

Babko Dmytro, Kulakov Yuriy.

TRAFFIC ENGINEERING IN SOFTWARE-DEFINED NETWORKS USING BIG DATA TECHNOLOGY

This article addresses the issue of traffic engineering in software-defined networks (SDN) using Big Data technology. A method using MapReduce technology to analyze and evaluate the traffic matrix (TM) is considered. This approach involves controlling the statistics of the ports of the OpenFlow (OF) switches. With this method, all traffic crossing the path between the origin point and the destination point (OD) is used to estimate the traffic matrix.

Key words: traffic engineering, SDN, Big Data, MapReduce, traffic matrix.

Relevance of the research topic. Traffic analysis is an important process in SDN. The collection of real-time traffic statistics plays a significant role in network monitoring for early detection of channel and route congestion. The need to process large amounts of data in real time is a current problem for the development and implementation of modern traffic monitoring and analysis systems.

Target setting. A common problem faced by traffic engineering systems was the generalization of the collected statistics. Using Big Data methods, large amounts of data can be easily processed, and statistics can be provided almost in real time. The identity of the type of data aggregation makes it possible to use Big Data methods for traffic analysis, with a difference only in the amount of data required.

Analysis of recent research and publications. In recent years, the number of traffic engineering work in SDN has increased significantly. One research introduced OpenTM, a traffic matrix analyzer for OF networks. OpenTM was implemented as an add-on to the controller, based on a periodic survey of switches [1]. Another approach was based on the aggregate SDN streams used to estimate the traffic matrix. Aggregation rules are divided into groups to increase the accuracy of the estimate [2].

Recent studies have evaluated the traffic matrix based on modeling and optimization procedures. Mathematical models for estimating TM based on sampling mechanisms were used for modeling.

Defining the unexplored parts of the general problem. The combination of existing solutions based on the use of mathematical modeling to generate the traffic matrix and Big Data paradigms, such as MapReduce or

lambda architecture, covers much of the task of monitoring and analyzing traffic with the solution of processing large amounts of data in real time.

The research objective. The task is to investigate and upgrade the algorithm for engineering traffic in SDN using MapReduce methodology.

Statement of the main material. In this article, a method of constructing traffic based on the MapReduce paradigm of Big Data technology are considered.

The OpenFlow switch stores statistics on ports. According to the OpenFlow protocol specification, the counters are updated with each packet passing through the switch [3]. The Big Data monitoring method periodically requests the SDN controller to obtain counter values from the ports.

Generally, in the process of traffic engineering with Big Data there are three main stages:

1. Data collection. Loads service data and traffic information using the SDN controller.
2. Data aggregation. Processing of the received data, calculation of network parameters almost in real time.
3. Data accumulation. Stores and provides service data and parameters to traffic analysis systems.

When data is collecting, the following entities are using: Host, Interface, Switch Port, Switch, Flow Table, and Flow. Based on the peculiarities of OF, the counters are updated with each new package, so the current data are different from those previously collected, so to implement system monitoring, the data must be updated periodically [4].

The data aggregation process receives the data collected through data collection activities, performs processing, generates the expected statistics and sends the result for storage. Receiving data includes the following steps: receiving a data collection message, saving the message data, sending the saved messages as requests for further processing.

