Risk Analysis of Blocked Rate Predictions for SDN Load Balancing Using Monte Carlo Simulation

Aymen Hasan Alawadi
Dept. of Telecomm. and Media Informatics
Budapest University of Technology and Economics
Budapest, Hungary
aymen@tmit.bme.hu

Sándor Molnár
Dept. of Telecomm. and Media Informatics
Budapest University of Technology and Economics
Budapest, Hungary
molnar@tmit.bme.hu

***Abstract*—The emergence of large data centers and virtualization needs better and smarter solutions for traffic scheduling and load balancing. Data centers benefit from SDN regarding centralized monitoring and management for traffic routing. In general, the traffic in the data center environment can be classified as elephant and mice flow. Researchers showed that there is a significant amount of data carried over elephant flows; therefore, it should be conserved and maintained thoroughly. In this work, we introduce a stochastic performance evaluation model for estimating blocked rate prediction and risk analysis of the elephant flows for a load balancing data center with fat-tree topology using the SDN paradigm. The general procedure of the evaluation includes the estimation of the distribution of the path available bandwidth, including bandwidth error tolerance. The proposed model relies on Monte Carlo simulation to generate future prediction behavior of the load balancing technique. The achieved results examined with Value at Risk (VaR) along with statistics to percept the complete picture of the load balancing behavior.**

Keywords*— *Load balancing, SDN, Risk analysis, Value-at-Risk, Simulation, Measurements techniques

I. INTRODUCTION

Nowadays, many companies tend to adopt data center solutions since they offer bandwidth preservation and flexibility to serve a large number of hosts and applications. However, the types of data center applications are diverse from ordinary web activities to scientific computing and MapReduce operations, that demand high available bandwidth and scalability [1]. Because of these substantial requirements, many data center topologies evolved like hyperx [2], flattened butterfly [3], and fat-tree [4]. For better traffic management, other techniques have emerged, like throughput forwarding and load balancing [5]. Ordinarily, the uses of the data center produce two types of flows classified as mice and elephant flows [6]. Mice flows, which are the smallest and shortest-lived TCP flows on the network and more conservative to delay.

On the other hand, the large and long-lived TCP flows (elephant flows) are more affected by the available throughput [5]. Regularly, the occurrence of the elephant flows found at the data centers is smaller than what mice flows have, but they occupy most transferred data due to the nature of the used applications especially in data mining, machine learning, and data analysis [6]. As long as the elephant flows continue to grow in size, they will be hard to be processed, managed, and scheduled, and will worsen the flow completion time of mice flows.

SDN (Software Defined Networking) appears as a new orientation in managing and controlling the data centers networks. The centralized paradigm, along with the characteristic of decoupling the control layer from the data layer enables effective resource management compared with the traditional networks through the OpenFlow protocol [7]. Hence, the SDN paradigm used to collect parameters from the entire network including information about any traffic in the data layer and use them for monitoring and prediction purposes in the control layer. Consequently, SDN is significantly employed by the research community in traffic load balancing and forwarding [8]. Therefore, several load balancing and traffic scheduling algorithms proposed to handle this issue, but there are many performance evaluation issues in term of preserving the elephant flows [1] [5]. The applying of deterministic and statistical evaluation approaches is inefficient, particularly in fully stochastic environments.

In SDN networks, OpenFlow protocol provides many statistical and numerical information about the monitored network regarding the flows and packets passed through any monitored port in a flow table [9]. Sokolov et al. in [10] suggested using this information to identify weak points of the network architecture and predicting the potential risks of problems that may occur primarily in the fail-safety task in the network. However, the information in the SDN flow table considered trivial and not adequate for the applications above the SDN architecture [11]. Therefore, Luo et al. in [11] proposed a context-aware traffic forwarding service for SDN applications assuming a constrained optimization path problem for decision making. The authors suggested two factors including capacity and cost of a service that would impact the service composition, and they found that cost context can be quantified from other context factors. So, the authors collected the cost and the factors data online to derive their relations by multiple linear regression analysis. As for the elephant flow management, several studies have emerged dealing with load balancing and scheduling issues like [5] and [12]. Al-Fares et al. in [1] proposed

