Containerized 5G+ Network Slices

Thesis Proposal of Degree

Edier Dario Bravo Bravo
Jefry Nicolas Chicaiza

Advisor: Ph.D. Oscar Mauricio Caicedo Rendon
Co-Advisor: Ph.D. Por definir

University of Cauca
Faculty of Electronic and Telecommunications Engineering

Degree in Electronic and Telecommunications Engineering
Line of Research in Advanced Services of Telecommunications
Popayan, May 2022

Contents

List of Figures

List of Tables

1 Problem Statement

Nowadays, the ever-growing Internet-based applications and services demand a huge amount of computing and storage resources. Data centers represent the key infrastructure that supports and provides such resources as a large number of servers interconnected by a specially designed network, called Data Center Network (DCN) [1]. The goal of DCN is to provide significant bandwidth capacity in order to achieve high throughput[1] and low-latency[2].

Everyday, DCN managers are looking for solutions that allow optimizing these performance requirements (*i.e.*, high bandwidth, high throughput, and low latency) without the need to add more capacity to the network. Traffic engineering represent a great opportunity in this realm, particularly, load-balancing is a desirable feature for reducing network congestion while improving network resource availability and application performance [2, 3]. A well-known technique for achieving load-balancing in DCN is multipath routing, which distributes traffic over multiple concurrent paths such that all the links are optimally loaded [4]. Hereinafter, when this document mentions routing in DCN is particularly referring to intra-DCN routing.

Currently, the most prevalent multipath routing solution in DCN is Equal Cost Multiple-Path (ECMP), which generally uses a hash function in every switch to assign each incoming flow to an output port [5, 6]. This hash function uses the header information of the flow (usually, the five-tuple) to compute a result (i.e., output port). For example, Facebook's DCN in Altoona employs BGP to populate the routing tables and ECMP for routing through the equal cost paths. Nevertheless, traffic in DCN shows a widespread distribution of flow sizes (i.e., mice and elephant flows), causing hot-spots in ECMP-based DCNs, i.e., some links are highly utilized while others are underutilized. For example, if two mice flows and two elephant flows arrive at the same switch, it is possible that the ECMP-based switch assigns the two mice flows to one of the outgoing ports and the two elephant flows to one of the another outgoing ports. Therefore, the link transporting the two mice flows is going to be free much faster than the link transporting the two elephant flows, causing an underusing and overloading of links, respectively.

Few research efforts [7, 8, 9, 10] have focused on analyzing the traffic characteristics in DCNs. These researches converge that the hot-spot problem is very common in ECMP-based DCN and that it is not a capacity problem. For example, in some DCNs, 10% to 25% of links are hot-spots even with less than 25% of capacity utilized [8]. Few approaches like Facebook's Altoona [11] propose to prevent the degrading of elephants by making the network multi-speed, however, this is not an efficient approach; rather than adding more capacity, the issue is selecting a routing mechanism for drawing traffic effectively.

Recent approaches [12, 13, 14, 15, 16, 17, 18] have addressed this issue by leveraging the Software-Defined Networking (SDN) paradigm. For example, Hedera [19] performs a periodic polling of edge switches to collect flow statistics and detect elephant flows. The SDN controller calculates and installs the paths for elephant flows, while maintaining ECMP for mice flows. Nevertheless, the short inter-arrival times of flows (less than 100ms) and the high number of active flows per second cause that solutions like Hedera increases traffic overhead and controller processing, even when just a small fraction of the flows (i.e., elephants) actually matter.

[1] Total number of packets processed per second
[2] Average processing time used for a single packet

In summary, considering the traffic characteristics in DCNs, it is difficult to build a solution for minimizing the maximum load in links in order to optimize the traffic routing (i.e., high throughput and low delay while effective use of resources). Therefore, this thesis project focuses on solving the following research question:

How to minimize the maximum traffic load of links in a DCN for enabling high throughput and low delay while maintaining efficient use of resources?

