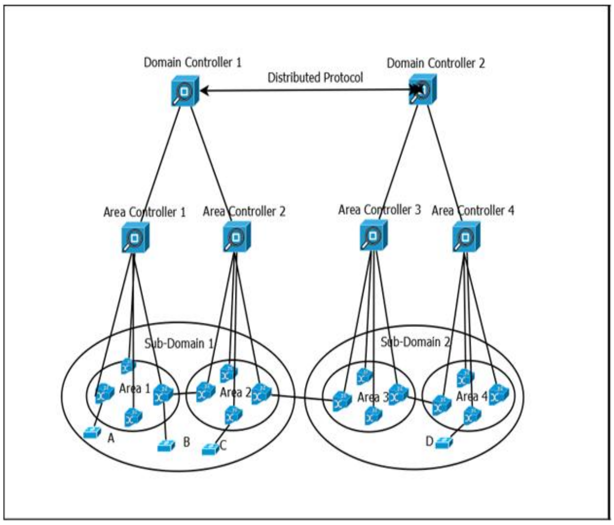

Figure 1.1: Architecture Diagram

A. *Intra-Area Routing*

The first example of intra-area routing, shows how to communicate between two hosts (such as Host A and Host B) in an area. When the Host A sends a data flow to the Host B, the switch connect to the host A generates a Packet-In message and sends the message to Area Controller 1. When Area Controller 1 receives the message, it checks whether the destination address of the data flow is in its area, as Host B is located in Area 1 Area Controller 1 look for the information of host B. Then it calculates the intra-area routing path from Host A to Host B based on the intra-area topology. Next, Area Controller 1 sends the routing rules to the switches in the path list, so that the switches install the rules for the data flow. Finally, when all the switches in the path list are installed routing rules, the data flow sent by Host A is forwarded to Host B. The intra-area routing sequence of proposed distributed hierarchical model is shown in Figure 1.2.

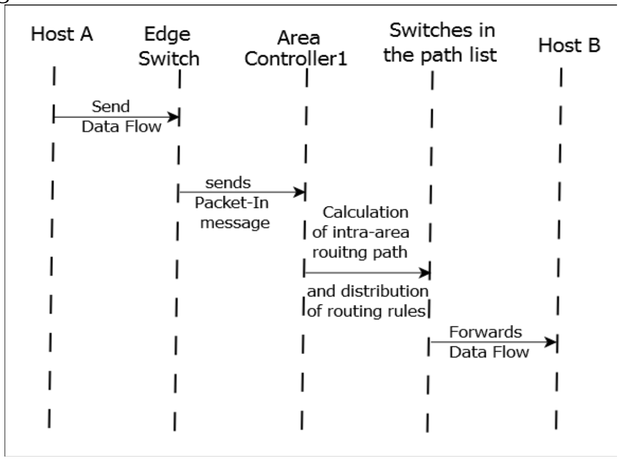

Figure 7.3: Intra-Area Routing Sequence of Proposed Architecture.