Hedera, which is a dynamic and central flow scheduling to utilize data center bisection bandwidth. The authors found the performance of Hedera primarily based on the rate and duration of the flows in the network, but they did not show the future prediction for the blocked elephant flows. Zakia et al. proposed another technique in [8] relies on flow priority to find the shortest paths. The authors evaluated it regarding throughput consumption, RTT delay, and packets loss in a fat-tree data center. Long et al. proposed LABERIO [13], which is a fat-tree dynamic load balancing technique based on the real-time bandwidth utilization rate by considering max-min remainder capacity strategy (MMRCS) for path selection. To generate the required traffic flows, the authors examined three different traffic patterns, including uniform, semi-uniform, and center-based, to distribute the flows among the nodes of the network [13]. As for the evaluation purposes, the authors did not examine the stochastic behavior of the traffic. Alternatively, they assumed a fixed flow size and an upper-bound bandwidth on each link for the tested topology.

Accordingly, in this paper, we present the following contributions:

- A new performance evaluation model for SDN load balancing of network elephant flows. The model includes estimation of the probability distribution of path throughput, i.e., available bandwidth and error tolerance with specific measurements scenarios and Anderson-Darling test.
- Prediction of the risks of the used load balancing technique by Value at Risk (VaR) analysis for the blocked elephant flows produced by implementing the load balancing technique.

The rest of the paper organized as follows. In section II, we describe the proposed model. In section III, we describe the simulation processes, results, and discussion. We finally conclude in section IV.

II. MODEL DESCRIPTION

The proposed model consists of selecting the appropriate SDN load balancing technique to investigate and evaluate the uncertainty behaviors of the fat-tree data center network in balancing elephant flow. Anderson-Darling (A-D) hypothesis testing applied upon the obtained uncertainty behaviors samples to get the appropriate probability distribution function for each of them. Monte Carlo simulation utilized as a Value at Risk (VaR) analysis model to predict the amount of the blocked elephant flows resulting from the used load balancing technique in a fat-tree topology.

A. Selecting a Load Balancing technique in Fat-tree topology

In general, the load balancing techniques used in SDN classified into static and dynamic strategies or sort of a combination of both [8]. The adopted technique to evaluate the proposed model is the dynamic one [9], since the static methods are incompetent and cannot predict the network changes over time. The selected technique begins by collecting the information about the topology to build a directed graph data structure by sending LLDP (Link Layer Discovery Protocol) packets. Then, it determines the shortest available path(s) between the source switch and destination switch in the directed graph. The residual bandwidth of the links is determined based on the current traffic periodically (every 10 seconds) and stored in a network data structure. Finally, the best forwarding path is decided based on the existing network traffic for load balancing and traffic scheduling. The preferred technique has been written as a Ryu controller application [9].

In this paper, fat-tree topology used in constructing the primary network environment; since it considered one of the essential topologies in building efficient, scalable, and cost-effective data centers. Fat-tree topology has constructed from three main layers of connected switches, including core, aggregate, and edge. However, a *K*-4 fat-tree data center interconnects topology tested in mininet environment with fixed links capacity from host to switch and switch to switch links with 10 Mbit each (Figure 1).

B. Collecting and normalizing the data

We conducted several measurement scenarios to collect the desired uncertainty data to perform Monte Carlo simulations analysis. These scenarios included estimating the available bandwidth with some error tolerance for the determined path(s) between arbitrary source and destination. In fat-tree topology, there are $(k/2)^2$ equal cost shortest paths available between any two hosts from different pods [10].

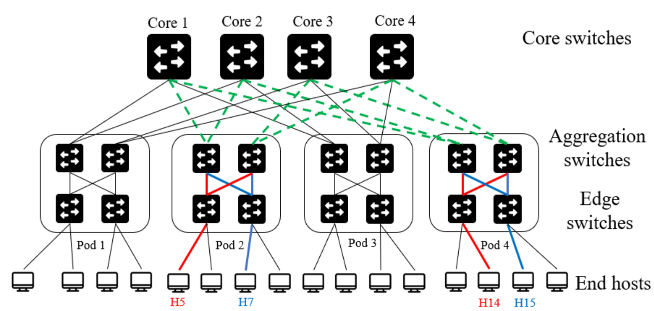

Fig. 1. *K*-4 fat-tree data center.

