Multinetwork Control Using OpenFlow

Natalia C. Fernandes, Diogo M. F. Mattos, Callebe T. Gomes, Laura G. Panzariello,
Victor T. Costa, Marcelo D. D. Moreira, Igor M. Moraes, Miguel Elias M. Campista,
Luís Henrique M. K. Costa, and Otto Carlos M. B. Duarte
GTA/COPPE - Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brasil

***Abstract*—In this paper, we propose to use OpenFlow as a platform for creating a virtual network environment. We analyze the advantages and disadvantages of using OpenFlow and develop a prototype to manage virtual networks. Our prototype allows network visualization, network monitoring, and flow control. Our analysis shows that the developed prototype can be used as the basis for creating a pluralist network infrastructure with high performance, easy management, and high security.**

I. Introduction

Although the Internet is a great success, it cannot handle new requisites such as end-to-end quality of service, security, and mobility. Hence, new approaches are being proposed and many researchers believe that the new Internet architecture will be based in a pluralist model [1]. This model states that the network must support multiple protocol stacks simultaneously. Hence, specialized networks could be created to solve specific problems, while the current Internet would be used for general purpose applications.

This paper addresses the issue of using OpenFlow as the platform for creating a pluralist network. We developed a prototype that provides the basic functions for managing virtual networks on a pluralist network based on OpenFlow.

II. The OpenFlow Pluralist Prototype

OpenFlow [2] is a Stanford initiative for using the wiring closets on university campus for the production network and also for experimental networks. It is based on a model that provides a physical separation of the network forwarding function and the network control function. The forwarding elements are dummy elements that have a generic shared forwarding table. They only forward the packets based on pre-established rules in their forwarding tables. The control of each virtual network is centralized on the element called network controller, which can access and configure all the forwarding elements. The main advantage of OpenFlow for creating a pluralist environment is its shared forwarding table. OpenFlow is based on the idea that each flow is described not only by IP addresses, but also by all header fields in the packet. Every time the first packet of a new flow reaches a forwarding element, it is forwarded to the controller. Then, the controller sets a path for the following packets of that flow in the chosen forwarding elements. Also, the controller can order the forwarding elements to modify header fields before forwarding the packets. This makes OpenFlow a flexible infrastructure that can be used for creating a pluralist platform.

We developed our prototype based on Nox [3], which is an OpenFlow controller. First, we selected the functions that should be available when managing virtual networks that share the same physical medium. Then, we developed Nox modules to create a platform for virtual network management via web service. Among the available functions, we visualize the network topology and monitor all the operations running on the network, such as the throughput of a specific flow or the throughput of a virtual network. Besides creating/deleting flow rules, our platform can move the flows from one forwarding element to another on real time, without packet losses.

We analyze our prototype in a testbed with personal computers and also with wireless routers running OpenFlow. The results show that our implementation is stable and can be used to manage the network with low complexity and low delays. Indeed, we believe that this prototype can be used for real-time control of the network.

III. Conclusion and Future Work

Our analysis shows that OpenFlow is a promising tool for creating a new platform for a pluralist network, not only because it allows multiple networks to run in parallel, but also because it virtualizes the network without a significant impact in the performance. Our prototype already presents the main functions needed for piloting virtual networks, because it can sense and change all the parameters of the network. Moreover, our prototype allows easy and flexible management of the virtual network infrastructure. The next steps will be the creation of intelligent mechanisms to control our prototype, which, up to the moment, is manually piloted. Also, we will invest on mechanisms to guarantee the virtual network isolation and minimal quality of service requirements for each virtual network. We also intend to develop new solutions for creating domains in OpenFlow networks, so that it can be used in large scale networks.

Acknowledgments

This work was supported by FINEP, FUNTTEL, CNPq, CAPES, and FAPERJ.

References

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *IEEE Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[3] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: Towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, July 2008.