2 State of the Art

2.1 Background

2.1.1 Data center network

DCN encompasses the communication infrastructure of data centers that aim to interconnect a large number of servers with significant bandwidth capacity in order to achieve high throughput and low-latency [1]. Different DCN topologies have been designed to meet such performance requirements. In general, these DCN topologies can be classified into three categories based on the routing and switching equipment used to forward or process network traffic [20, 21, 22]: *switch-centric*, *server-centric*, and *hybrid*. Switch-centric topologies consider switches as the only relay nodes (*i.e.*, routing decisions) and servers as mere endpoints. Server-centric topologies define servers as both endpoints and relaying nodes. Hybrid topologies use a combination of electrical, optical, and/or wireless equipment to add extra bandwidth capacity to the DCN.

Currently, most of the DCN deployments follow the switch-centric topology, particularly the tree-based design, such as Facebook's Altoona [7] and Google's Jupiter [23]. In switch-centric, the switches are interconnected in a hierarchical model with multiple layers and the servers are connected to the switches of the lowest layer, known as *edge* or Top-of-Rack (ToR). The tree-based design is an instance of the Clos network with a degree defined according to the network scale indicating the number of layers. For example, VL2 [24] and Fat-tree [25] arrange low-cost commodity switches in a tree-based topology with fourth-degree, namely, from bottom-up, one layer of servers and three layers of switches: *edge*, *aggregation*, and *core* (see Figure 1).

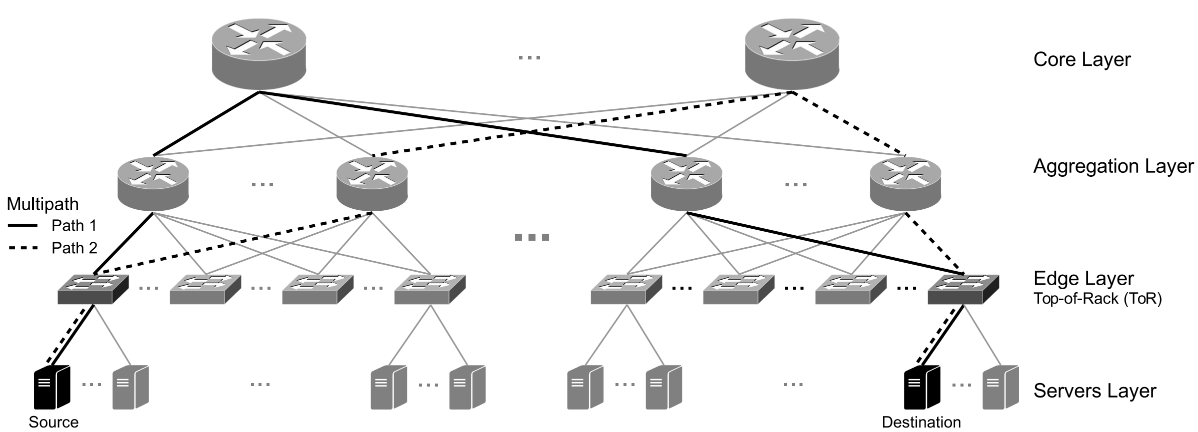


Figure 1. Tree-based DCN with fourth degree and multipath routing

2.1.2 Traffic engineering

Traffic engineering involves the methods for measuring and managing network traffic to optimize the performance of the network [3, 26]. This optimization requires providing appropriate traffic requirements (*e.g.*, throughput, delay, packet loss) while efficiently—in terms of cost and reliability—utilizing network resources (*e.g.*, bandwidth).

Load-balancing is one of the most well-known traffic engineering methods. As part of the flow management dimension[3], load-balancing controls and optimizes the routing function to minimize the maximum load across the links [2]. The goal is mapping traffic flows from the heavily loaded paths to the lightly loaded paths for avoiding congestion (*i.e.*, hot-spots) and increasing network throughput and resource utilization. *Multipath routing* has shown to effectively achieve load-balancing by distributing traffic over multiple concurrent paths such that all the links are optimally loaded [4]. Figure 1 depicts two disjoint[4] paths in a tree-based DCN. In practice, multipath routing protocols may split the traffic at different levels of granularity, such as per-flow, per-sub-flow, and per-packet. These protocols may run at distinct layers of the TCP/IP model. For example, ECMP [28] and Valiant Load Balancing (VLB) [29] work at the network layer, while Multi-Path TCP (MPTCP) [30] operates at the transport layer.