Since all of the links in the tested topology have the same configurations concerning links bandwidth. Therefore, only a sample of the network selected for the testbed. For this reason, the longest path considered between a server and a client found in different pods. The sample involves two parts, the first one arranged for generating the traffic noise (mice flow), as depicted in Figure 1 with the red line between (H5) and (H14). To achieve network flow contention, the first part is intersected with the second part in all available paths between pod 2 and pod 4 as shown in the green dotted line in Figure 1. The second path is designated for generating and testing the elephant flow between (H7) and (H15) as represented in the blue line in Figure 1. Note that the balancer should choose among the four shortest paths between the sources and the destinations in both parts.

The process of measurement the testing sample depends on injecting high-density traffic to immerse the network. Therefore, multiple arbitrary small files i.e., TCP mice flow generated between (H5) and (H14) varying between 100 and 400 Kbyte in size for 2000 files, including 654 files in sizes within 100 to 200 Kbyte, plus 659 files within a range of size 200- 300 Kbyte, and 687 files in sizes of 300-400 Kbyte. However, it has been proven in [14] that the traffic in the typical data center can be so bursty; hence, it cannot be predicted in any link. However, the main focus of this measurement is to saturate the testing part of the network as stated in Hedera testbed [1], to evaluate the maximum amount of data the network can handle with this kind of flow contention. Taking into consideration that the possible places of collisions are inside pod 2, pod 4 besides the zone of the core switches. Therefore, the chosen sample network measured using iperf (between H7 and H15) for 100 seconds to check the maximum amount of data the network can handle. This testing period decided because the load balancing technique initially configured to monitor the data center links every 10 seconds. However, the sample mean of the measured throughput was 1.05 MB/sec, i.e., 105 Mbyte could be delivered from the server to the client without any loss.

As for measuring the available bandwidth for different elephant flow sizes along with other flows, a file of 105 Mbyte was divided by weights of 10, 9, 8, 7, 6, 5, 4, 3, 2 and 1, respectively. The yielded files transferred as a TCP flow from H7 to H15, as shown in the blue line in Figure 1. The transformation of the generated data will be in the same duration of the former 100 seconds to check how the load balancer process will affect the throughput, taking into consideration the uncertainty of the applied mice flow in terms of size and chosen path between the source and destination. The measurement process repeated six times for each weight to estimate the available bandwidth by taking the arithmetic sample mean of the measurements. Figure 2 below shows the proposed scenarios with measurements values.

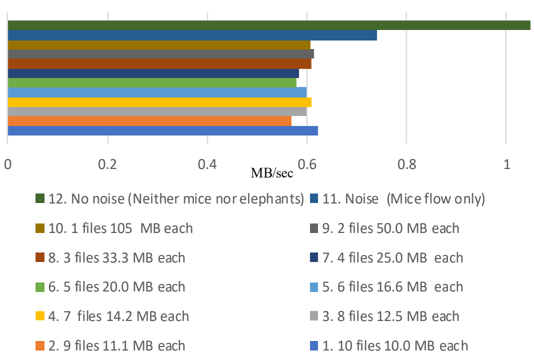

Fig. 2. The sample mean values of the available bandwidth measurements.

To have some residual variation, the tolerance of error in bandwidth measurement was estimated using the arithmetic sample standard deviation for the available bandwidth measurements. The maximum value of the calculated standard deviation considered since it indicates a more significant value than the estimated sample mean in the worst-case evaluation concerning the high-density traffic environment (Figure 3).

C. Goodness of fit

The Goodness of Fit (GOF) test applied to assess the compatibility of the collected data with some of the well-known distribution functions. For this purpose, EasyFit professional [15] adopted, which is a specialized program in deciding the best fitting for the trained sample of data. Accordingly, the Anderson-Darling (A-D) test chosen as a hypothesis testing to evaluate the distribution of the collected data. Anderson-Darling defined as (A^2).

$$A^2 = -N - S \tag{1}$$

Where S:

$$S = \sum_{i=1}^{N} \frac{(2i-1)}{N} [\ln F(Yi) + \ln(1 - F(Y_N + 1 - i))] \tag{2}$$

Where F is the cumulative distribution function of the compared distributions and Yi are the ordered data.

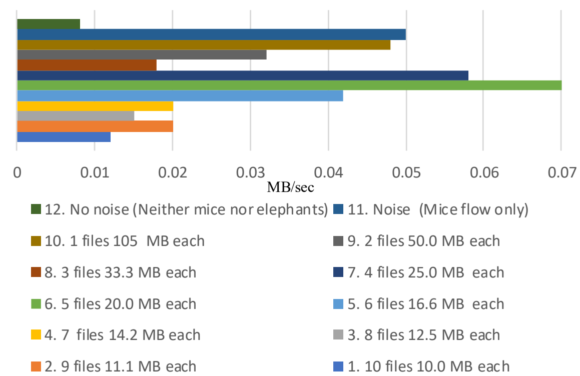

Fig. 3. The available bandwidth error tolerance measurement.

