Developing NOX Applications for Network Control

Diogo M. F. Mattos, Callebe T. Gomes,
Natalia C. Fernandes and Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação - PEE/COPPE - DEL/POLI
Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brazil
Email:{menezes,callebe,natalia,otto}@gta.ufrj.br

***Abstract*—Network control and management is one of the greatest challenges of the future Internet and, in a pluralist approach, virtual networks must be individually controlled over the shared physical network substrate. This work presents a framework for NOX applications development to control Openflow switched networks. The framework is based on the concept of web server and allows system interoperability with multiple end-user interfaces.**

I. Introduction

Network virtualization technology enables a pluralist network approach, sharing multiple logical networks over a same physical substrate. This approach is flexible because several networks can run, at the same time, and offer different services. One network virtualization technology is Openflow [2] switching that is based on defining sets of forwarding flows. OpenFlow differs from conventional IP switching technology because we can program its flow forwarding table. A flow, in Openflow context, is a set of ten parameters contained in the header of packets forwarded by the switches.

When a packet arrives in an Openflow switch, the switch checks if the packet matches one flow defined in the forwarding table. If it does, the packet is then forwarded for its destination. Otherwise, the packet is sent to a controller, which is a computer program that decides which action to be taken by the switch. The main controller for the Openflow technology is Nox [1].

Nox controller is a software layer that implements an interface between Openflows switches and applications. These applications are responsible for determining which action must be taken by a switch when a new flow is to be instantiated. This work proposes a software architecture composed by six Nox applications. These applications intend to be used for managing and controlling virtual networks over an Openflow network.

II. Architecture

The main idea of the proposed architecture is to implement a virtual network over a switched physical network, using Openflow as switching technology and Nox as network controller. The network controlling role will be performed by Nox applications. The proposed software architecture is based on the concept of web service. One of the developed applications is a web server that communicates with other applications. The web server publishes functions provided by these applications. As these management functions are published as a web service, they can either be accessed by an automated tool for network controlling and management or by another tool which provides an human interface to interact with the network.

III. Nox Applications

The main application is the web server application. Its function is to publish services provided by the controller, which allow a network administrator to perform a maintenance action over the network. The other applications are the implementation of the services provided by the web server application. The services implemented are network topology discovery, statistics collection and aggregation, flows management and flows migration. It also has a Spanning Tree application, which does not provide a service, but allows the existence of loops in the network.

The applications can be divided into two main sets: monitoring applications and acting applications. The first set is responsible for picking up information about the network, such as topology, flow, switch port, table, byte count and spanning tree information. All these pieces of information are published as a XML (Extensible Mark-Up Language) in a web service. The second set is composed by the applications responsible for adding, deleting and migrating flows. These applications are called by a URL sent to the web service with its parameters.

IV. Conclusion

Openflow and Nox are two complementary technologies that claim to be a good option to virtualized networks. However, their development still in progress, and there are some missing functionalities. Therefore, this work intended to develop some applications to be deployed on Nox in order to overcome network management issues. The applications were developed and were tested over a testbed network. The tools developed work fine and are ready to be used. The next steps are develop new functionalities and execute some tests in order to analyzes the system overhead, caused by monitoring tools, and performance.

Acknowledgment

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

References

[1] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: Towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, july 2008.

[2] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, April 2008.