Classification of SDN distributed controller approaches: a brief overview

Mpho Nkosi, Albert Lysko, Lusani Ravhuanzwo, Thulani Nandeni;
Meraka Institute, CSIR,
Pretoria, South Africa
e-mail: {mnkosi2, alysko, lravhuanzwo}@csir.co.za

Andries Engelberencht
Department of Computer science
University of Pretoria,
Pretoria, South Africa

***Abstract*—Software defined networking offers a capability of separating control plane from data plane. The control plane is the key component of a network which ensures smooth management and operation of the entire network. However, network's resiliency and availability offered by SDN single centralized controller architecture is prone to single point of failure. Network operators are rather looking at distributed controllers' deployment to ensure better network resiliency and availability. This paper gives a brief overview of approaches available for distributed controller deployment and also highlights some of the challenges experienced in the SDN distributed controller environments.**

Keywords—Software Defined Networking (SDN);Application Programming Interface(API); distributed Controller;

I. Introduction

The inception of software defined networking (SDN) came with the notion of an SDN controller. An SDN controller is in essence the main operating system of an SDN based network. It is a software that runs on a server and provides required resources and intellection to oversee programming of the data plane based on network global view. The controller makes it easier to automate orchestration of network by using programmable control Application Programming Interfaces (APIs) that are available in an SDN environment. These APIs provide the controller with the ability to communicate with data plane, application plane and other controllers.

An SDN network can be deployed using a single centralized controller approach or a distributed controller approach. In a single centralized controller approach, a controller is used by a network to facilitate and manage all the forwarding devices in data plane. In a distributed controller approach, multiple controllers are used to facilitate and manage the forwarding devices.

Since the adoption of SDN paradigm, researchers have put more focus on the single centralized controller approach as the answer to overcoming the traditional networking limitations such as, but not limited to, complexity, scalability, and vendor dependence. However, the single centralized approach suffers from single point of failure and may not be scalable enough when a network grows.

Initially, distributed controller approach was particularly intended for large networks, for example, interconnecting large data centers. Recently, network operators are focusing on distributing the control plane by deploying multiple controllers so as to overcome limitations, such as single point of failure, scalability and reliability, which come with the centralized controller approach.

This paper describes the classification of distributed controller approaches and some of the limitations thereof. The paper starts by providing an overview of the APIs used in SDN.

II. Overview of the SDN API

In an SDN environment, as depicted in Fig 1, APIs are used by a controller to communicate with the application plane, data plane and other controllers. A data plane is made up of forwarding devices, for example, SDN switches/routers. A controller uses southbound APIs to communicate, manage and configure the data plane. The widely used southbound API is the OpenFlow which is a protocol defined and maintained by OpenFlow Networking foundation (ONF) [1]. Although there are many other southbound APIs to date, for example: Forces [2], Open vSwitch database (OVSDB) [3], Hardware Abstraction Layer (HAL) [4], OpenFlow is the most supported in the commercial SDN switches/routers [5]. The control plane support the use of different southbound APIs. This is to allow the use of different devices and protocols in a network [6].

To communicate with the application plane, the control plane uses the Northbound APIs. The application plane is made up of end user applications that feed on the SDN network services. Though there is no standard northbound APIs, most controllers support a couple of the different northbound APIs such as RESTFUL APIs, ad-hoc API and others [7].

Eastbound/westbound APIs are used by controllers in distributed controller architectures to communicate with each other. For instance, distributed controllers communicate with each other to exchange information and update each other. Unlike the south and north bound APIs, there are no popularly used eastbound/westbound interfaces. Each controller implements and uses its own eastbound/westbound API. This poses a challenge in environments where interoperability is required.

The APIs are the key components of the SDN which makes it a powerful tool for network control and operation.