To accomplish this kind of testing, a null hypothesis testing performed, where H_0 identified when the tested data specify the distribution, and H_1 recognized when the data do not follow the distribution. To come up with the desired distribution, A-D assumes a significance level α like (0.01 and 0.05) and compares the tested statistics (A^2) with some of the critical values of the most widely used distribution. The hypothesis of the measured distribution will be discarded if the value of (A^2) exceeds the critical value at a significant level. Note that the significant level of 0.05 is typically used for most applications [16]. As a consequence of conducting the null hypothesis testing on the samples of the available bandwidth measurements, the Negative Binomial distribution (N-B) determines as the nearest distribution for the evaluated data. Generally, N-B distribution recognized as a discrete probability distribution to represent the number of successes in a sequence of independent Bernoulli trials until reaching the specified number of non-random failures occurs. Studies like [17] and [18] proved that the N-B distribution is reliable to simulate real data. However, the N-B probability mass function has been applied to produce the required random samples of the available bandwidth data in the proposed evaluation model used in Monte Carlo simulation (equation 3).

$$P_r(A) = \binom{r+c-1}{c} p^c (1-p)^r \quad (3)$$

Where A is the random variables of the available bandwidth, r is the number of failures with $1-p$ probability, c is the number of success or failure and, p is the probability of success.

We evaluated the error in bandwidth measurement samples (tolerance data) for fitting to obtain the probability distribution function of the error tolerance. The measured data found suited the Binomial Distribution (B-D) at a significant level of 0.05. Generally, B-D is a discrete probability distribution used to find the probability of a successful event in the case of two possible outcomes. The probability mass function of the B-D (equation 4) used to generate the random variables of the error in bandwidth measurements.

$$P_r(E) = \binom{n}{g} p^g (1-p)^{n-g} \quad (4)$$

Where E is the random variable of the error measurement, n is the number of trials, g is the number of successes with the probability of p^g, while $n-g$ is the number of failures occur what the probability $1-p$.

D. Monte Carlo Simulation

Monte Carlo method is a technique used to simulate the stochastic behavior of a system or to evaluate a set of uncertainty input of a deterministic model. Mathematically, it is not possible to predict and determine all possible outcomes of a system [19]. Therefore, Monte Carlo simulation process applies to create multiple predicted scenarios by considering the probability distribution of the stochastic input parameters of the system. Accordingly, this process repeated hundreds or thousands of times to produce potential scenarios or solutions with a range of probabilities. However, the affecting of stochastic parameters of the link available bandwidth studied with some of the error tolerance to determine the value at risk of the elephant flow during traffic load balancing. Accordingly, after obtaining the required samples from equations 3 and 4, along with the estimated elephant flow in size and volume (Table 1), the proposed evaluation equation is to be formed as follows.

$$Pre\ (V, S, A, E) = B_m = V_i \times (S_j - (A_k + E_l)) \quad (5)$$

Where B_m is the predicted blocked rate, V_i is the different volumes of the evaluated elephant flows, S_j is the sizes of the elephant flows (Table 1), A_k is the available bandwidth measurements and E_l is for the error tolerance variables. The assumed values for the size *S* represent the maximum bandwidth the physical link can handle (10 Mbps) to the minimum elephant flow size (10% of link capacity) determined by Al-Fares et al. in Hedera [1].

TABLE I. ELEPHANT FLOW PARAMETERS.

Elephant flow	***Size S***	***Volume V***
Large	1 Mbyte	100
Normal	0.5 Mbyte	75
Small	0.1 MByte	50

The volume parameter *V* describes the amount of the flow within a particular path, and it measured by the unit of second.

III. RESULTS AND DISCUSSIONS

To produce the final probability distribution of the blocked traffic prediction, the proposed model (equation 5) was repeated one million times (Figure 4). In general, Monte Carlo simulation produces the total number of iterations that represent the prediction of loss and profit. In network environments and particularly in available bandwidth (throughput) consumption aspect, the network is not forming any beneficial outcome. Therefore, these values (positive values) recognized as an incompatible with the proposed model of loss prediction (blocked traffic). However, the false values achieved 64.24% of the total iterations for the evaluated values, and it means that most of the elephant flow including small and regular sizes have passed without affecting by the stochastic parameters.