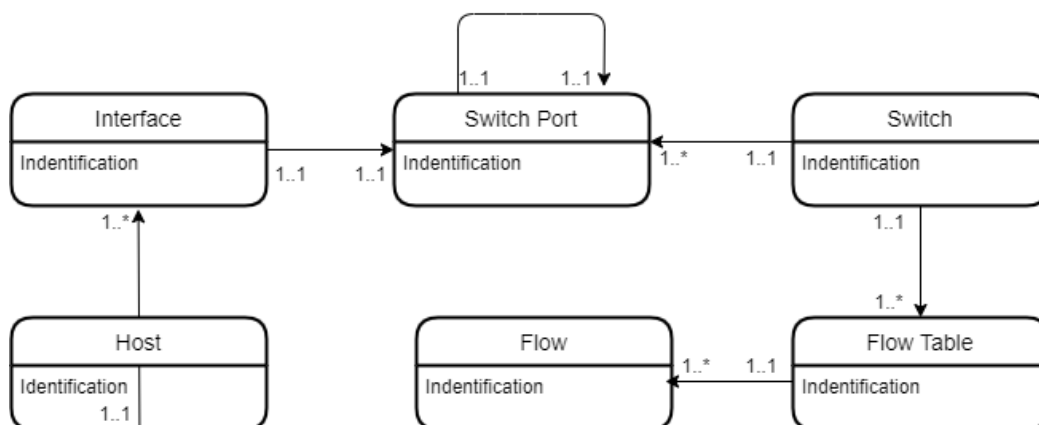


Fig. 1. ER-model for inventory data

In the process of data accumulation, the collected data and created statistics are stored. This data can be used by a specific system for monitoring, traffic analysis or for visualization.

The introduction of the Big Data monitoring method has provided a significant amount of statistics that reflect network traffic. One of the main parameters of statistical data is the traffic matrix. Consider using MapReduce technology to evaluate it.

This approach means that the keys will consist of all parts of the routes between pairs of origin host - destination host. The following values are used to generate the LP set of all paths passing through a certain port:

- N is the set of all network points;
- H is the set of all hosts;
- S is all network switches;
- U1 is the set of host switch connections;
- U2 is the set of ' switch-switch connections;
- U - union of U1 and U2.

The network is represented by a graph G defined by (N, U) on Figure 2.

• L is the set of channels between the host and the switch and the switch-switch channels;

- HC is the set of the origin-destination pair combination;
- P is the host origin-host destination path defined by the connection sequence.

As a result, we get an LP network - all possible paths passing through a particular port [5].

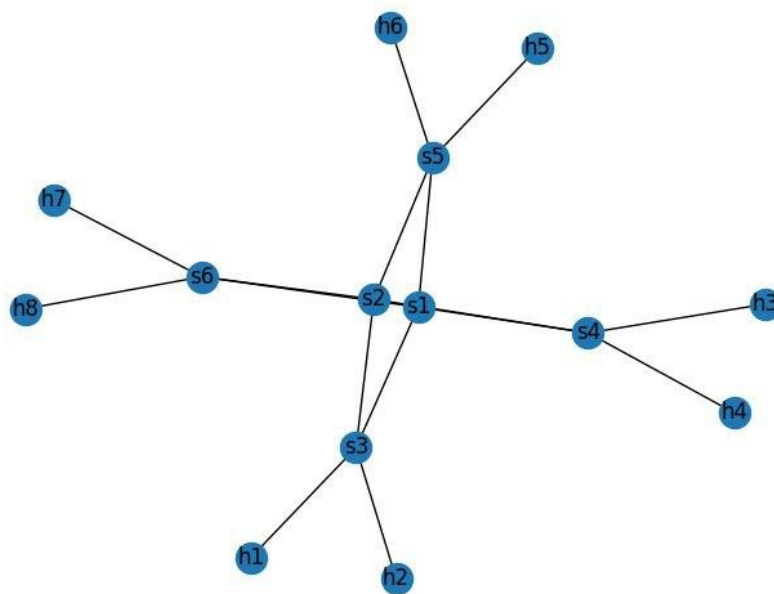


Fig. 2. Simulated network graph

Statistical analysis of these values can be described by the following:

$$Thr_i(t) = \frac{(BR+BT)}{S}, \quad (1)$$

where $Thr_i(t)$ is the accumulated bandwidth of the channel i and at time t , S is the time of connection activity.

According to the obtained set LP it is necessary to generate key / value pairs. The key in this case is the path and timestamp, and the value contains the calculated current and accumulated bandwidth.