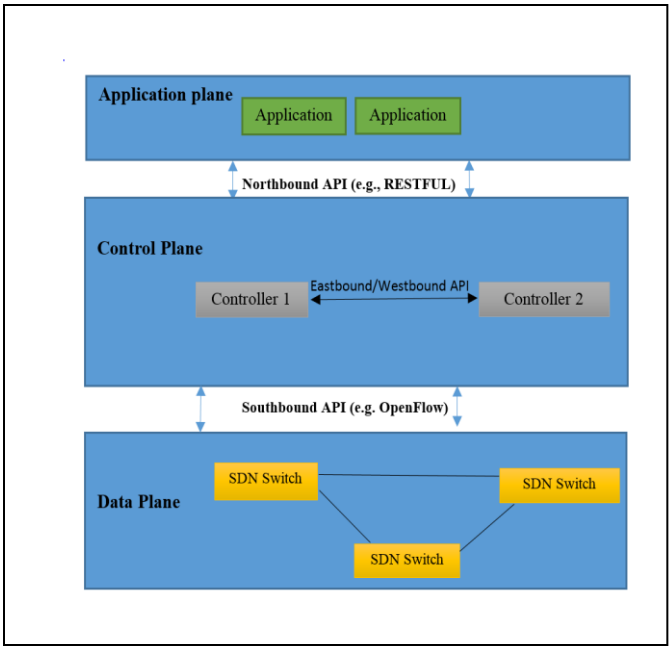

Fig. 1. An SDN architecture

The APIs provide the SDN with "*nice to have*" features such as programmability, protocol independence, ability to modify network parameters as required, elasticity and granularity [8].

A Control plane requires the APIs to oversee, manage and facilitate all other SDN planes. The control plane can either be made of a single centralized controller or multiple controllers. The latter offers more benefits such as resiliency and scalability. There are different approaches that can be used to deploy a control plane with multiple controllers

Although such approaches offer promising solutions for efficient, reliable, fault tolerant and scalable SDN networks, they come with their shortcomings. The next section describes different approaches in distributed controller SDN network.

III. DISTRIBUTED CONTROLLER APPROACHES

Limitations that come with the SDN single centralized controller architecture led to more research on the distributed controller approaches. In a distributed controller approach, the control plane is made up of a set of controllers that work together towards achieving a resilient, scalable, highly available network. For example in [9], distributed controller approaches can be classified into four categories: Logically distributed controller approach, physically distributed approach, hierarchically distributed approach and hybrid controller distribution approach.

Physically distributed controller approach. Controllers are physically distributed in the control plane. Usually, this approach is used to interconnect multiple domains. For example, in large scale networks, a network could be divided into smaller networks with each network maintained and facilitated by a local controller. All controllers have the same global view of the network. ONIX distributed controller [10] is an example of this approach.

Logically distributed controller approach. This approach utilizes the notion of a centralized controller but with each controller having different responsibilities. Like in the physically distributed approach, this approach can be used in large networks wherein the network can be divided into smaller networks thereby distributing the control logic. Each controller is responsible for its domain but still has a global view of its network. Each update in one controller has to be shared with all others to maintain the global view of the network. This makes it strenuous to maintain such controllers. DiSCO [11] is one of such controllers.

Both in Logically and physically distributed approaches, each controller has a global view of its network. The difference lies in how the controllers are placed and type of communication among the controllers. In a logically distributed approach, each controller has to update other controllers each time there is a network state change. In physically distributed approach, the controllers update each other periodically.

Hierarchically distributed controller approach. The control plane is made up of more than one layer of controllers. Each controller may have different responsibilities and make decision based on a partial view of the network. However, the upper layer of the control plane act as a central controller which poses the single point of failure syndrome [12]. Kandoo controller [13] is an example of a hierarchical distributed controller.

Besides in the hierarchical controller approach which follows the vertical structructure, the control plane in all other approaches usually, follows a flat structure. That is to say, controllers are placed horizontally on a single layer. In large scale networks, the flat structure may experience computational complexity problems whereas the vertical structure can experience a path stretch problems [14].

Hybrid controller approach. The control plane of this approach can be deployed as a mixture of all other approaches to try combine the benefits of each and avoiding limitations of each. SOX/DSOX [15] controller are examples of hybrid approach.

As with the single centralized controller approach, the distributed controller approaches have their challenges. For instance, some of the challenges that researchers have been looking into are: i) the placement of the controllers- controllers have to be positioned in way that the overall performance of the control plane is not degraded; also taking into consideration the number of controllers used [16]. ii) The inter-controller latency which is also affected by number of controllers in the control plane [17]; iii) most of the mapping between a switch and controller is static. That is to say, the links and positions of the controllers and switches cannot be changed [18].This poses a challenge of the controller being overloaded when the controller experience large number of flows.

As compared to the traditional network architectures, there are few SDN network management tools (e.g., Balance flow [19], DIFANE [20]) to date. To ensure efficiency and resiliency of the distributed controller approaches, network monitoring and management tools compatible with the distributed controller's approaches are required.

IV. CONCLUSION

SDN has gained momentum and popularity in the networking industry. It is seen as the promising future of networking offering features like protocol independence, programmability and granularity. The control plane is the key component that oversee and manage the SDN network by using APIs offered in an SDN environment. The adoption of SDN emerged with single centralized controller in the control plane as the main architecture. However, this architecture is prone to single point of failure. Distributed controllers architecture offers promising alternate solutions that can aid the centralized control limitations. This paper has presented different approaches that can be used when deploying a control plane using distributed controllers, namely: the logically distributed controller approach, physically distributed controller approach, hierarchical controller approach and the hybrid controller approach. Limitations of each approach were also described.