B. *Inter-Area Routing*

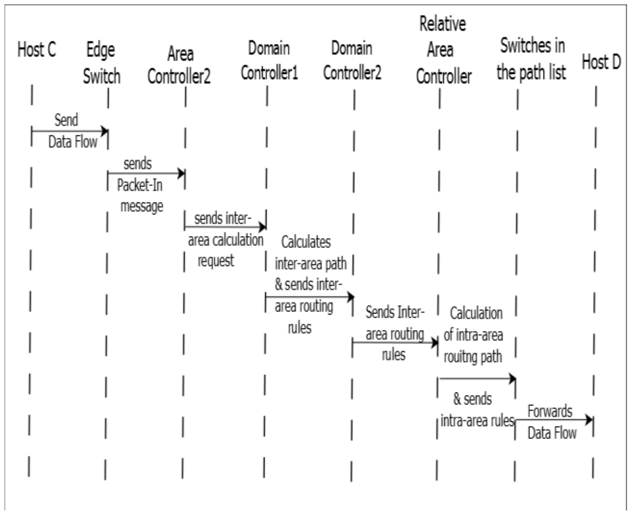

Figure 1.3: Inter-Area Routing Sequence of Proposed Architecture

The second example is an example of the inter-area routing. The example illustrates how Host C sends data flow to Host D with proposed architecture. When Host C sends a data flow to Host D, the data flow reaches the switch which Host C connects to. Then the switch generates a Packet-In message and sends the message to Area Controller 2. As host D is not in Area 2, when Area Controller 2 receives the message, it extracts the source address and destination address from the Packet-In message and encapsulates it to a simple request, sends the request to Domain Controller 1, and buffers the Packet-In message with an index. When Domain Controller 1 receives the request, it calculates the inter-area path according to the global abstracted network view. For the destination Host D is in the sub-domain of Domain Controller 2, Domain Controller 1 publish the inter area path routing rules to the Domain Controller 2 through the NoSQL distributed database. When Domain Controller 2 receives the inter-area routing rule messages, it sends the messages to the area controllers on the routing path. The inter-area routing sequence of proposed distributed hierarchical model is displayed in Figure 1.3.

IV. SIMULATION ENVIRONMENT

We need VMware workstation 11.0.0 and Mininet. Mininet used for simulation of distributed control plane. Mininet is most widely used network emulator for simulation of Software Defined Networking. MiniEdit is a graphical editor for mininet which is useful to draw particular topology.

A. *The Mininet:*

Prototyping and simulating large networks virtual mode is better option because when one wants to simulate large network having number of hosts, switches and SDN controllers. All having different configurations might be problematic. For simulating such large SDN network popular simulating tool Mininet is available [14].

The Mininet is a system that allows rapidly prototyping large networks on a single computer. It creates scalable Software-defined networks using lightweight virtualization mechanisms, such as processes and network namespaces. These features permit the Mininet create, interact with, customize and share the prototypes quickly.

B. Emulation environment specifications:

For carry out the experiment we have used Lenovo Z50 Forth generation Laptop having Intel Core i5 processor, 4GB of RAM running the O.S. Ubuntu/Windows 64 bits and VMware Workstation 11.0. In VMware Workstation 11, we installed the guest operating systems: Mininet Emulator version 2.0 on Ubuntu 14.04/Windows 8 64bits with 4Gb of RAM; Floodlight/OpenDay light Controller version 0.90. Wireshark and dpctl are used for measuring and monitoring the functionality of mininet.

In our system there are two domain controllers, four area controllers, any number of switches are used in designing of distributed hierarchical control plane. There is two type of data our system has first is deterministic and non-deterministic. Source address, destination address, number of packets to be sent are first type of data. Routing path, number of intermediate switches through which packets flow are fall in later category.

C. Experimental Result

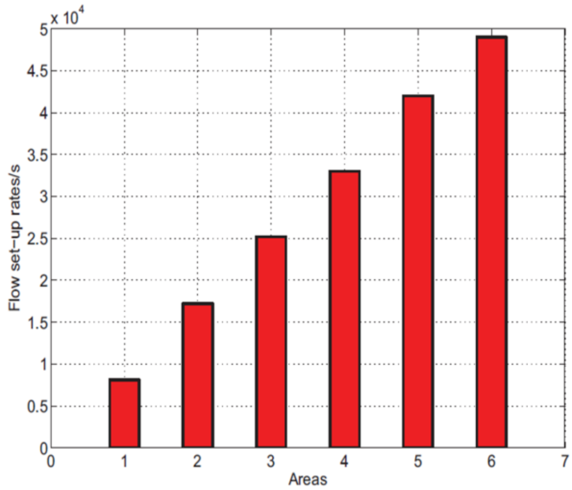

Figure 4.Throughput of Distributed Control Plane.

In this experiment, each area has 100-120 switches. There is some hardware and bandwidth limitations therefore random network topology is used. First of all domain controllers are started and then four area controllers. In this test, the number of areas to test, the flow set-up rate per second is increases. From Figure 4 above, One can say that single area controller handle 8126 new flows per second. With the increasing number of areas, the average throughput of the control plane is stable, and the overall throughput of our distributed control plane architecture increases steadily.