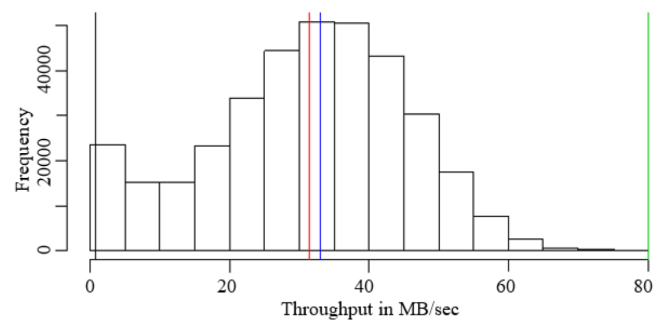

Fig. 4. Histogram of Monte Carlo simulation for the blocked rate.

However, the rest of the values (35.76%) separated and formed the blocked rate prediction, as depicted in Figure 4.

Several kinds of information can be discovered from the histogram plot. First of all, the distribution starts with high frequency for the value of traffic blocking with a minimum of 0.75 Mbyte, then dramatically decreases.

A. Distribution shape analysis

Common distribution shape measurements were calculated, like Skewness and Kurtosis, to analyze the bias of the blocked traffic rate histogram. Primarily, Skewness used for measuring the symmetry of the distribution, and it has two values; positive and negative. The positive value (right skew) indicates that the mean value is higher than the median value, while the negative value (left skew), suggests the opposite. Equation 6 below describes the skewness degree calculation for the observed distribution.

$$skew = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2)^{\frac{3}{2}}} \quad (6)$$

Where x_i holds n observations and $\bar{x}$ is the mean values of the observations.

Kurtosis is another important shape measurement utilized for describing the distribution tail thickness compared to the Normal distribution. There are three types of Kurtosis; including mesokurtic, leptokurtic, and platykurtic distributions. Mesokurtic distribution has the same characteristics of the Normal distribution concerning the extreme tail values, while leptokurtic has higher tail values due to the long tail, as for the platykurtic type, it has a precise tail with fewer outliers [20].

$$Kurtosis = \frac{\frac{1}{n}\sum_{m=1}^{n}(B_m-\overline{B_m})^4}{\left(\frac{1}{n}\sum_{m=1}^{n}(B_m-\overline{B_m})^2\right)^2} - 3 \quad (7)$$

Where B_m holds n observations of the predicted blocked rate and $\overline{B_m}$ is the mean values of the observations.

Based on the sample distribution and the calculated Kurtosis (-0.43) and Skewness (-0.31), the distribution of the prediction is not formed like a complete Normal distribution. Alternatively, it is left-skewed since the median value (33 MB/s depicted by the blue line in Figure 4) precedes the mean value (31.45 MB/s represented by the red line). Therefore, the concentration of the produced distribution leans slightly to the right, and this what makes the blocking rate increases in the future. Likewise, the prediction distribution considered as a platykurtic distribution since it has a value (- 0.43) comparing with the Normal distribution. Hence the model distribution produces fewer extreme values for the outliers at the tail (80 MB/s) as presented in the green line in Figure 4.

B. Value at Risk (VaR) analysis

Even though, the histogram and the statistics provide comparative information about the behavior of the model and the blocked rate prediction, Value at Risk (VaR) analysis could provide more deep analysis based on some confidence [21]. Mainly, the Monte Carlo simulation model considered one of the three common types of VaR, with Normal Linear model and Historical Simulation model. To calculate VaR based on Monte Carlo simulation, the proposed model should produce independent and random future simulations with kind of normality assumed based on standard deviation [21]. In this research and for better generalizability, the chosen confidence level was 95%, since outlier results would appear clearly with the more significant percentage. The probability of the confidence level calculated by taking the quantile function (equation 8) [21].

$$VaR = -\mu_n + \emptyset^{-1}(1-u)\sigma_n \quad (8)$$

Where μ_n is the mean of the returns of the prediction, $\emptyset$ is the function of the standard Normal distribution, σ_n is the standard deviation of the returns and $(1-u)$ used for the chosen confidence level.