2.1.3 Software-defined networking

SDN represents one of the most accepted and attractive trends—in research and industry—for defining the architecture of future networks [31, 32]. From a general aspect, SDN decouples the control and forwarding planes for enabling a simpler network operation from a logically centralized software program, usually known as the *controller* [33]. The control plane (*i.e.*, the controller) compiles decision policies and enforces them on the forwarding plane (*i.e.*, switches and routers) through a vendor independent protocol. OpenFlow [34] is the most well-known open standard SDN protocol because its widespread use by vendors and research.

SDN provides four major advantages for operating networks [27]: *(i)* a centralized global view about the network state (*e.g.*, resource capabilities and dynamic status) and the deployed applications (*e.g.*, QoS and SLAs), *(ii)* a dynamic programmability of multiple forwarding devices (*e.g.*, allocating resources to prevent congestion and improve performance), *(iii)* open interfaces for handling the Data Plane (*e.g.*, OpenFlow) and for developing the Applications (*e.g.*, APIs based on protocols and programming languages); and *(iv)* a flexible flow management (*e.g.*, multiple flow tables in OpenFlow). These unique features lead the SDN architecture to emerge as a promising scenario for efficiently and intelligently implementing management techniques, particularly for traffic engineering.

2.1.4 Machine learning

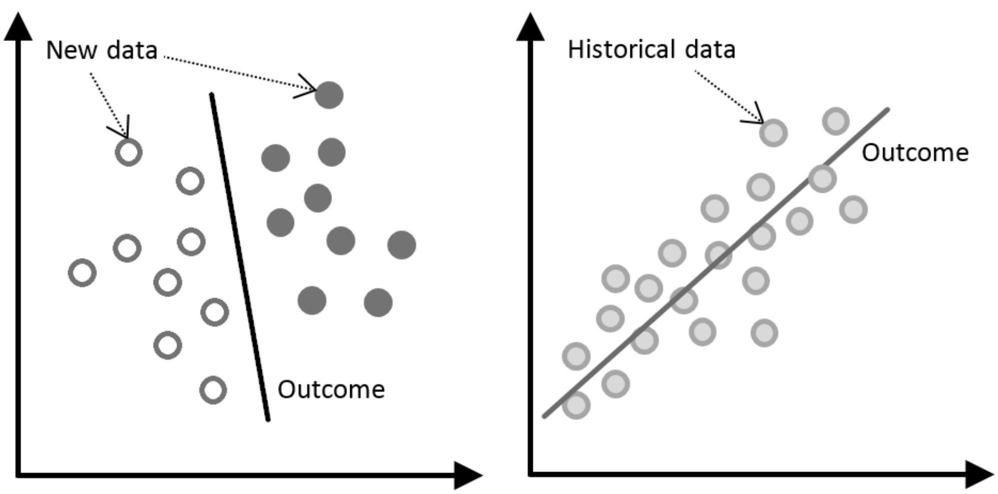
In 1959, Arthur Samuel coined the term “Machine Learning”, as *“the field of study that gives computers the ability to learn without being explicitly programmed.”* However, ML goes beyond simply learning or extracting knowledge, to utilizing and improving knowledge over time and

[3]Traffic engineering dimension that involves mapping and controlling the traffic flows in the network for optimizing the routing function to steer traffic in the most efficient way [27].

[4]No common nodes or links except for source and destination

with experience. Broadly, ML can be divided into three categories, based on how the *learning* is achieved [35, 36]: *supervised*, *unsupervised*, and *reinforcement*. Supervised learning uses labeled training datasets to create models that map inputs to their corresponding outputs. Unsupervised learning uses unlabeled training datasets to create models that find dominating structure or patterns in the data. Reinforcement learning uses the feedback from the environment to learn the correct sequence of actions that maximizes a cumulative reward.

