Role-Based Multiple Controllers for Load Balancing and Security in SDN

Dharmendra Chourishi, Ali Miri, Mihailo Milić and Salam Ismaeel

Department Computer Science, Ryerson University, Toronto, Canada
dharm, ali.miri, m4milic, salam.ismaeel@ryerson.ca

***Abstract*—Software Defined Networks (SDNs) are gaining success in modern IT. While using SDN, enterprises and the research sector have encountered many challenges related to security and exposed weaknesses in heavy load management systems. In this paper, three architectures are proposed which are designed to work with all existing OpenFlow, OpenStack and OpenDaylight platforms. These architectures improve the controller framework without any interruption to network operation. At the same time, some of the key issues related to load balancing and security in a nSDN framework will be addressed.**

This work will support the IT community not only by reducing the cost of SDN network architecture but also by reducing the cost of load balancing and security that are very important in the networking world.

***Keywords*- Software-Defined Networks, controller, security, load balancing**

I. Introduction

Cloud computing with powerful SDN architecture will provides a new business paradigm for resources. It enables organizations and communities to create and use IT and business services on demand from optimal sources to maximize utilization and cost-effectiveness. This can be between enterprises or within a single enterprise. Economic, environmental, and global activities shaping regional markets, products, and services in many industrial sectors would be benefited.

In a cloud network model, secure transformation of packets plays a crucial role. There are many networking configurations and installation strategies adopted for various environments. Several security experts suggest that SDN is natively insecure as it removes hardware boundaries such as firewalls that maintain security [1]. Securing the controller and establishing trust between controllers is a key challenge. Similarly, DDoS and long waiting queues are threats that exist in SDN [2]. Providing load-balancing with improved security is key to SDN adaption.

This work highlights the need for an architecture, which can strengthen the capacity of the controller and decrease the hurdles in the packet forwarding processing. The controller should be distributed and mapped properly with switches for better network management.

The rest of the paper is organized as follows. After discussion of related work in Section II. Section III gives details about SDN architecture. Security and load balancing based analysis, work done in these fields and findings are described in Sections IV and V. Section VI gives a description of three proposed SDN architectures and their impact. Conclusions are given in Section VII.

II. Related work

There are a number of recent works, which deal with distributed implementations of OpenFlow controllers. Yazici *et. al.* [3] presented a distributed controller and an associated coordination framework. While Dixit *et. al.* [4] proposed ElastiCon, an elastic distributed controller architecture in which the controller pool is dynamically grown or shrunk according to traffic conditions and the load is dynamically shifted across controllers. Schmid and Suomela [5] proposed a supported locality model of distributed computing, in which each controller only needs to respond to an event that takes place in its local neighbourhood. Koponen *et. al.*. [6] introduced a new platform called *Onix*: a distributed control platform for large-scale production networks, where control planes written within Onix operate on a global view of the network, and uses basic state distribution primitives provided by the platform.

To the best of our knowledge, the separation of the application controller and packet controller has not been done before. This work introduces the concept of distributed role-based controller architecture, by keeping this in mind and utilizing the already implemented concepts and architecture of Hadoop's distributed system. Therefore, this unique approach can offer possible solutions to dynamic load balancing and security in SDN architecture. We believe that this hybrid architecture will bring better scalability and reliability in a generalized environment for SDN than is available in previously proposed solutions.

III. SDN Architecture

SDN networking proposes that the control plane (decision maker) be decoupled from the data plane (follower) [7], so that their roles and places are also divided in network administration.

The control plane resides in a server, is executed through software, and is responsible for managing the flow of data in the network. It decides where and how to forward packets, as it is controlling the network traffic.

The data plane is a follower of the controller and manages packet forwarding flow by maintaining routing flow tables

2015 IEEE Canada International Humanitarian Technology Conference (IHTC)
978-1-4799-8963-8/15/$31.00 ©2015 IEEE

and access control lists. It forwards packets in the network, following the rules given by the control plane [7], [8]. If any unidentified data packet is encountered, it is immediately forwarded to the control plane. The control plane assigns rules for the requested packet, and gives these to the data control plane. As per the rule given, the data plan updates its flow table and follows the new instruction.

IV. SDN Security

SDN removes hardware boundaries such as firewalls that maintain security and reveals new, unprotected surfaces to attack. To protect from attacks, firewalls, Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) can be deployed in the network dynamically in real-time. It is claimed that this enables SDN to introduce stronger security practices and overcome challenges to hardware boundaries.

