Uncovering Secrets of Microbursts in Datacenter Network Traffic

Mohammad Hosseini[‡] Sina Darabi[§] Mohammad Nakhjiri[‡] Patrick Eugster[§]
[‡]*Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran*
[§]*The Faculty of Informatics, Università della Svizzera italiana (USI), Lugano, Switzerland*
m-hosseini@sbu.ac.ir, {darabs, eugstp}@usi.ch, m.nakhjiri@mail.sbu.ac.ir

***Abstract*—Designing efficient methods and policies for mitigating microbursts requires a thorough understanding of microburst characteristics and behaviors. However, the lack of detailed studies on microburst characteristics and comprehensive tools for measuring and analyzing them has been a significant challenge for researchers in this field. We introduce BurstVision, a tool that extracts various characteristics of microbursts from traffic traces. Using BurstVision, we analyze several traffic traces from various cloud datacenter applications and report on the diverse characteristics of microbursts observed. Our analysis reveals that microburst characteristics significantly vary across applications. Moreover, we discuss how these varying characteristics can influence the effectiveness of different microburst mitigation solutions. Our findings highlight the importance of considering the specific type and characteristics of microbursts in traffic when adopting a microburst mitigation solution.**

***Index Terms*—Network traffic, Microburst, Datacenter**

I. Introduction

Understanding network traffic and its behavior has been crucial since the emergence of computer networks and the Internet. Many studies have been conducted in this area to gain insights into traffic characteristics. In light of these studies, more efficient algorithms and architectures have been designed for computer networks and their devices. As computer networks continue to evolve and become more complex, understanding traffic behavior becomes increasingly important to ensure that networked systems can continue to operate effectively and efficiently.

A critical aspect of network traffic is the phenomenon of *bursts* — sudden, short-lived spikes in data transmission. Bursty traffic can create bottlenecks and congestion, significantly degrading network performance. To mitigate these effects, researchers have developed various techniques, including traffic shaping, pacing, buffering, and prioritization. However, the discovery of *microbursts* in datacenter networks has opened new avenues of research in this field. Groundbreaking studies by Benson *et al.* [1], [2] first revealed that datacenter links with low average utilization can incur losses, indicating that these links experience momentary bursts. These microbursts proved too short-lived for traditional traffic engineering approaches to detect or address effectively. Notably, they observed that most losses occur during these short-lived bursts. Subsequent research by Roy *et al.* [3] at Facebook's datacenters corroborated Benson's findings and further demonstrated that even heavy flows can exhibit internal burstiness.

Since then, many researchers have focused on this kind of bursts, i.e., microbursts. Typically, periods of high utilization lasting less than 1 ms are considered microbursts. Measurements by Zhang *et al.* [4] showed that 90% of bursts last less than 200 µs. Microbursts originate from various sources, including application behaviors, TCP artifacts, NIC offloading features and packet coalescing, and OS protocol stack processing [5]–[7]. Several solutions have been proposed in the literature to mitigate microbursts, including switch-centric solutions (absorbing micro-bursts in the switch buffer, flow table management) [8]–[10], network-centric techniques (load balancing, traffic deflecting, network architecture) [11]–[16], and host-centric techniques (feedback-based congestion control protocols, adding jitter, credit-based transport protocols) [17]–[22]. Some of the solutions deal with bursty flows and aim to detect and manage these flows (we call them *flow-based* solutions), while others do not deal with flows but instead treat all traffic packets as a single stream and try to mitigate its bursts (we call them *traffic-based* solutions). However, due to the diverse and ever-changing root causes and timing features of microbursts, none of the solutions can guarantee to alleviate the negative impacts of microbursts in all situations and applications. A recent study has demonstrated that existing countermeasures to microbursts have significant limitations and do not yield satisfactory results [23], underscoring the need for continued research in microburst mitigation.

The foundation of microburst research lies in accurately measuring, characterizing, and analyzing these phenomena. However, a significant gap exists: there is no comprehensive tool for measuring microbursts and extracting their features. Current microburst mitigation solutions are largely based on characteristics derived from a limited set of ad-hoc measurements in specific datacenters. This approach overlooks the potential variability of microburst characteristics across different applications and network environments. Furthermore, many aspects of microbursts remain unexplored. We posit that this insufficient understanding of microburst dynamics is the primary factor contributing to the limited efficacy of existing mitigation solutions.