The chosen confidence level showed that the worst loss prediction would not exceed 53 MB/s, while in the case of 75%, the value would be 42 MB/s. This amount of the blocked rate represented the future prediction amount of the affected elephant flow produced by implementing the load balancing technique. The blocked TCP data would be retransmitted and causes extra network bottlenecks and would affect the mice flows. Therefore, this kind of investigations provides a dynamic analysis of how the elephant flows will be treated while using such a load balancing technique and how it will continue to perform in the future.

As a conclusion for the research findings, we suggest that the performance evaluation of the new developing algorithms that handling the elephant flows should consider the uncertainty behaviors of the tested network and predict the amount of the blocked data resulting from the use of the algorithm. To the best of our knowledge, most of the developed heuristic algorithms are evaluated using the average values for the obtained data without naming the probability distribution function. Note that the expected value (average value) for random variables does not exist for some distributions that have long tails such as Cauchy distribution [22]. Therefore, taking the average for any sample of the data does not actually represent the expected value of the predicted data. We summarize some load balancing works that rely on considering the average values when evaluating the performance of the algorithm (Table 2).

TABLE II. SELECTED PAPERS WITH THEIR PROPERTIES.

Paper	*Main idea*	*Performance evaluation*
Lei et al. [23]	Proposing an adaptive algorithm to determine multipath routing for the flows based on the flows demand and available network resources such as available bandwidth and shortest path.	Consider end to end delay and throughput.
Fizi et al. [24]	Rerouting algorithm based on bandwidth utilization and loss rate of the paths.	Considering the average yielded throughput on the hosts.
Long et al. [13]	Routing algorithm-based bandwidth utilization rate by considering a max-min remainder capacity strategy for path selection.	Overall average bandwidth utilization.
Tu et al. [25]	Programmable middleboxes to collect information from the switches and the servers to perform the load balancing.	To describe the delay of the flows by using the algorithm, they took the average delay of the original data center.

However, the quality of the prediction produced by independent and random variables relies on current observations to predict future performance. Therefore, the model and assumptions need to be accurate enough.

There are some further improvements to the methodology by studying other stochastic parameters such as link transmission delay and queuing delay to evaluate their extent impact on the elephant flows. Nevertheless, in this study, some outliers appeared at the end of the predicted distribution tail. Generally speaking, such an outcome is undesirable. Therefore, the simulation will be more efficient if it ends until arriving at the desired precision [26].

IV. CONCLUSIONS

In this work, we have designed, implemented, and analyzed a new load balancing performance evaluation model based on a high-density stochastic data center network to estimate the value at risk for the blocked elephant flows. Although some existing algorithms deal with the elephant flow traffic scheduling or load balancing, they did not predict the future uncertainty impacts of the network environment on the elephant flows. It can be observed from our results that 35.76% of the evaluated TCP elephant flows is exposed to be blocked, and it is expected to be higher in the future since the probability distribution of the blocked rate prediction is left-skewed. On the other hand, the current risk analysis indicates that the worst case of the blocked rate will not exceed 53 MB/s from transmitted flows with a confidence interval of 95%. However, the evolution process of the load balancing needs to have proper awareness in term of predicting future behavior regarding elephant flows preservation. Note that our research relies on applying a standard load balancing technique with certain assumptions and observations on the stochastic behaviors of the data center topology to predict the future of the balancing impact of the elephant flows. Finally, further research hence is needed to evaluate more complicated load balancing techniques with other factors for the network uncertainty.

References

[1] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, 'Hedera: dynamic flow scheduling for data center networks', *Proceedings of the 7th USENIX conference on Networked systems design and implementation*. USENIX Association, pp. 19–19, 2010.

[2] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, 'HyperX: topology, routing, and packaging of efficient large-scale networks', in *Prof. of Conf. on High Performance Computing Networking, Storage and Analysis*, 2009, p. article no. 41.

[3] J. Kim, W. Dally, and D. Abts, 'Flattened butterfly: a cost-efficient topology for high-radix networks', *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 126–137, Jun. 2007.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat, 'A scalable, commodity data center network architecture', *Proc. ACM SIGCOMM 2008 Conf. Data Commun. - SIGCOMM '08*, p. 63, 2008.

[5] J. A. Rashid, 'Sorted-GFF: An efficient large flows placing mechanism in software defined network datacenter', *Karbala Int. J. Mod. Sci.*, vol. 4, no. 3, pp. 313–331, 2018.

