Context-Based Management for Virtual Networks

Igor M. Moraes, Marcelo D. D. Moreira, Carlo Fragni, Diogo M. F. Mattos, Leonardo G. Valverde,
Miguel Elias M. Campista, Luís Henrique M. K. Costa, and Otto Carlos M. B. Duarte
Universidade Federal do Rio de Janeiro, GTA/COPPE/Poli - Rio de Janeiro, Brazil
Email:{igor,marcelo,carlo,menezes,valverde,miguel,luish,otto}@gta.ufrj.br

***Abstract*—In this paper, we present the information provided by Xen and OpenFlow that can be used as the context of virtual networks. In addition, we discuss the actions that can be taken based on the context monitored. The context information and these actions can be used to develop reasoning mechanisms to manage the resources allocated for the multiple virtual networks.**

I. Introduction

The current Internet architecture allows only one protocol stack running over the physical substrate. Thus, packets from all applications are forwarded according to a single model. In order to provide customized services for the different applications in the future Internet, researches have proposed a pluralist architecture, which allows different protocol stacks running simultaneously over the same shared physical substrate [1]. Most of the pluralist architectures proposed are based on the concept of network virtualization. The idea is to run multiple virtual networks with the substrate providing separate resources for each network [2]. Thus, the efficient sharing of the underlying network resources is a fundamental challenge. In Horizon project, we have introduced a piloting system to deal with the resource-allocation problem among the different virtual networks. This system is context aware, thus, we have to first monitoring the environment to acquire context information surrounding a network element, reasoning about the context acquired and then act to achieve a goal. In this paper, we focus on the context information we can acquire from physical and virtual networks and also on the actions we can take to pilot these networks. We also present the prototypes of virtual network elements developed in Horizon project, discuss context information utilization, and we also point out our future directions.

II. Monitoring and Acting on Virtual Networks

Currently, we have developed two testbeds to evaluate virtual networking. The first one uses Xen to implement virtual routers and the second one are developed with OpenFlow switches. Both consider centralized network management by using a controller node. This node is now acting as the piloting system: it receives and stores all the context information acquired by sensors but the decisions and actions are taken by the network administrator. The *sensors* are programs running on each physical network element, routers and switches, and also on each virtual element, virtual machines running on a physical router. Basically, the sensors acquire context by using measurement tools and then process and represent the information using XML (eXtensible Markup Language). Thus, when the controller needs the context, it sends a request to the sensors.

With Xen, we can acquire context information about computational resources and network states. With `xentop`, for each virtual router, sensors acquire the CPU usage, the memory usage, the number of CPUs allocated, and the number of virtual network interfaces for each virtual router. We have also estimated the usage of physical CPUs summing the usage of virtual CPUs. The topologies of physical and virtual networks are discovered with `nmap`. Thus, a node knows its neighbors, physical and virtual, and can determine the RTT to them by using `ping`. Finally, with `ifconfig`, sensors acquire the number of packets/amount of bytes transmitted and received by each physical and virtual interfaces and thus we can calculate the transmission and reception rates. With OpenFlow, we can obtain context information related to switch ports and flows. For both, sensors acquire the number of packets/amount of bytes transmitted and received. For a flow, we can also determine its duration, priority, and idle time. We can also estimate the CPU usage by counting the number of messages exchanging between the controller and the switches and the memory usage based on the size of flow tables.

Xen and OpenFlow allow to *instantiate/delete* and also to *migrate* network elements and flows and *set* its resource-allocation parameters. These are the basic actions we are proposing for the piloting plane. For example, traffic engineering can use the migrate primitive to move virtual network elements/flows along the physical infrastructure to minimize energy costs or other objective functions.

III. Future Directions

We are now defining the techniques used to represent the context acquired and defining reasoning mechanisms to enable the piloting system to act autonomously. The alternatives are to develop a knowledge base by using facts and rules and ontology and also to use contextual graphs. The reasoning technique must be chosen by taking into account its complexity and, consequently, its time to react to network changes.

Acknowledgment

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

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] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, and M. Chiang, “DaVinci: dynamically adaptive virtual networks for a customized Internet,” in *ACM CoNEXT Conference*, Dec. 2008.