Data-monitoring Visualizer for Software Defined Networks

Luz Angela Aristizábal
Dept. Informatics and Computation
National University of Colombia
Manizales, Colombia
e-mail: laaristizabalq@unal.edu.co

Nicolás Toro.
Dept. Electrical, Electronics and Computation
National University of Colombia
Manizales, Colombia
e-mail: ntorog@unal.edu.co

***Abstract*— Monitoring the behavior of a data network is a starting point for its analysis, and must be a constant activity that allows operators and administrators to quickly notice changes in the network. A view of the topology associated with the network traffic could speed the response to possible network failures. The study's principal contribution involves using graph signal processing theory as method to structure a signal composed of statistical data provided by the software defined network switches and establishing correspondence between the statistical traffic patterns with color.**

Keywords- Monitoring; Graph Signal Processing, Software Defined Network.

I. INTRODUCTION

The network monitors usually use lines, bars, pie charts, and area charts to show network traffic [1]. The analysis of this information takes time. It is necessary to reduce the time invested in the information monitoring analysis, by creating methods that allow for visualization of the traffic information correlation with network topology.

The goal of this investigation is to implement a mechanism that facilitates the visual detection of congestion for network managers. The proposed strategy makes use of two current technologies: Graph Signal Processing (GSP) and Software Defined Networks (SDN).

GSP is a new area of study in Digital Signal Processing that provides us with conceptual and practical tools to model complex networks and graphically show their evolution. The advantage of applying GSP in data network analysis is the possibility to relate network topology with its behavior throughout time. [2]. This form of network behavior visualization allows for timely detection of congestion levels and abrupt changes that can be consequences of failures.

With the emergence of SDN in 2008, a new prospect for the implementation of network monitors was visualized. In its operation model, each switch connected to a controller includes the generation of statistics associated with data flows circulating through its ports. This makes it possible to obtain the switches activity statistics, taken at regular intervals, which allows us to generate signals that feed the network graph model [3][4].

The principal contribution of this investigation is to show how SDN activity can be modeled with GSP theory and how one may implement an application that takes advantage of the statistical information that an SDN's nodes calculate in run-time, in order to construct a signal that characterizes network traffic.

This paper is divided in three sections: Section 1 describes the conceptual aspects of GSP and SDN involved in the development of this study. Section 2 presents the method utilized in order to graphically show the network topology associated with traffic information taken from the statistics sent by SND switches to the controller. Section 3 specifies the results analysis, and finally, the conclusions.

II. SOFTWARE DEFINED NETWORK AND GRAPH SIGNAL PROCESSING

This section will present the relevant concepts involve in the application development.

A. Software defined networking

SDN consists of three element types: switches, controllers, and a secure communication channel that communicates the controller with the switches. Communication between devices uses Openflow messages [5][6], as is shown in Figure 1.

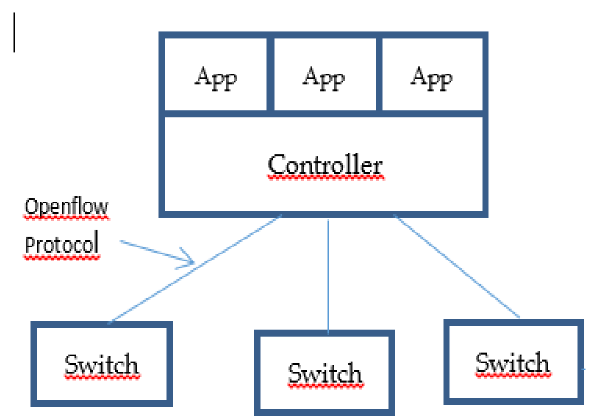

Figure 1. Structure of a Software Defined Network.

The controller periodically sends the S*tatistic request* message to the switch, which answers with the message *Statistics reply* that includes: transmitted and received bytes, transmitted and received packets, and transmitted and received errors and collisions.

With this information, it is possible to form the data sequences that will constitute the network traffic signal in a specific time interval.

 ISBN: 978-1-61208-655-2

B. Graph Signal Processing (GSP)

A data network can be represented by a graph: **G**=(V,A), where V is the set of nodes, $V=\{v_0,v_1,\ldots.v_{N-1}\}$ and A is the adjacency matrix, N number of nodes . Each v_i is a node that is connected with a v_j node.. A(i,j) determines the existence of a directed edge from v_i to v_j.

For an SDN, V is the set formed by network devices, such as: switch, server, and host. A(i,j) represents a connection between the nodes or devices *i* and *j* . See Figure 2.