[6] T. Mori, S. Naito, R. Kawahara, and S. Goto, 'On the characteristics of Internet traffic variability: Spikes and elephants', in *Proceedings - International Symposium on Applications and the Internet*, 2004, pp. 99–106.

[7] N. McKeown *et al.*, 'OpenFlow: enabling innovation in campus networks', *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69, Mar. 2008.

[8] U. Zakia and H. Ben Yedder, 'Dynamic load balancing in SDN-based data center networks', in *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2017*, 2017, pp. 242–247.

[9] Open Networking Foundation, 'OpenFlow Switch Specification', 2013.

[10] V. Sokolov, I. Alekseev, D. Mazilov, and M. Nikitinskiy, 'A network analytics system in the SDN', in *2014 First International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)*, 2014, pp. 1–3.

[11] S. Luo, J. Wu, J. Li, L. Guo, and B. Pei, 'Context-Aware Traffic Forwarding Service for Applications in SDN', in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015, pp. 557–561.

[12] Y. L. Lan, K. Wang, and Y. H. Hsu, 'Dynamic load-balanced path optimization in SDN-based data center networks', *2016 10th Int. Symp. Commun. Syst. Networks Digit. Signal Process. CSNDSP 2016*, vol. 4, no. 3, pp. 313–331, 2016.

[13] H. Long, Y. Shen, M. Guo, and F. Tang, 'LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks', in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2013, pp. 290–297.

[14] T. Benson, A. Anand, A. Akella, and M. Zhang, 'Understanding data center traffic characteristics', *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, p. 92, 2010.

[15] A. Drokin, 'Easyfit - Distribution fitting software'. p. www.mathwave.com, 2010.

[16] A. Drokin, 'EasyFit - Anderson-Darling Test'. [Online]. Available: http://www.mathwave.com/help/easyfit/html/analyses/goodness_of_fit/anderson-darling.html. [Accessed: 18-Nov-2018].

[17] J. Lu, J. K. Tomfohr, and T. B. Kepler, 'Identifying differential expression in multiple SAGE libraries: an overdispersed log-linear model approach', *BMC Bioinformatics*, vol. 6, no. 1, p. 165, Jun. 2005.

[18] M. D. Robinson and G. K. Smyth, 'Moderated statistical tests for assessing differences in tag abundance', *Bioinformatics*, vol. 23, no. 21, pp. 2881–2887, Nov. 2007.

[19] J.-C. Walter and G. T. Barkema, 'An introduction to Monte Carlo methods', *Phys. A Stat. Mech. its Appl.*, vol. 418, no. 1, pp. 78–87, Jan. 2015.

[20] Investopedia, 'Kurtosis'. [Online]. Available: https://www.investopedia.com/terms/k/kurtosis.asp. [Accessed: 21-Nov-2018].

[21] C. K. Thim, M. Nourani, and Y. V. Choong, 'Value-at-risk and conditional value-at-risk estimation: A comparative study of risk performance for selected malaysian sectoral indices', in *ICSSBE 2012 - Proceedings, 2012 International Conference on Statistics in Science, Business and Engineering: 'Empowering Decision Making with Statistical Sciences'*, 2012, pp. 224–228.

[22] R. W. Hamming, *The Art of Probability for Engineers*. Addison-Wesley, 1991.

[23] Y. C. Lei, K. Wang, and Y. H. Hsu, 'Multipath routing in SDN-based Data Center Networks', in *2015 European Conference on Networks and Communications, EuCNC 2015*, 2015, pp. 365–369.

[24] F. S. Fizi and S. Askar, 'A novel load balancing algorithm for software defined network based datacenters', in Proceedings of the 2016 International Conference on Broadband Communications for Next Generation Networks and Multimedia Applications, CoBCom 2016, 2016, pp. 1–6.

[25] R. Tu, X. Wang, J. Zhao, Y. Yang, L. Shi, and T. Wolf, 'Design of a load-balancing middlebox based on SDN for data centers', in *Proceedings - IEEE INFOCOM*, 2015, vol. 2015–Augus, pp. 480–485.

[26] R. A. C. Capuruço and L. F. Capretz, 'Evaluation and assessment of recommenders using Monte Carlo simulation', *CEUR Workshop Proc.*, vol. 872, 2012.