Considering the problem stated in this proposal, supervised learning is the most likely ML category to use because it allows solving *classification* and *regression* problems that pertain to predicting discrete and continuous valued outputs, respectively (*c.f.*, Figure 2). For example, a classification problem can be to identify mice and elephant flows. Whereas, a regression problem can be to predict the size of each flow. Supervised learning includes ML techniques such as linear and logistic regression (*i.e.*, linear models), neural networks, k-nearest neighbors, decision trees, and support vector machines. In addition, supervised learning allows applying ensemble methods to select and combine multiple variations of the above ML techniques to achieve better results at little extra effort.



(a) Classification (b) Regression

Figure 2. Problem categories that benefit from supervised learning

2.2 Multipath routing for load-balancing in data center networks

Traffic engineering in DCNs is a very broad topic that includes multiple solutions for addressing different requirements about optimizing the network performance. Therefore, this proposal explores multipath routing for load-balancing in DCNs by using a systematic study to review the literature. This systematic review study defined an electronic search procedure in digital libraries (*e.g.*, IEEE, ACM, Science Direct, Springer) for works published since 2011.

Prunning based on citations and venue. Other seminal works were added based on the references and citations of the reviewed works.

The systematic review study relied on an electronic search procedure in digital libraries that defined different search queries constructed from the following keywords: "multipath routing", "load-balancing", "data center", and "software-defined networking". Includes works published since 2011.

From a general perspective, the systematic review study shows that multipath routing in DCNs has been traditionally implemented as a distributed solution. In a distributed solution, each switch learns the network status, based on exchanged messages, and constructs a table for packet forwarding. Several research efforts addressed load-balancing by following such distributed solution. Others have introduced centralized solutions by leveraging the potential of SDN. In a centralized solution, a controller has a global view of the network status and instructs the switches for packet forwarding. The centralized solution has been adopted by few recent investigations that propose the utilization of ML techniques for enabling traffic prediction in DCNs.

This proposal focuses on centralized solutions.

more dynamic and network-aware load-balancing solutions.

propose centralized load-balancing solutions. The latter , including those that apply ML for ... .

This section explores the status of multipath routing for load-balancing in DCNs by using a systematic study to review the related works from the literature. This systematic review study

The purpose of the systematic mapping study described in this protocol is to explore—at a high level of granularity—the status of Traffic Engineering in SDN-based networks by using a secondary study to review the related work.

It is worth noting that Traffic Engineering in SDN is a very broad topic that includes multiple solutions for addressing different requirements about optimizing network performance. Therefore, this systematic mapping will enable to identify suitable areas for conducting a more specific systematic literature review and future primary studies.

This systematic mapping study will include papers that propose traffic engineering solutions for managing SDN-based networks. Our intention is to review articles published since 2006 considering that the work of Casado et al. by this time [**? ?**] represented the starting point for defining the current SDN architecture. Nevertheless, it is important to mention that there are some important articles before 2006 that provided significant contributions in the road to SDN [**?**].

This section reviews some of these works from the literature.

2.2.1 Distributed Solutions

ECMP is the most prevalent multipath routing solution, which generally uses a hash function in every switch to assign each incoming flow to an output port [5]. This hash function uses the header information of the flow (usually, the five-tuple) to compute a result (i.e., output port). Nevertheless, traffic characteristics in DCN causes hot-spots in ECMP-based DCN, i.e., some links are highly utilized while others are underutilized.

Multi-Path TCP (MPTCP) [30] is very effective for load-balancing because it offers great parallelism. MPTCP may use a middlebox for traffic from the outside of the DCN (*e.g.*, Internet).

However, since MPTCP uses sub-flows, in case of high traffic burstiness, it increases congestion and generates more packet losses due to overflow of buffers.

Random Packet Spraying [37] forward packets through different shortest paths. Nevertheless, it can cause potential packet reordering, confusing the TCP congestion control. It is important to mention that this approach relies on the symmetry of DCNs.