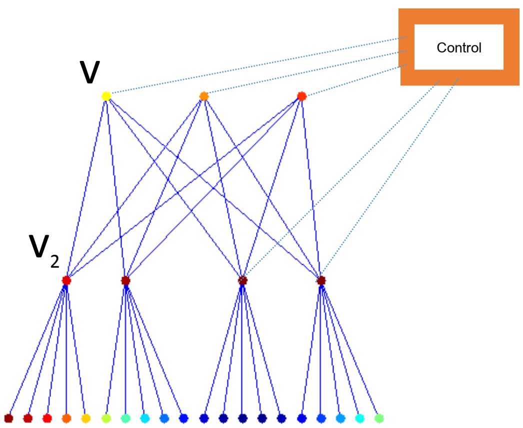

Figure 2. Data network graph.

In this case, A(1,2) has a value of one, indicating that there is a connection that relates node v_1 with node v_2.

III. DATA-MONITORING VISUALIZER

In this section, the actions taken to achieve the following two objectives will be explained:

- Associate the results of a monitoring process with network topology.
- Generate a graph structure that allows for visualization of the changes in network traffic during a time interval.

Initially, the statistics sent by the SDN switches to the controller were taken as response to the "*Statistic request"* message. With this information, a structure was formed, which contained those bytes transmitted and received by each of the devices connected to the network switches. The aim was to form an ***S*** signal with *T* length, which contained the information received by the controller regarding various time instances (1).

$$S_j = \{s_1, s_2, \ldots\ldots s_T\}$$
$$1 <= t <= T$$
$$1 <= j <= N \quad (1)$$

Each s_t represents the transmitted bytes in a *t* time for device v_j (N is the number of network devices). The entire network visualization process is generated with value s_i for each S_j. This is:

$$MV_t = \{G(V, NS_t, A)\}$$
$$0 <= t <= T \quad (2)$$

Where MV_t is the visualization of the entire network at time t (2), V is the set of network devices, NS_t is the signal formed by the bytes transmitted from all network devices in instant t, and A is the adjacent matrix. The algorithm implemented is shown in Figure 3.

1. From the topology of the network, generate graph G(V,A)
2. Read the statistics from the software defined network during time T.
3. Associate each device or network node with the signal Sj
4. For t=0:Δt:T
   for j=1:N
   NS(t,j)=V(j).s(t)
   End
   MV(t)=G(V,NS(t),A)
   end

Figure 3. Algorithm for graph generation with traffic signal.

And finally, how does color correspond to network traffic states? What color represents a congested node? What color represents a node with low traffic? Color relationships are shown in Figure 4.

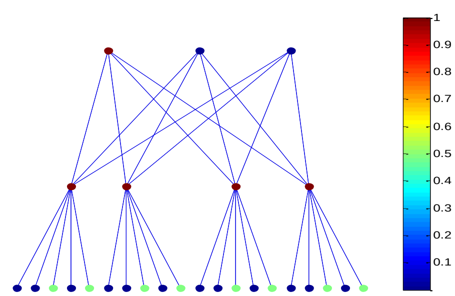

Figure 4. Correspondence between network traffic and color.

Red indicates that the node is congested, blue indicates that the node has low traffic. Thus, the lower nodes, that in the figure are green, have a moderate level of traffic.

IV. RESULTS

How was the monitoring result associated with network topology? How was it made graphically tangible? It was necessary to implement three processes:

- SDN was implemented in mininet, using a Ryu controller. The topology implemented is shown in Figure 4, in which the two first levels are openflow switches and the third level contained hosts. In order to illustrate the advantages of the proposal described in this document, some hosts don't have information flow, and others use *lperf.* Figure 4 shows the effects of these traffic differences. The inactive hosts are blue, and the *lperf* hosts are green.

- The statistical data was taken each 30 ms. and saved in a file. With this information the traffic signal associated with the topology was created This signal determines the color for each node in the visualization.
- The network topology was created using the matlab GSP toolbox. The statistics file was taken, and the Figure 3 algorithm was implemented. Its execution allowed viewing of color changes associated with network traffic. Figure 5 shows some of the graphs generated by the application at different moments in time.

