A SURVEY ON MULTI THREADED CONTROLLER-FORWARDER QoS ARCHITECTURE FOR MULTIMEDIA OVER SDN

Ms. Pooja A. Baleghate, Mr. Sachin B. Takmare

P.G. Student, Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering, Kolhapur, India.

Assistant Professor, Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering, Kolhapur, India.

Abstract - Software-defined networking (SDN) is a way to computer networking that allows network administrators to manage network services through abstraction of higher-level functionality. This is accomplished by separating the system that makes conclusions about where traffic is sent (the control plane) from the systems that forward traffic to the selected destination (the data plane). Software defined networking is an emerging architecture that is dynamic, manageable and cost effective. This paper presents an overview of several techniques to improve the QoS of multimedia and we propose a novel approach to improve QoS of multimedia using multi-threaded controller-forwarder mechanism over SDN. We will apply this mechanism to streaming of layered videos which will provide a better QoS to videos and will perform faster than single controller-forwarder mechanism.

Keywords – SDN, QoS, Multimedia streaming, and OpenFlow.

I. INTRODUCTION

Streaming media is multimedia which is continuously awarded to and received by an end consumer while being delivered through a provider. With streaming media, a consumer does no longer have to wait to download a file to play it. As the media is sent in steady stream of data so as to play as it arrives. The current Internet architecture, which is developed for best effort data transmission, cannot make any assurance about end-to-end delay of a packet or the delay variation (jitter) between sequential packets which are crucial for media streaming [1]. In order to permit the network to sustain some level of Quality of Service (QoS) for multimedia traffic, the Internet Engineering Task Force (IETF) proposed several QoS architectures, like IntServ [2] and Diffserv [3], still none has been globally implemented and truly successful. This is because they are built on top of the current Internet's distributed hop-by-hop routing architecture lacking the end-to-end information of available network resources.

OpenFlow is a successful Software Defined Networking (SDN) paradigm that separates the control and forwarding layers in routing. SDN is an emerging architecture that is dynamic, manageable and cost effective. It is the physical separation of the network control plane from the forwarding plane and where a control plane controls several devices.This is achieved by shifting routing control functions from network devices to a centralized unit, called controller, while data forwarding function remains itself in the routers, called forwarders. The forwarders are organized via the OpenFlow protocol, which defines the communication between the controller and forwarders.

After a while, numerous application defined Networking conventions will quite often increase, yet for the present, the OpenFlow is most often utilized as a part of SDN. In a conventional approach, switches manage both high-level routing (the control plane) and packet forwarding (the data plane). In SDN, the control plane is decoupled from the physical system and set into a centralized controller. These controllers use OpenFlow to correspond with all components on the system. Countless community device retailers have effortlessly begun to provide OpenFlow-empowered switches or routers. Hence, SDN or OpenFlow will incrementally spread all by means of the arena quicker rather than later as new OpenFlow empowered switches are dispatched. OpenFlow has additionally attracted the awareness of countless organizations providing cloud administrations, and it's going to further permit system administration suppliers to present inventive multimedia administrations with gradually reconfigurable QoS. This is the fundamental concept behind using OpenFlow architecture in this work. Yet, present OpenFlow specification does now not provision communication between different controllers managing separate network domains. It is important to implement a distributed control plane with multi-threaded controllers and forwarders to manage multi-domain, multi-operator SDNs.

II. RELATED WORKS

Several non-standard distributed control plane architectures have been proposed to improve the QoS of multimedia in the literature.

Hilmi E. Egilmez and A. Murat Tekalp [4] proposed novel QoS extensions to distributed control plane architectures for multimedia delivery over large-scale, multi-operator Software Defined Networks (SDNs) using single controller-forwarder mechanism.

T. Koponen et al. [5] proposed Onix - a distributed platform that defines a general set of APIs to implement a control plane. In this model, a network-wide control platform, running on one or more servers in the network, oversees a set of simple switches. The control platform handles state distribution collecting information from the switches and distributing the appropriate control state to them, as well as coordinating the state among the various platform servers and provides a programmatic interface upon which developers can build a wide variety of management applications.

A. Tootoonchian and Y. Ganjali [6] introduced an event-based distributed control plane design, called HyperFlow, allowing efficient distribution of the network events among controllers.

Feamster et al. [7] proposed a software defined Internet exchange (SDX) architecture whose extensions will allow multi-site deployments of SDN. The single-node SDX architecture bears some resemblance to a conventional route server, but its design makes a few significant departures from a route server. First, the SDX controller allows each AS to apply a custom route selection process to select one or more best routes to each Internet destination. This feature contrasts with existing inter domain routing practices, whereby each AS must apply the conventional BGP route selection process to select a single best route to each destination. Second, the SDX controller can directly affect forwarding by updating switch-table entries, as opposed to indirectly affecting route control via BGP policy mechanisms.

Kotronis et al. [8] proposed a control plane model focusing on evolving inter-domain routing so that the legacy BGP remains compatible. The paper presents SDN concepts to improve inter-domain routing. The paper proposes to outsource the routing control plane of an ISP to an external trusted provider, i.e., the service contractor. The contractor specializes in routing management.

Raghavan et al. [9] introduced Software Defined Internet Architecture (SDIA) considering both intra and inter-domain forwarding tasks. The goal of this paper is to change architectural evolution from a hardware problem into a software one. And our solution is rather standard, borrowing heavily from long- standing (e.g., MPLS) and emerging (e.g., SDN) deployment practices. From the survey it can be seen that, none of current proposals provide a solution for overall network wide QoS control using multi-threaded architecture. A distributed control plane consisting of multiple controllers has been developed to provide better QoS for multimedia streaming. But instead of using single controller-forwarder mechanism in a distributed control plane, it would be beneficial to use a multi-threaded controller-forwarder mechanism which will perform faster to improve the QoS for multimedia streaming.