LocalFlow [38] uses a spatial flow splitting based on TCP sequence numbers. This work enables local congestion-awareness. However, in case of asymmetry caused by failures, LocalFlow performs worse than ECMP.

CONGA [39] is based on flowlets [40], that is a burst of packets from a flow separated by enough time gaps. CONGA enables global congestion-awareness. Nevertheless, the switch hardware implementation does not exist for CONGA, as well as for LocalFlow. Unfortunately, a software implementation produces sub-optimal results.

2.2.2 SDN-based Solutions

Hedera [19] carries out a periodic polling of edge switches to collect flow statistics and detect elephant flows. When an elephant flow is detected, the controller calculates and installs a path for the elephant flow. Otherwise, it routes the rest of flows (*e.g.*, mice flows) using ECMP. Hedera performs slow for variable traffic; it requires at least 100ms polling for approaching performance of distributed solutions, which increases traffic overhead.

Mahout [17] also detects elephant flows but at end hosts. When an elephant flow is detected, the end host marks the subsequent packets of the detected elephant flow. The controller computes and installs a path for the marked packets. As in Hedera, the rest of flows (*e.g.*, mice flows) are routed using ECMP. Mahout uses a coarse granularity for classifying the flow sizes: mice and elephants. However, elephants flows can vary from tens of KB to tens of MB, causing the hot-spot problem to persist.

MicroTE [18], as Mahout, also performs elephant detection at end hosts. MicroTE uses a short term and partial predictability of the traffic matrix. The controller compute paths for predictable traffic, while the unpredictable traffic is routed in a weighted form of ECMP. Besides having the same problem as Mahout, the bursty nature of the traffic in DCN makes its traffic matrix prediction questionable.

Devoflow [41] reduces the interaction between the controller and the switches by using OpenFlow wildcards, triggering for detected elephant flows. Devoflow requires small sFlow sampling, however, the implementation is in hardware and gives little adaptability to flow dynamics.

2.2.3 Traffic Engineering using Machine Learning

[42] presents an online flow size prediction that uses ML techniques (e.g., NN and Bayesian Networks) to identify elephant flows in real datasets. As metrics, they employ True Positive Rate (TPR) and True Negative Rate (TNR), finding that the Gaussian Process Regression is the more robust method. This approach present two problems from the above SDN-based solution. First, it assumes that the data is centralized, however, as observed in Hedera, collecting

data from the switches to a centralized controller increases traffic overhead. In this case the collection of data exponentially increments the traffic overhead because this approach requires per-packet headers; in addition, it would present delay due to data transmission and processing. The second problem is related to Mahout and MicroTE, since this approach employs the same coarse granularity for predicting flow sizes: mice and elephants; nevertheless, flow sizes can vary from tens of Bytes to tens of MB, making the hot-spot problem more difficult than a simple two-class classification.

Although [43] is not for intra-DC routing, it is worth mentioning because combines wavelet transform with NN for predicting Inter-DC traffic (*i.e.*, traffic between Data Centers), which is dominated by a few large applications producing elephant flows. This prediction can be used for scheduling traffic to reduce peak bandwidth in communication links (many ISPs charge for bandwidth by the peak bandwidth that a costumer uses). This approach was tested at Baidu, one of the largest Internet company in China, reducing the prediction errors by 5% to 10% and the peak bandwidth for about 9% on average.

Table. Modified Hosts. Granularity of flow size. Centralized data. Machine Learning.

3 Hypothesis

To address the research question stated in Section 1, this thesis proposal raises the following hypothesis: **Using ML techniques for finer-granularity prediction of flow characteristics and SDN for dynamic control of traffic routing would allow building an effective load-balancing solution for DCNs.**

4 Research Contributions

The present thesis proposal aims to achieve the following contributions:

- A high-level framework that integrates the concepts of ML and SDN into the design of a DCN topology—potentially, switch-centric—for enabling the development of an effective load-balancing solution.
- A mechanism that uses one or a combination of ML techniques for fine-granularity prediction of flow characteristics—potentially, size and time length—in a DCN with the appropriate accuracy and performance requirements.
- A multipath routing mechanism based on SDN that uses the flow characteristics predicted by the ML-based mechanism for minimizing the maximum load of links—optimizing load-balancing—in a DCN aiming to provide high throughput and low delay while maintaining efficient use of resources.