V. CONCLUSIONS

The study of existing software defined network is carried out. Traditional implementation of Software Defined Networks has problem of single point failure. To solve above problem multi-controller implementation of Software Defined Networks is proposed so that one controller fails another takes over. Multi-controller implementation of Software Defined Networks introduces a new problem of computational complexity which is tackled by making arrangement of controllers such that domain level controller controls the area controllers. Overall workload is divided among multiple controllers which solves the computational complexity issue.

VI. FUTURE ENHANCEMENT

In a large scale Software Defined Network, a hybrid control plane can be used. In this, the domain controllers should be Open Daylight controllers as they can communicate with each other easily and the area controllers should be Floodlight controllers so as to maximize performance.

For multi-controller communication we need inter controller communication protocols, which work differently at domain level and differently at area level.

REFERENCES

[1] A.C. Risdianto, E. Mulyana, "Implementation and analysis of control and forwarding plane for SDN," Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on, Bali, 30-31 October, 2012.

[2] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using Open-Flow: A survey," Communications Surveys Tutorials, IEEE, vol. 16, no. 1, pp. 493512, January, 2014.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," SIG-COMM Comput. Commun. Rev., vol. 38, no. 2, June, 2008

[4] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky and Steve Uhlig, Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol.103, no.1, pp.14,76, Jan. 2015.

[5] Thomas D. Nadeau and Ken Gray, "Software Defined Networks," OReilly Media, Inc., August 2013: First Edition. Available at http://itebooks. info/book/2888/

[6] Agarwal S., Kodialam, M., Lakshman, T.V., "Traffic engineering in software Defined networks," INFOCOM, 2013 Proceedings IEEE, vol. no.pp.2211, 2219, 14-19 April 2013.

[7] `Implementing Shortest Path Routing Mechanism using Openflow POX Controller" by Rihab JMAL and Lamia CHAARI FOURATI, 2014 IEEE.

[8] C. Rothernberg, C. N. A. Correa, R. Raszuk Revisiting Routing Control Platforms with the Eyes and Muscles of Software-Defined Networking,ACM-SIGCOMM HotSDN Workshop, 2012.

[9] Khondoker, R.; Zaalouk, A.; Marx, R.; Bayarou, K., "Feature-based comparison and selection of Software Defined Networking (SDN) controllers," Computer

[10] Applications and Information Systems (WCCAIS), 2014 World Congress on , vol., no., pp.1,7, 17-19 Jan. 2014

[11] Khattak, Z.K.; Awais, M.; Iqbal, A., "Performance evaluation of OpenDaylight SDN controller," Parallel and Distributed Systems (ICPADS), 2014 20th IEEE

[12] International Conference on, vol., no., pp.671,676, 16-19 Dec. 2014 Shivaleela Arlimatti, Suhaidi Hassan, Adib Habbal and Suki Arif, "Software Defined Network and Openow: A Critical Review," ARPN Journal of Engineering and Applied Sciences, Asian Research Publishing Network (ARPN), Vol. 10, NO. 3, February, 2015.

[13] Deepankar Gupta, Rafat Jahan, "Inter-SDN Controller Communication: Using Border Gateway Protocol," White Paper by Tata Consultancy Services (TCS), June, 2014.

[14] S. H. Yeganeh, Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," Proceedings ACM SIGCOMM HotSDN, September, 2012.

[15] Mininet : An Instant Virtual Network on your Laptop (or other PC). March, 2013. [Online]. Available: http://mininet.org/. accssed on : August, 2014.

[16] OpenDaylight, "http://www.opendaylight.org/software/" Accssed on : January, 2015.

[17] OpenDaylight, "OpenDaylight: A Linux Foundation Collaborative Project," Available: http://www.opendaylight.org Accessed on: March, 2014.

[18] OpenDaylight project "ODL-SDNi App:Main" Available: https://wiki.opendaylight.org/view/ODL-SDNi_App:Main Accessed on: March, 2015.

[19] "The case for SDNi SDN controller interconnection" Available: https://www.ietf.org/proceedings/84/slides/slides-84-sdnrg-5.pdf Accessed on: March, 2015.