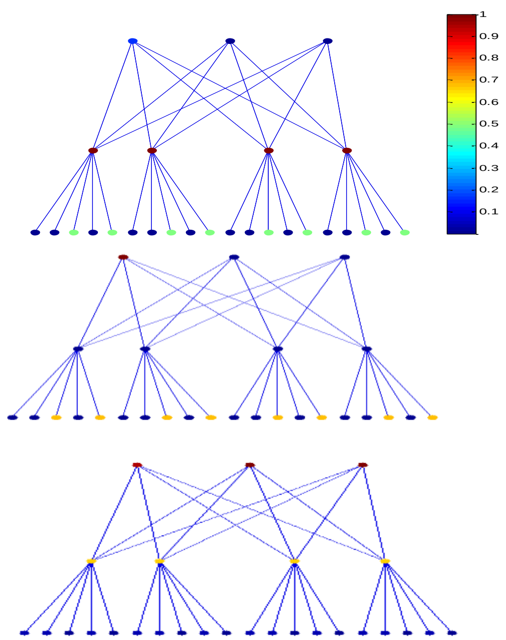

Figure 5. Three SDN colored topology frames.

In the superior graph in Figure 5,observe that the hosts are green. This indicates greater activity than those hosts which are blue, and the distribution switches have greater activity than the core switches. In the graph below, the activity is shifted from the lower nodes, the computers, to the distribution switches, and from these to the core switches.

V. CONCLUSIONS

The network must be analyzed as a dynamic system, with monitoring applications that consider time to be an essential element associate of the entire network.

The software implemented allows for visualization of the network dynamics, in animation form, which shows the way in which it changes through time. This permitted agile detection of areas or nodes with atypical behavior. The most common ways to present monitoring information, using static graphics (line, bar, pie charts, area charts, etc.), requires considerable time investment on the part of the network operator to analyze these graphics, especially when it is necessary to find the cause of a failure. With such an application, a congestion failure can be detected by simply observing the animation.

This implementation motivates to continue the investigation, creating mechanisms that automatically analyze the network graphic signal and the implementation of algorithms for traffic distribution from SDN congested areas to nearby areas less congested.

REFERENCES

[1] SevOne "Software Defined Network Monitoring" [Online] Available from https://www.sevone.com/solutions/sdn_old. [accessed: 2018-02-20]

[2] D Shumman, S. Narang, P.Frossard, A. Ortega, and P. Vanderghesynst, "The Emerging Field of Signal Processing On Graphs: Extending High- Dimensional Data Analysis to Networks and Other Irregular Domains". IEEE Signal processing Magazine, May 2013, pp 83 – 98, doi: 10.1109/MSP.2012.2235192, ISSN: 1053-5888

[3] D. Luong, A. Outtagars, and A. Hebbar, "Traffic Monotoring in Software Defined Netwoks Using Opendaylight Controller" Lecture Notes in Computer Science. LNCS, Vol. 10026. Springer. Dec 2016, pp 38-48, doi: doi.org/10.1007/978-3-319-50463. ISBN: 978-3-319-50463-6

[4] L. Niels, M. van Adrichem, C. Doer, and F. Kuiper."OpenNetMon: Network Monitoring in Openflow Software-Defined Networks". Network Archictures and Services. Delft University of Technology. IEEE Xplore. July. 2014.doi: 10.1109/NOMS.2014.6838228, ISBN: 978-1-4799-0913-1

[5] M, Jammal, T. Singh, A. Shami, R.l Asal, and Y. Li, "Software defined networking: State of the art and research challenges," Computer Networks, vol. 72, Oct 2014, pp 74-98, doi:10.1016/j.comnet.2014.07.004

[6] OpenFundation. S. Bailey, Deepak Bansal, Linda Dunbar, Dave Hood, and Zoltán Lajos Kis, "SDN Architecture Overview", [Online] Available from:
https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf. [accessed: 2018-03-05]

[7] A. Sandryhaila and F. Moura, "Discret Signal Processing: on Graph". IEEE Transacion signal processing. Vol 61. No 7. April. 2013, pp 1644 -1656, doi: 10.1109/TSP.2013.2238935.

[8] N. Le Magoarou, J. Paratte, D. Shuman, V. Kalofolias, P. Vanderghesynst, and D. Hammond, "GSPBOX: A toolbox for Signal Processing on Graphs", ArXiv e-prints. Mar 2016, [Online] Available from: https://arxiv.org/abs/1408.5781v2

[9] A. Sandryhaila and F. Moura, "Discret Signal Processing: on Graph" Proceedings of the Sixth International Conference on Advances in Computer-Human Interactions (ACHI 2013) IARIA, Feb. 2013. Nice France, pp. 7-12, ISSN: 2308-4138, ISBN: 978-1-61208-250-9

 ISBN: 978-1-61208-655-2