5 Objectives

5.1 General Objective

To develop a framework based on ML and SDN technologies for effective load-balancing in DCNs.

5.2 Specific Objectives

- To design a conceptual solution that incorporates the capabilities of ML and SDN for optimizing load-balancing in DCNs.
- To construct and evaluate[5] a mechanism based on ML that predicts, in a fine-granularity way, flow characteristics in DCNs.
- To construct and evaluate[6] a multipath routing mechanism based on SDN that uses predicted flow characteristics for optimizing the routing function in DCNs.

6 Methodology

Figure 3 depicts the phases of the scientific research process that will guide the development of this thesis: Problem Statement, Hypothesis Construction, Experimentation, Conclusion, and Publication. Problem Statement, for identifying and establishing the research question. Hypothesis Construction, for formulating the hypothesis and the associated fundamental questions. In addition, this phase aims to define and carry out the conceptual and technological approaches. Experimentation, for testing the hypothesis and analyzing the evaluation results. Conclusion, for outlining conclusions and future works. Note that Hypothesis Construction has feedback from Experimentation and Conclusion. Publication, for submitting and publishing papers for renowned conferences and journals. The writing of the dissertation document also belongs to this last phase.

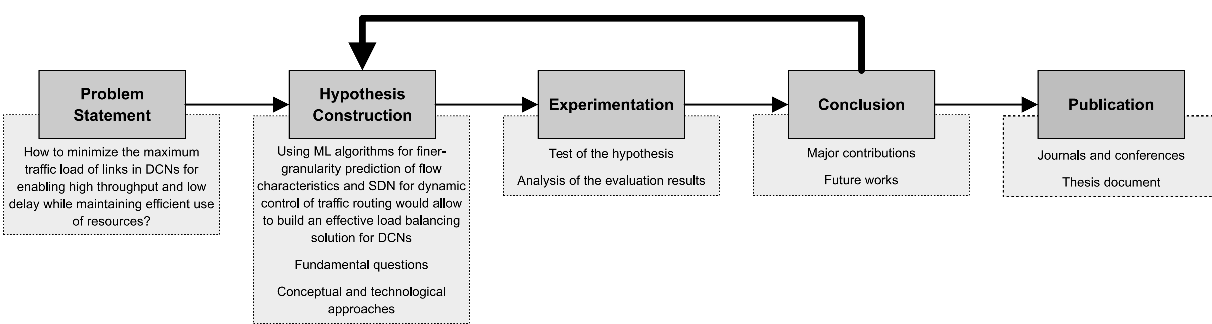


Figure 3. Thesis phases

[5] In terms of accuracy (*e.g.*, true/false positives/negatives) and processing requirements (*e.g.*, training data, training time, run-time).

[6] In terms of traffic performance (*e.g.*, throughput, delay) and resource utilization (*e.g.*, links load).

7 Budget

According to the reference criteria for budget preparation in thesis proposals established by the FIET Research Committee from the University of Cauca and knowing that the value of the point for 2017 is COP$ 12.939, Table 1 presents the budget for the proposed thesis project.

