

An Interactive Visualization Framework for Next Generation Networks

Sriram Natarajan
University of Massachusetts, Amherst
snataraj@ecs.umass.edu

Xin Huang
Deutsche Telekom Inc R&D Lab USA
xin.huang@telekom.com

ABSTRACT

Next-Generation Internet applications demand inherent flexibility and programmability from the underlying network. OpenFlow-based substrate, as a potential solution, provides the required support to enable network innovations by separating the intelligent control plane from the datapath processing. Several visualization systems have been designed in the context of OpenFlow to facilitate network management. However, the fundamental drawback of these systems is the lack of generality. In this paper we propose ROVIZ, an interactive visualization framework for next generation networks. The contributions of our work consists of the following: i) compatible with any OpenFlow based networks, irrespective of the controller(s), ii) maintains real-time information about the entire network as well as individual devices, and iii) provides interactive menus for selective information retrieval improving bandwidth constraints.

Categories and Subject Descriptors

H.4 [Future Internet Architecture]: Network Visualization

Keywords

Network Visualization, Network Management, Next Generation Networks, OpenFlow

1. INTRODUCTION

Next-Generation Internet designs envision inherent flexibility and programmability in the underlying network infrastructure. OpenFlow [1] provides a potential solution to enable network innovations based on commercial switching hardware(s) by separating the intelligent control plane from the datapath processing. OpenFlow-

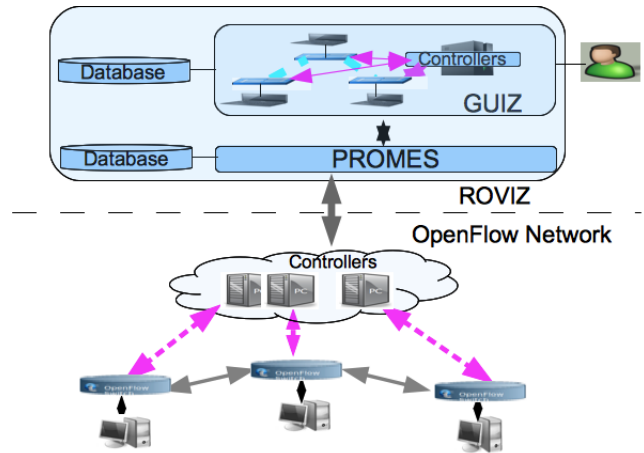


Figure 1: ROVIZ: Network Visualization Framework

based networks usually consist of two parts: switching hardware(s) and controller(s). The mapping between these two could be either one-to-one or many-to-one. OpenFlow refers to the communication protocol between controllers and switches.

Network visualization provides extensive useful information about the underlying network, e.g., network topology, traffic statistics, network and device configuration parameters etc. Gathering real time statistics about the network state through visualization can help us perform network management decisions and monitor network related problems. Several visualization systems have been designed in the context of OpenFlow, however, such systems are constrained by their fundamental limitations. For example, LAVI [3] and NOX GUI [4] have restricted towards visualizing the network that are connected to a NOX controller. Both systems lack generality to work with other controllers. ENVI [2], the frontend unit, communicates with LAVI by requiring an intermediate binary to JSON format translator. Such intermediate levels of abstraction requires significant development time and making it hard

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM CoNEXT Student Workshop, November 30, Philadelphia, USA.

Copyright 2010 ACM 978-1-4503-0468-9/10/11 ...\$10.00.

to port when extended to work with other OpenFlow based controllers.

We propose a generic framework that enables the visualization unit to communicate using OpenFlow protocol. This design choice requires the controller to just relay OpenFlow messages and introduces generality by being able to connect to any OpenFlow based controller. In the next section we discuss our ROVIZ system architecture. Section 3, elaborates our implementation details. We conclude this abstract with our extension details and future work.