From a security point of view, SDN actually monitors packets at multiple levels throughout the network. Vague boundaries make it impossible to determine where to deploy security devices such as firewalls. However, SDN can route all traffic through one central firewall. Traffic flowing through a single point is easy to attack, but facilitates real time capture and analysis for IDS and IPS [9]–[11]. Likewise, load balancing is also an obvious challenge in IT, particularly in handling extreme load conditions. However, centralized SDN controllers intelligently manage loads by prioritizing packets across the entire network, without having to write rules across individual switches.

In this regard, Software-Defined Security (S.D.Sec) is a dynamic way to design, deploy and manage networking services, by decoupling network functions. Firewall and IDS are separated from basic hardware devices, so that they can run in software, similar to SDN decoupling, where the control plane and data forwarding plane are separated. This is a step in the transition of traditional to virtualized infrastructure, which will have virtual servers, storage and even firewalls [12].

V. SDN Security and Load Balancing Issues

Security concerns are consistently identified as a major barrier when dealing with data transformation. Securing user data is a major field of research within mobility and virtualization, so new security issues must be well understood and handled. End user devices and data centre resources including hypervisors, storage devices, servers, switches, and routers must be secured in the network. Various available security solutions are difficult to deploy and manage within a large-scale multi-vendor environment. Even with regular security solutions provided by researchers, SDN is still struggling to gain the confidence of the enterprise world. Security challenges and load balancing are two very crucial problems emerging in the SDN Enterprise market [13], [14]. As far as security policies are concerned, it is difficult to control computing, storage, and network domains, and multiple data centres across the network. There are various security issues, which can pose major threats to SDNs [15]:

- Securing and protecting the controller

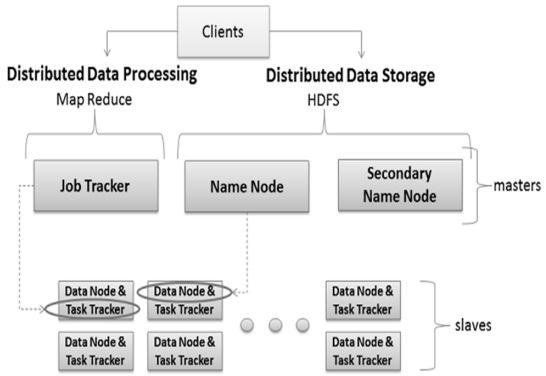

Figure 1: Hadoop Architecture

- Establishing trust between the controller and the switch
- Ensuring the integrity of the application
- Developing a robust policy framework
- Forensics and remediation

The following are some of the major threats to SDNs:

- DoS / DDoS
- Long waiting queues

VI. Proposed Solutions

We are proposing three architectures, which will help in load balancing and security of the centralized logical control plane in SDN technology. The proposed architectures are related to that of the Hadoop architecture depicted in Fig.1, where the Name Node (NN) and Second Name Node (SNN) play the role of a centralized server and provide a risk-avoiding and -recovering architecture.

A. First Proposed Controller Architecture

This architecture is designed to eliminate bottle neck issues, where all forwarding packets must go through a fixed point of the controller in the name of security, causing delays and increasing risk due to its centralized nature. This newly proposed controller architecture depicted in Fig.2 decouples application monitoring and packet monitoring. A separate application controller will trace malicious applications and a packet controller, with the newly proposed concept of Role-Based Controllers, will perform packet monitoring. Furthermore, the packet controller is distributed for security and load balancing reasons, as Hadoop's architecture is used. By separating application monitoring and packet forwarding monitoring points, the controller gets more space to concentrate on these crucial roles.

Like Hadoop's NN and SNN architecture, this new model will have a main controller and a secondary controller. In the case of a failure of the main controller, the secondary controller will take charge and play the same role as the main one. The secondary controller will maintain a flow/log table, update all transmissions, and continue with normal processing.

When the main controller gets stuck or goes down due to DDoS or for some other reason, this secondary controller will start from where the main controller left off, as per the