Item	Sources				Total
	Student	**FIET**	**Colciencias**	**UWaterloo***	
Human resources	24,842,880.00	9,937,152.00	72,000,000.00	4,456,940.44	111,236,972.44
Hardware resources	1,494,646.15	11,981,538.46	0.00	1,015,384.62	14,491,569.23
Software resources	81,000.00	21,000.00	0.00	60,000.00	162,000.00
Library resources	0.00	5,250,000.00	0.00	12,750,000.00	18,000,000.00
Visits and travels	2,000,000.00	4,000,000.00	4,000,000.00	11,000,000.00	21,000,000.00
Publications	0.00	3,000,000.00	0.00	0.00	3,000,000.00
Other resources	899,413.39	2,810,666.85	0.00	1,049,315.63	4,759,395.87
Subtotal	*29,317,939.55*	*37,000,357.31*	*76,000,000.00*	*30,331,640.68*	*172,649,937.54*
A.U.I.	0.00	25,897,490.63	0.00	8,632,496.88	34,529,987.51
Total	**29,317,939.55**	**62,897,847.95**	**76,000,000.00**	**38,964,137.56**	**207,179,925.05**

** Includes the funding from the Government of Canada* *All the values are in COP$*

Table 1. Budget of the thesis project

8 Submission Terms

By the end of this thesis project it must be delivered the following items.

- A dissertation document that contains the state-of-the-art, the conceptual and technological solutions, and the experimental results.
- Additional documents that complement the dissertation document.
- A compact disc that gathers all the information generated during the development of this thesis project, including the dissertation document, source code and executable files, related documentation, among others.
- One (1) paper accepted and one (1) submitted in conferences with h-index greater than 20.
- One (1) paper accepted and one (1) submitted in journals from the first quartile (Q1) of the SCImago Journal & Country Rank (SJR).

9 References

[1] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, “Data center network virtualization: A survey,” *IEEE Communications Surveys Tutorials*, vol. 15, pp. 909–928, Second 2013. 1, 2

[2] R. K. Singh, N. S. Chaudhari, and K. Saxena, “Load balancing in IP/MPLS networks: A survey,” *Communications and Network*, vol. 4, pp. 151–156, May 2012. 1, 3

[3] N. Wang, K. H. Ho, G. Pavlou, and M. Howarth, “An overview of routing optimization for internet traffic engineering,” *IEEE Communications Surveys Tutorials*, vol. 10, pp. 36–56, First 2008. 1, 3

[4] S. K. Singh, T. Das, and A. Jukan, “A survey on internet multipath routing and provisioning,” *IEEE Communications Surveys Tutorials*, vol. 17, pp. 2157–2175, Fourthquarter 2015. 1, 3

[5] M. Chiesa, G. Kindler, and M. Schapira, “Traffic engineering with equal-cost-multipath: An algorithmic perspective,” *IEEE/ACM Transactions on Networking*, vol. 25, pp. 779–792, April 2017. 1, 5

[6] G. Detal, C. Paasch, S. van der Linden, P. Mérindol, G. Avoine, and O. Bonaventure, “Revisiting flow-based load balancing: Stateless path selection in data center networks,” *Computer Networks*, vol. 57, pp. 1204–1216, Apr. 2013. 1

[7] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, “Inside the social network’s (datacenter) network,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM ’15, (New York, NY, USA), pp. 123–137, ACM, 2015. 1, 2

[8] T. Benson, A. Akella, and D. A. Maltz, “Network traffic characteristics of data centers in the wild,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC ’10, (New York, NY, USA), pp. 267–280, ACM, 2010. 1

[9] T. Benson, A. Anand, A. Akella, and M. Zhang, “Understanding data center traffic characteristics,” *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 92–99, Jan. 2010. 1

[10] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, “The nature of data center traffic: Measurements & analysis,” in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, IMC ’09, (New York, NY, USA), pp. 202–208, ACM, 2009. 1

[11] A. Andreyev, “Introducing data center fabric, the next-generation Facebook data center network,” November 2014. [Online]. Available: https://www.mirantis.com/blog/whats-opendaylight/. 1

[12] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, “HULA: Scalable Load Balancing Using Programmable Data Planes,” in *Proceedings of the Symposium on SDN Research*, SOSR ’16, (New York, NY, USA), pp. 10:1–10:12, ACM, 2016. 1

[13] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, “Fastpass: A Centralized "Zero-queue" Datacenter Network,” in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM ’14, (New York, NY, USA), pp. 307–318, ACM, 2014. 1