2. SYSTEM ARCHITECTURE

As show in Figure 1, ROVIZ communicates with the underlying OpenFlow network through OpenFlow controllers. It establishes a direct TCP connection with each OpenFlow controller and exchanges OpenFlow protocol messages. ROVIZ provides an easy way to visualize the network, using a two-tier architecture. PROMES (message processor) gathers network information from the OpenFlow controllers and displays them using GUIZ (frontend user interface). Both PROMES and GUIZ maintains lightweight, dynamic database which records and updates the network information in real time.

PROMES receives messages from the controllers, with a unique socket connection associated with every controller. PROMES talks OpenFlow protocol and processes the received information based on the type of message as per the specification. Individual functional units namely NODIZ/FLOWIZ/STATIZ handle node/flow/statistics information in parallel. PROMES also handles queries from GUIZ. It converts the queries into OpenFlow requests and sends them to the corresponding controllers(switches).

GUIZ receives the updated data by establishing an active TCP connection with PROMES. GUIZ uses a message analyzer to determine the type of message received and forwards the data to a decoder to understand the message, before updating the visual unit using a layout manager. GUIZ also provides a real time interactive user interface to update and talk to the OpenFlow-based devices through PROMES. This interactive information retrieval feature makes our system unique by providing a good balance between gathering required information and improving bandwidth constraints, thus increasing scalability.

3. IMPLEMENTATION

To evaluate our system design, we developed ROVIZ prototype and tested on different OpenFlow switches, e.g., Software implementation, HP ProCurve 5400, and NEC IP8800. ROVIZ worked efficiently with both NOX and non-NOX based controllers, gathered topology information, real time statistics, and enabled us to control the switches with our visual unit. The prototype was

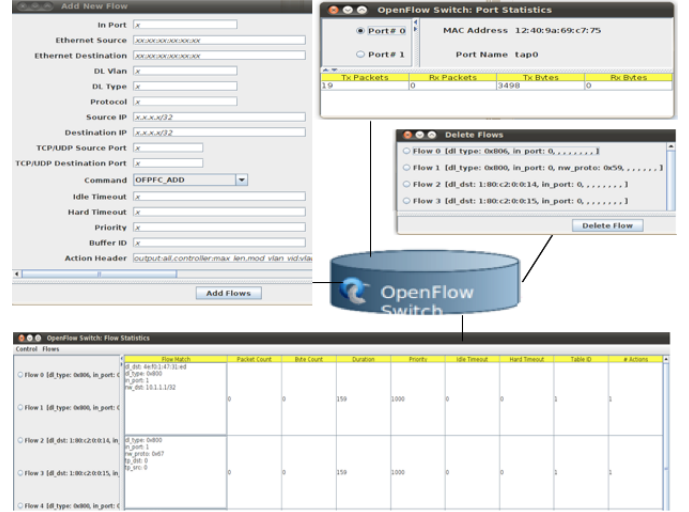


Figure 2: GUIZ: Interactive Visual Menu

found to be very useful for debugging the network and determine the faulty units.

Figure 2 shows menus that interact with the underlying OpenFlow switches in our implementation. Flow and Port Statistics in the statistics menu shows accurate real time information. A new flow could be added to the given switch using Add Flow menu by filling out the required fields. Delete Flow menu shows list of current active flows, and by selecting one, we deleted the corresponding flow from the switch.

4. FUTURE WORK

In our current prototype implementation we have focused our work to prove the ROVIZ architecture design and demonstrate how it facilitates network management and debugging with OpenFlow based networks. In the future, we plan to extend the current framework to support i) non-OpenFlow devices, ii) efficient topology discovery.

5. REFERENCES

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks., April 2008.
- [2] D. G. Underhill. An extensible network visualization and control framework. Thesis, Stanford University, May 2009.
- [3] K. K. Yap. Network visualization. <http://www.openflowswitch.org/wk/index.php/LAVI>, May 2009.
- [4] K. Zarifis. Nox gui. <http://noxrepo.org/noxwiki/index.php/NOX-GUI>, October 2010.