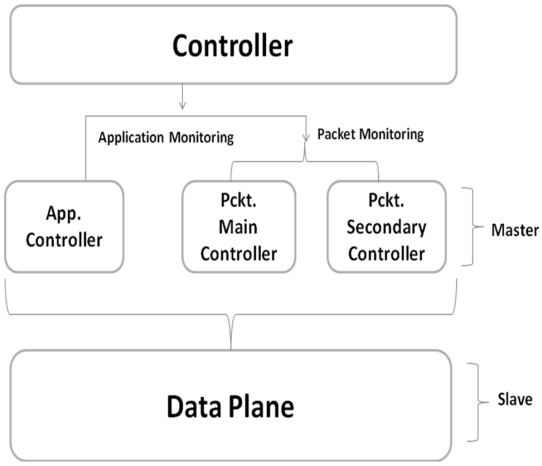

Figure 2: Controller Architecture

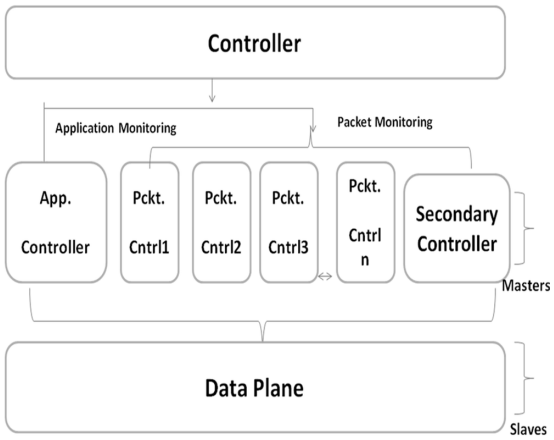

Figure 3: Multiple Controller Architecture

recording on flow table. The proposed controller architecture will sort out the risk factor where a single logical controller with security checkpoint could become an easy target for the attackers.

B. Second Proposed Controller Architecture

In the second proposed architecture, instead of a main controller and secondary controller, multiple controllers are added as depicted in Fig.3. This is an upgrade for load balancing. It will solve the problem of unnecessary queuing and delay in packet transmission. We propose multiple controllers that will have self-upgraded and self-synchronisation systems with other controllers.

A protocol for selection of packet and load distribution will be followed in parallel with overall functioning. Built-in tables or database files will be maintained in each controller, which will keep track of and record packets and sequence of that individual controller. Moreover, complete transformation details will be recorded in a secondary controller.

In case of any controller failure, the secondary controller will take charge and provide its services as proposed in the first case. This proposed architecture will address the problem of load balancing and unnecessary queuing, which causes delay in packet transformation.

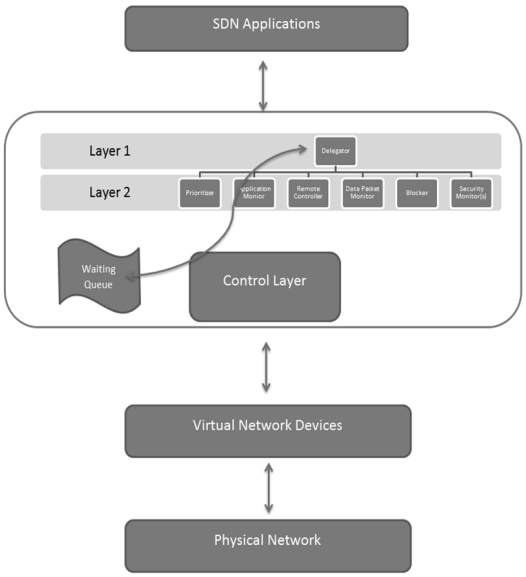

Figure 4: Hierarchical Controller Architecture

C. Third Proposed Controller Architecture

In the first and second proposed architectures we introduced the concept of role-based controllers to better aid in the completion of the varies tasks that a controller is asked to perform. However, it is difficult to organize and define roles for controllers in this new system, since in our previous two solutions we focused only on splitting the application monitoring from the packet controller. This was done to deal with the issue of load balancing in a software-defined network. The third proposed architecture introduces the concept of hierarchical role-based controller architecture to encompass other tasks that a controller must be able to perform.

As shown in Fig. 4, we would propose that the Control Layer be split into a two-layer system with a delegator or master controller, as well as lower level controllers that perform specific tasks, which are assigned by the Delegator. This architecture would become more precisely defined based on the demands of the network and as the individual network administrator sees fit.

Various roles of controller:

- Delegator: The Delegator's primary role is to monitor the status of all the controllers to which it assign tasks. It communicates with each controller and is tasked with making sure that the system view is consistent across the second layer.
- Prioritizer/ Deprioritizer/ Blocker/ Remote Controller/ Application Monitor/ Data Packet Monitor: These are all controllers that have been assigned a specific task by the Delegator. There may be more than one of each of these, depending on what the administrative needs of the network may be. Should one of these controllers

become compromised, they can be replaced by one of the controllers in the waiting queue until such time as they can be made available again.
- Security Monitor: The primary task of this controller is to monitor the system as a whole. There may be multiple security controllers to monitor other controllers, the Delegator or the network as deemed necessary. These monitors would be responsible for handling a disabled controller as well as monitoring upper level applications that are security specific.