The process of traffic analysis can be described by performing a sequence of actions: data flow collection, calculation of switch port statistics, generation of switch statistics, generation of network switch statistics, generation of network port statistics; switch traffic analysis generation, port traffic analysis generation, connection traffic analysis generation, path traffic analysis generation.

The *packet received* (PR) and *packet transmitted* (PT) fields are also added to the above counters to calculate switch port statistics [3]. In addition to the counters used, it is also proposed to collect statistics in the fields of *frame error received* (FR), *collision count* (CC) and *CRC error received* (CRC). OpenFlow switches allow the collection of this data.

Using the MapReduce paradigm, based on the obtained data, we generate statistics on switches, using a pair (*timestamp*, *switch*) as a key. The following points are considered for the analysis of traffic on switches [7]:

Summation - the value of the throughput of the switch depends on the value of the counters on each port of the switch, so each value must be summed. Mathematical expectation - calculates the mathematical expectation of each value for the switch. The standard deviation is used to analyze the current values of statistics.

Using the key (*timestamp*), network statistics are generated for ports and switches. For traffic analysis was used z-value, which characterizes the deviation of the current values from the expected:

$$z = \frac{x - \mu}{\sigma}, \quad (2)$$

where x is the current value of the counter, μ is the mean value of the counter, σ is the standard deviation.

The result of the analysis of traffic on routes is a table with the data received at collection of statistics and calculation in the course of performance of algorithms of generation.

After performing the analysis using the MapReduce paradigm, we obtain a table that contains as a key the values obtained during the operation of the algorithm for generating key / value pairs, and as the values of the path statistics and the analyzed data.

Table 1

Path traffic analysis result model

s1:h1-h7	s1:h1-h7 statistics columns	s1:h1-h7 traffic analysis columns	path statistics columns
s1:h2-h5	s1:h2-h5 statistics columns	s1:h2-h5 traffic analysis columns	path statistics columns
s2:h1-h7	s2:h1-h7 statistics columns	s1:h1-h7 traffic analysis columns	path statistics columns
s2:h4-h6	s2:h4-h6 statistics columns	s2:h4-h6 traffic analysis columns	path statistics columns
.....			

Conclusions. The problem of designing traffic in SDN networks is thought over. Influenced by existing solutions for traffic analysis and modern technologies, the solution of traffic engineering using Big Data paradigms was considered. Described by the sequence in which the data are obtained to obtain complete statistics. The algorithm for generating key/value pairs and traffic analysis has also been implemented and supplemented. In further research could be considered the method of lambda architecture to solve the problem.

References

1. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic Matrix Estimator for OpenFlow Networks," in Springer, pp. 201–210, 2010.
2. Y. Yu, C. Qian, and X. Li, "Distributed and collaborative traffic monitoring in software defined networks," in Proceedings of the third workshop on Hot topics in software defined networking - HotSDN '14, (New York, USA), pp. 85–90, 2014.
3. Z. Bholebawa, R. K. Jha, and U. D. Dalal, "Performance Analysis of Proposed OpenFlow-Based Network Architecture Using Mininet," Wireless Personal Communications, vol. 86, no. 2, pp. 943–958, 2015.
4. K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. Capretz, "Challenges for MapReduce in Big Data," in 2014 IEEE World Congress on Services, pp. 182–189, IEEE, Jun. 2014.
5. Queiroz, Wander, "Big Data for Traffic Engineering in Software-Defined Networks" (2019) pp. 60-67. Electronic Thesis and Dissertation Repository. 6147.
6. J. Lin, "In Defense of MapReduce," IEEE Internet Computing, vol. 21, pp. 94–98, May 2017.
7. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine Learning With Big Data: Challenges and Approaches," IEEE Access, vol. 5, pp. 7776–7797, 2017.

AUTHORS

Dmytro Babko – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: dmitry.babko@gmail.com

Yurii Kulakov (supervisor) – Doctor of Technical Sciences, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.