[14] D. Arora, T. Benson, and J. Rexford, “ProActive Routing in Scalable Data Centers with PARIS,” in *Proceedings of the 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing*, DCC ’14, (New York, NY, USA), pp. 5–10, ACM, 2014. 1

[15] R. Trestian, G. M. Muntean, and K. Katrinis, “MiceTrap: Scalable traffic engineering of datacenter mice flows using OpenFlow,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pp. 904–907, May 2013. 1

[16] R. Wang, D. Butnariu, and J. Rexford, "OpenFlow-based Server Load Balancing Gone Wild," in *Proceedings of the 11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, Hot-ICE'11, (Berkeley, CA, USA), pp. 12–12, USENIX Association, 2011. 1

[17] A. R. Curtis, W. Kim, and P. Yalagandula, "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection," in *2011 Proceedings IEEE INFOCOM*, pp. 1629–1637. 1, 6

[18] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine grained traffic engineering for data centers," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pp. 8:1–8:12, ACM. 1, 6

[19] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pp. 19–19, USENIX Association. 1, 6

[20] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (dcn): Infrastructure and operations," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 640–656, Firstquarter 2017. 2

[21] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, and A. V. Vasilakos, "Survey on routing in data centers: insights and future directions," *IEEE Network*, vol. 25, pp. 6–10, July 2011. 2

[22] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, "A cost comparison of datacenter network architectures," in *Proceedings of the 6th International COnference*, Co-NEXT '10, (New York, NY, USA), pp. 16:1–16:12, ACM, 2010. 2

[23] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, A. Kanagala, J. Provost, J. Simmons, E. Tanda, J. Wanderer, U. Hölzle, S. Stuart, and A. Vahdat, "Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, (New York, NY, USA), pp. 183–197, ACM, 2015. 2

[24] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, (New York, NY, USA), pp. 51–62, ACM, 2009. 2

[25] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, (New York, NY, USA), pp. 63–74, ACM, 2008. 2

[26] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, Internet Engineering Task Force, 2002. 3

[27] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in sdn-openflow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014. 3

[28] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Internet Engineering Task Force, 2000. 3

[29] R. Zhang-Shen and N. McKeown, "Designing a predictable internet backbone with valiant load-balancing," in *Proceedings of the 13th International Conference on Quality of Service*, IWQoS'05, (Berlin, Heidelberg), pp. 178–192, Springer-Verlag, 2005. 3

[30] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath tcp," in *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, (New York, NY, USA), pp. 266–277, ACM, 2011. 3, 5

[31] N. Feamster, J. Rexford, and E. Zegura, "The road to sdn: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–98, Apr. 2014. 3

[32] P. Lin, J. Bi, H. Hu, T. Feng, and X. Jiang, "A quick survey on selected approaches for preparing programmable networks," in *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC '11, (New York, NY, USA), pp. 160–163, ACM, 2011. 3

[33] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, pp. 114–119, February 2013. 3

[34] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 69–74, Mar. 2008. 3

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016. 4

[36] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., 2010. 4

[37] A. Dixit, P. Prakash, Y. C. Hu, and R. R. Kompella, "On the impact of packet spraying in data center networks," in *2013 Proceedings IEEE INFOCOM*, pp. 2130–2138. 6

[38] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pp. 151–162, ACM. 6

[39] M. Alizadeh, T. Edsall, S. Dharmapurikar, R. Vaidyanathan, K. Chu, A. Fingerhut, V. T. Lam, F. Matus, R. Pan, N. Yadav, and G. Varghese, "CONGA: Distributed congestion-aware load balancing for datacenters," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pp. 503–514, ACM. 6

[40] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 51–62. 6

[41] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks," in *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pp. 254–265, ACM. 6

[42] P. Poupart, Z. Chen, P. Jaini, F. Fung, H. Susanto, Y. Geng, L. Chen, K. Chen, and H. Jin, "Online flow size prediction for improved network routing," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–6. 6

[43] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting inter-data-center network traffic using elephant flow and sublink information," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 782–792. 7