III. PROPOSED WORK

As of now, it is hard to dynamically change network routing on a per-flow basis. Conventionally, when a packet comes at a router, it checks the packet's source and destination address pair with the entries of the routing table, and forwards it as indicated by generally fixed, predefined rules (e.g., routing protocol) configured by the network operator. OpenFlow offers a new view to mostly cure this deficiency by allowing network operators to flexibly define different types of flows (i.e., traffic classes) and link up them to some set of forwarding rules (e.g., routing, priority queuing). The controller is the key network component where per-flow forwarding decisions can be made dynamically and forwarding tables, called flow tables, are updated accordingly.So as to ensure ideal end-to-end QoS for multimedia delivery, collecting up-to-date global network state information is important. However, over a large-scale multi-domain network, this is a difficult task because of dimensionality. The problem becomes much more difficult due to the distributed architecture of the current Internet.

The current Internet's inter-domain routing protocols such as BGP4 are hop-by-hop, and accordingly not reasonable for improving end-to-end QoS. OpenFlow solves this problem by employing a centralized controller. Instead of sharing the state information with all other routers, OpenFlow empowered forwarders directly send their local state information to the controller using the OpenFlow protocol. The controller processes each forwarder's state information and decides the best forwarding rules using up-to-date global network state information. Yet, the current OpenFlow specification is not suitable to large scale multi user telecommunication networks. Therefore, there is need for a distributed control plane consisting of multiple controllers each of which is responsible for a part (domain) of the network.

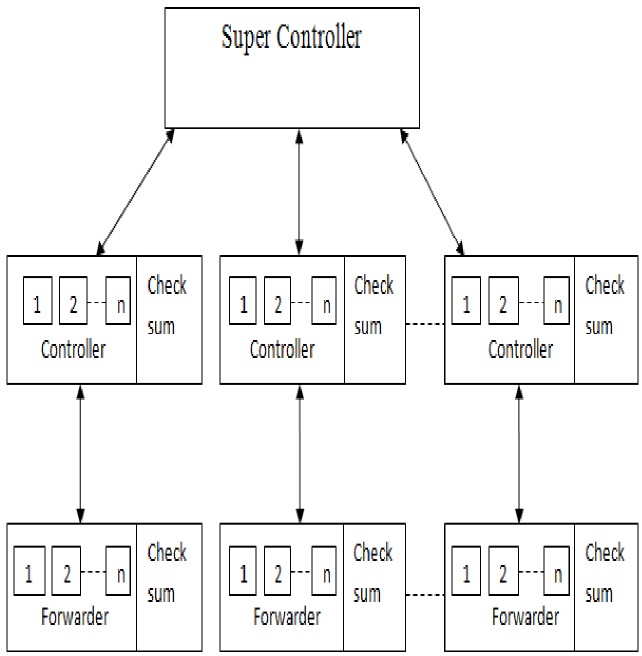

Fig.1 Multi-Threaded Controller-Forwarder Design

In the proposed design, a network will be divided into domains and every single domain will be controlled by a controller. Each controller will be in charge of intra-domain routing and forward the aggregated routing data of its domain to the super controller. The forwarder will be in charge of data forwarding function to the controller. Based on this aggregated routing information, the super controller will be in charge of deciding inter-domain routing. It will also forward the inter-domain routing information to every one of the controllers. After considering these outcomes, an optimized QoS routes will be decided. This design will then be employed to streaming of videos, which will give a superior quality of service to videos faster than a single controller-forwarder mechanism.

IV. CONCLUSION

This paper discusses the problems with current techniques to enhance the QoS of multimedia and proposes a multi-threaded controller-forwarder mechanism, which will perform faster than a single controller-forwarder mechanism to improve the QoS for multi-media streaming over software defined network.

ACKNOWLEDGMENT

There have been many contributors for this to take shape and authors are thankful to each of them.

REFERENCES

1. J. H. Saltzer, D. P. Reed, and D. Clark, "End-to-end arguments in system design,"ACMTrans.Comput.Syst.,vol.2,no.4,No v.1984.
2. R. Braden, D. Clark, and S. Shenker, RFC 1633 Internet Engineering Task Force, "Integrated services in the internet architecture: An overview," Jun. 1994.
3. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, RFC 2475 Internet Engineering Task Force, "An architecture for differentiated services," Dec. 1998.
4. Hilmi E. Egilmez, A. Murat Tekalp, "Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks", IEEE Trans. on Multimedia, vol. 16, no.6, Oct.2014.
5. T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks", OSDI, pp. 351–364, 2010.
6. A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for OpenFlow", in Proc.INM/WREN'10, pp.3–3, 2010.
7. N. Feamster, J. Rexford, S. Shenker, D. Levin, R. Clark, R. Hutchins, and J. Bailey, "SDX: A software-defined internet exchange", Open Netw. Summit, Apr. 2013.
8. V. Kotronis, X. Dimitropoulos, and B. Ager, "Outsourcing the routing control logic: Better internet routing based on sdn principles", in Proc. 11th ACM Workshop Hot Topics Netw., pp. 55–60, 2012.
9. B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker, "Software-defined internet architecture: Decoupling architecture from infrastructure", in Proc. 11th ACM Workshop Hot Topics Netw., pp. 43–48, 2012.
10. B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem", in Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN '12), pp. 7–12, 2012.
11. D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized? : State distribution trade-offs in software defined networks", in Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN '12), pp.1–6, 2012.