REFERENCES

[1] Open Networking Foundation (ONF) : https://www.opennetworking.org

[2] Haleplidis, E., Joachimpillai, D., Salim, J.H., Lopez, D., Martin, J., Pentikousis, K., Denazis, S. and Koufopavlou, O., 2014, September. ForCES applicability to SDN-enhanced NFV. In *2014 Third European Workshop on Software Defined Networks* (pp. 43-48). IEEE.

[3] Caba, C. and Soler, J., 2015, April. APIs for QoS configuration in Software Defined Networks. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on* (pp. 1-5). IEEE

[4] Ogrodowczyk, L., Belter, B., Binczewski, A., Dombek, K., Juszczyk, A., Olszewski, I., Parniewicz, D., Corin, R.D., Gerola, M., Salvadori, E. and Pentikousis, K., 2014, May. Hardware abstraction layer for non-OpenFlow capable devices. In *TERENA Networking Conference* (pp. 1-15).

[5] Richardson, L., Amundsen, M. and Ruby, S., 2013. *RESTful Web APIs*. " O'Reilly Media, Inc.".

[6] Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S. and Uhlig, S., 2015. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), pp.14-76

[7] Hoang, D.B. and Pham, M., 2015, September. On software-defined networking and the design of SDN controllers. In *Network of the Future (NOF), 2015 6th International Conference on the* (pp. 1-3). IEEE

[8] Jarschel, M., Zinner, T., Hoßfeld, T., Tran-Gia, P. and Kellerer, W., 2014. Interfaces, attributes, and use cases: A compass for SDN. *IEEE Communications Magazine*, *52*(6), pp.210-217.

[9] Akyildiz, I.F., Lee, A., Wang, P., Luo, M. and Chou, W., 2014. A roadmap for traffic engineering in SDN-OpenFlow networks. *Computer Networks*, *71*, pp.1-30.

[10] J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. and Shenker, S., 2010, October. Onix: A Distributed Control Platform for Large-scale Production Networks. In *OSDI* (Vol. 10, pp. 1-6).

[11] Phemius, K., Bouet, M. and Leguay, J., 2014, May. Disco: Distributed multi-domain sdn controllers. In *2014 IEEE Network Operations and Management Symposium (NOMS)* (pp. 1-4). IEEE.

[12] Akyildiz, I.F., Lee, A., Wang, P., Luo, M. and Chou, W., 2016. Research challenges for traffic engineering in software defined networks. *IEEE Network*, *30*(3), pp.52-58.

[13] Fu, T., Hu, L., Yu, X., Hu, J. and Zhao, K., 2016. Role-based intelligent application state computing for OpenFlow distributed controllers in software-defined networking. *Soft Computing*, pp.1-9.

[14] Fu, Y., Bi, J., Gao, K., Chen, Z., Wu, J. and Hao, B., 2014, October. Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks. In *2014 IEEE 22nd International Conference on Network Protocols* (pp. 569-576). IEEE.

[15] M. Luo, Y Tian, Q Li, J Wang, W Chou. 2012. "SOX –A Generalized and Extensible Smart Network Openflow Controller(X)", The First SDN World Summit, Germany

[16] Hock, D., Hartmann, M., Gebert, S., Jarschel, M., Zinner, T. and Tran-Gia, P., 2013, September. Pareto-optimal resilient controller placement in SDN-based core networks. In *Teletraffic Congress (ITC), 2013 25th International* (pp. 1-9). IEEE

[17] Ros, F.J. and Ruiz, P.M., 2016. On reliable controller placements in software-defined networks. *Computer Communications*, *77*, pp.41-51.

[18] Koponen, T., Casado, M., Gude, N., Stribling Blial, O., Ben Mamoun, M. and Benaini, R., 2016. An Overview on SDN Architectures with Multiple Controllers. *Journal of Computer Networks and Communications, 2016*

[19] Hu, Y., Wang, W., Gong, X., Que, X. and Cheng, S., 2012, October. Balanceflow: controller load balancing for openflow networks. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems* (Vol. 2, pp. 780-785). IEEE.

[20] Yu, M., Rexford, J., Freedman, M.J. and Wang, J., 2010. Scalable flow-based networking with DIFANE. *ACM SIGCOMM Computer Communication Review*, *40*(4), pp.351-362.