There are other roles that the controllers may have to complete, and as the SDN architecture continues to be developed those roles may be further redefined as needed.

As to the security issues that are present in today's SDN paradigm, we believe that this new hierarchy concept would be able to improve the handling of issues such as direct attacks on the controller, as well as attacks on portions of the network. This is because the system will be able to more dynamically allocate its resources and more quickly identify and respond to threats due to the removal of the long queue waiting problem.

Furthermore, because a security specific subset of controllers is present, a more robust security framework can be introduced to allow the network to become better insulated from attacks. Finally, because we now have controllers whose specific task is to monitor the system, intrusion detection and prevention mechanisms can work much more quickly to intercept potential threats. As the concepts of machine learning and fuzzy logic are introduced into this paradigm it is possible to have a system that handles almost all threats on its own.

This architecture would also help deal with load-balancing issues by removing the bottleneck in communication with a controller. By distributing the controller's roles, we increase the speed of service to data packets that may otherwise have been stuck in a wait-queue while a single controller was handling higher priority issues. This architecture also improves monitoring, routing, and traffic flow in the network by splitting those roles more efficiently among these highly flexible controllers.

Our proposed architectures can be further enhanced by the following additions:

- Securing communication among controllers: A trusted connection must be established in controller-to-controller communication by creating strong protocols and encryptions, among other things.
- Defining controller interaction: How these controllers interact must be defined so that they may work together effectively.
- Designing generic controllers: By delegating tasks to a set of generic controllers, we can ensure that they perform their tasks correctly and in conjunction with the other controllers in the network.
- Defining DDoS policies: How the system reacts when a controller is compromised must be precisely defined.

VII. Conclusions

With the introduction of OpenFlow, OpenStack and OpenDaylight, we are now able to cater to multi-vendor and multi-network in a programmable environment. However, SDN is still in its developmental stage, with security and load balancing representing some of the key open challenges. The three proposed role-based, multiple-controller architectures address some of these key issues, allowing SDNs to become more sustainable, and have improved availability, policy enforcement, mitigation, management, and overall security. It is also worth mentioning that our proposed role based multiple controller approach can also be applied in other areas to provide better consumer service in a cloud network, improve resource provisioning, and provide vital information for charge-back in the network.

References

[1] D. Kreutz, F. M. V. Ramos, and P. Verissimo, Eds., *Towards Secure and Dependable Software-Defined Networks HotSDN'13*. Hong Kong: the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Jun 2013.

[2] "Onf solution brief." [Online]. Available: www.opennetworking.org

[3] V. Yazici, M. Oguz Sunay, and A. O. Ercan, "Controlling a Software-Defined Network via Distributed Controllers," *ArXiv e-prints*, Jan. 2014.

[4] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed sdn controller," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 7–12, Aug. 2013.

[5] S. Schmid and J. Suomela, "Exploiting locality in distributed sdn control," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 121–126.

[6] T. Koponen, M. Casado, J. S. Natasha Gude, L. Poutievski, R. R. Min Zhu, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: a distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX conference on Operating systems design and implementation (OSDI'10)*, Vancouver, BC, Canada, 2010, pp. 1–6.

[7] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, pp. 2181–2206, Fourthquarter 2014.

[8] K. Dhamecha and B. Trivedi, "Sdn issues â a survey," *International Journal of Computer Applications*, vol. 73, no. 18, pp. 30–35, 2013.

[9] L. Zhang, G. Shou, Y. Hu, and Z. Guo, "Deployment of intrusion prevention system based on software defined networking," in *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, Nov 2013, pp. 26–31.

[10] C. Chaudet and Y. Haddad, "Wireless software defined networks: Challenges and opportunities," in *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, Oct 2013, pp. 1–5.

[11] S. Dotcenko, A. Vladyko, and I. Letenko, "A fuzzy logic-based information security management for software-defined networks," in *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, Feb 2014, pp. 167–171.

[12] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "Fresco: Modular composable security services for software-defined networks," *Internet Society NDSS*, pp. 1–16, Feb 2013.

[13] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 151–152.

[14] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "Sdn security: A survey," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, Nov 2013, pp. 1–7.

[15] X. Wen, Y. Chen, C. Hu, C. Shi, and Y. Wang, "Towards a secure controller platform for openflow applications," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 171–172.