Computers and Electrical Engineering 57 (2017) 162–175

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

An East-West interface for distributed SDN control plane: Implementation and evaluation

Fouad Benamrane*, Mouad Ben mamoun, Redouane Benaini

LRI, Faculty of Sciences of Rabat, Mohammed V University in Rabat, Morocco

ARTICLE INFO

Article history:
Received 25 February 2016
Revised 9 September 2016
Accepted 12 September 2016
Available online 28 September 2016

Keywords:
Software-defined networks
OpenFow
Multiple controllers
Distributed control plane
East-West interface
Performance

ABSTRACT

Software Defined Networking (SDN) is a new concept within the networking field that allows programmability, automation, agility of services, and innovation using physically or logically centralized controllers. However, there is a lack of SDN in broadly distributed domains, where each domain has its own Controller. In this paper, we propose and implement a Communication Interface for Distributed Control plane (CIDC) that allows synchronization, the exchange of notifications as well as services between multiple distributed SDN Controllers. We have implemented distributed services such as Firewall and Load Balancer to improve the security and overall quality of service in distributed SDN architecture. The proposal for this research was evaluated in real wide-area network topologies, and the results show the feasibility of our CIDC in term of performance compared to the previous models based on a set of controllers arranged in a cluster.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Like significant advances that have been made in recent years in various areas of information technologies such as computer vision [1,2], big data, cloud computing [3], etc., we believe that the networking field is experiencing a major change with the emergence of SDN technologies. The main idea behind SDN is to remove the decisional part or the control plane from network devices such as switches and routers and shift it to a single point of control, the SDN control plane that manages all equipment in a unified way. The communication between the SDN control plane and the physical infrastructure devices is established via a southbound interface, such as OpenFlow. Network devices, which contain only the data plane in SDN, become simple packet forwarding devices that can be programmed through rules that are set by the controller. One advantage of the SDN concept is the centralization of the control that provides an abstract view of the underlying infrastructure to applications running above the SDN control plane via a northbound API. This abstraction allows these applications to treat the network as a unique entity, which simplifies configurations, the management, and the development of new services.

SDN was initially designed for testing new protocols in campus networks, but thereafter several works have shown that this concept is suitable for several types of networks and could resolve real issues, such as data centers, Wireless, and WAN networks. For example, it limited the consumption of electric power in data centers [4], reduced the cost of equipment and complexity of network configuration and management in Wireless networks [5]. However, using a single centralized

* Corresponding author.
E-mail addresses: benamranefouade@gmail.com (F. Benamrane), ben_mamoun@fsr.ac.ma (M. Ben mamoun), benaini@fsr.ac.ma (R. Benaini).

http://dx.doi.org/10.1016/j.compeleceng.2016.09.012
0045-7906/© 2016 Elsevier Ltd. All rights reserved.

controller generates new challenges in term of scalability and performances, especially the capacity of the controller to handle a lot of requests and by consequence the ability to scale when the network size grows.

To tackle the above issues, two main propositions of control plane designs with multiple controllers are given in the literature: logically centralized and logically distributed control planes. The logically centralized control plane balances charges between controllers and uses a shared database to unify decisions. However, it requires extensive synchronization between controllers and it is not suitable for large and highly distributed networks. So, the second proposition of control plane designs that uses logically distributed controllers was proposed to extend SDN for large distributed networks, where each controller manages its domain and distributes the necessary data to other controllers. The primary use of this category is in large data centers and WAN networks that suffer from the high cost and latency, due to the complexity of the infrastructure and protocol (e.g., BGP, and MPLS) that handle the traffic.

In logically distributed SDN architecture, the communication between multiple controllers is of primary importance. However, to the best of our knowledge, there is no standard for this communication nowadays. In this paper, we aim to contribute to the development of logically distributed SDN control planes, by providing an east-west interface that we call Communication Interface for Distributed Control plane. Our CIDC provides (i) communication modes such as Notification, Service, or Full to exchange messages between controllers and customize the desired behavior of each controller in the network, and proposes (ii) new mechanism based on policy sharing to support distributed services such as Distributed Firewall Service (DFS), and Distributed Load Balancer Service (DLBS) and secure the communication between controllers using Secure Socket Layer (SSL).

The remainder of this paper is organized as follows. Section 2 reviews previous works on distributed SDN architectures and highlights their main results. Section 3 presents the algorithms of our CIDC. Section 4 describes the implemented services in CIDC interface. In Section 5, the emulation setup of the experiments is detailed. The results of these experiments are thoroughly analyzed in Section 6. Finally, Section 7 concludes the paper and depicts some lines for further investigation.

2. Related work

The most used controllers that support physically centralized architecture are Nox [6], Floodlight (FL) [7], and Opendaylight (ODL) [8]. However, this architecture is not suitable for large networks, because the processing capacity of the controller is limited. Afterward, two main approaches are used to tackle the lack of managing distributed architecture with a single controller, the logically centralized and logically distributed approaches. In the logically centralized approach, which a set of controllers collaborate to manage the network and they have the same view of the network and the same shared database, the logically distributed approach was proposed for networks that are highly distributed over multiple domains. In this approach, each domain is managed by its controller and can share only some useful information with the other controllers to achieve some services such as the topology view.

Many implementations in the literature are following the logically centralized control approach, like Onix [9], HyperFlow [10] and ODL . Onix uses the Distributed Hash Table (DHT) to store the distributed network information and runs on a cluster of one or more physical servers. HyperFlow adopts WheelFS [11] as distributed file system, to build a global network view and each controller takes charge of its network. The synchronization between controllers should be announced for some events such as link status changes that could affect the network view. This approach operates for a few applications that are based on the global view and has a lack of flexibility regarding the distribution of events. The ODL controller, which is well-supported SDN controller, needs more description because it will be used in our experiment. The ODL starts by building the data structure trees using the Yang modeling language and MD-SAL. These data structures are mainly the Configuration tree to store the desired state of the system, and Operational tree to provide current runtime status. After building the trees, the ODL could form the cluster to support multiple controllers based on Akka framework,[1] and build robust concurrent and distributed applications. The cluster provides Elasticity, Fault-tolerant, Decentralization, Peer-to-Peer, and non-Single Point Of Failure. But, it is not feasible for logically distributed architecture where each controller manages its domain and has its own database. Moreover, the cluster consumes more resources to build the information trees. For example, the ODL divides the information trees into subtrees, where each subtree contains particular information (e.g., topology, inventory, ...) that is stored on one member. To achieve high availability (HA), ODL replicates the sub-tree to other members using Raft mechanism [12]. These operations require more bandwidth and time to process, and could affect the performance of the network.

To achieve a scalable control plane, logically distributed architectures are proposed. Some works suggest a design for the east-west interface between multiple controllers such as SDNi [13], and East-West (EW) Bridge [14]; however, there is no standard for this interface. Firstly, the EW Bridge is designed to support different controllers/ NOSes with various local network view storage systems. The EW Bridge uses new functions to guarantees the data integrity between controllers if it is needed, such as Network Virtualization, EW Bridge, and LLDP Extension. To synchronize data between controllers it uses publish/subscribe (pub-sub) [15] model. The pub-sub model originally addresses the problem of multicast or group messaging, it is used for SDN because it is more scalable than client-server model, and uses parallel operation and message caching to publish the message to the queue. However, the scalability under high load is still a research challenge, and

[1] http://akka.io/.

Download English Version:

https://daneshyari.com/en/article/4955374

Download Persian Version:

https://daneshyari.com/article/4955374

Daneshyari.com