In this paper, we introduce *BurstVision*, a comprehensive tool for measuring microbursts and extracting their characteristics. BurstVision processes PCAP files to generate detailed reports on microburst occurrences within the captured traffic. These reports include a wide range of metrics and distribu-

978-3-903176-66-9 ©2024 IFIP

tions, such as the number of bursty flows, burst frequency per flow, burst duration, and inter-burst intervals. A key innovation of BurstVision is its dual-mode analysis capability. It can perform *traffic-oriented* analysis, treating all packets as a single stream for burst statistics, as well as *flow-oriented* analysis, processing individual flows to collect per-flow burst statistics. This flexibility is crucial, as these two approaches can reveal distinct and complementary characteristics of microbursts.

Using BurstVision, we analyzed traffic traces from four distinct cloud applications within a cloud service provider's datacenter. Our analyses demonstrate that microburst characteristics vary significantly across different applications. We present, for the first time in the literature, a discussion on how these varying characteristics can impact the performance of different microburst mitigation solutions. Additionally, we discuss how this analysis can help us adopt a more efficient microburst mitigation approach, especially when it comes to choosing between flow-based and traffic-based solutions. Our results show considerable variation in microburst frequency, length, intensity, and intervals across different traces, highlighting that a one-size-fits-all solution is unlikely to be optimal for all datacenter applications. Finally, we note that the source code of BurstVision is freely available[1], and researchers can extend its features to suit their specific requirements.

The rest of this paper is organized as follows: In Section II, we review existing works on burst measurement. Section III presents an overview of BurstVision. In Section IV, our microburst analyses are presented. Finally, Section V concludes the paper and discusses future work.

II. BACKGROUND AND RELATED WORK

Most previous studies reporting statistics on microbursts have been conducted in the field of traffic monitoring. Online traffic monitoring systems are used to identify and react to any anomalies or performance issues in datacenters. The design of these systems is traditionally based on coarse-grained SNMP counters or traffic sampling. However, coarse-grained measurements fail to detect microbursts. Hence, several studies have been carried out to provide high-resolution monitoring systems that can capture and monitor microbursts. These studies adopted various approaches, such as fine-grained sampling of switch statistics [4], programmable data planes [19], [24], [25], and host-based monitoring [23], [26].

While these studies have made significant strides in capturing and monitoring microbursts, they fall short in providing comprehensive reports and statistics on microburst characteristics. Many aspects of microbursts remain unexplored, particularly from a flow-oriented perspective. Existing studies primarily adopt a traffic-oriented approach, analyzing bursts by considering all traffic packets as a single stream. This method may overlook variations in burst characteristics within individual flows. Furthermore, each study typically focuses on a specific datacenter hosting a particular application, limiting the generalizability of findings. To gain a holistic understanding of microbursts, it is crucial to evaluate diverse datacenters and applications. However, deploying existing microburst monitoring solutions across various datacenters poses significant challenges. These solutions are often resource-intensive and costly, with programmable data plane-based approaches incompatible with regular switches, and host-based solutions requiring substantial modifications to network stacks. Additionally, many researchers lack direct access to datacenters or may prefer to analyze pre-recorded traffic traces.

Detailed microburst characteristics can be extracted through offline processing of traffic traces. To our knowledge, the only available tool for processing a traffic trace and extracting its burst statistics is IPFIXPROBE [27]. It has been developed by Tropkova *et al.* [28] to evaluate their proposed HTTPS classifier (which relies on analyzing bursts within flows). IPFIXPROBE analyzes each flow individually and extracts its burst details, including the number of bursts and their size. However, IPFIXPROBE has several shortcomings. Its millisecond-level resolution proves insufficient for detecting microbursts. It only reports the first 10 bursts of each flow. Its report is not comprehensive, and it does not analyze all characteristics of microbursts, such as inter-burst duration. Finally, it does not offer traffic-oriented processing, focusing solely on flow-level analysis. These shortcomings highlight the need for a more robust and versatile tool capable of providing a comprehensive analysis of microburst phenomena across both flow-oriented and traffic-oriented perspectives.

III. BurstVision

BurstVision takes a PCAP file, a minimum burst ratio, and the analysis mode as inputs. The burst ratio of a burst is the burst's peak rate divided by the average rate of the traffic within a specified time window centered on a burst. The user can adjust the length of this time window as another input parameter to the tool. The minimum burst ratio is used as the threshold for detecting bursts. The analysis mode can be either traffic-oriented or flow-oriented. In the traffic-oriented mode, all packets of the input trace are considered a single stream, while in the flow-oriented mode, each flow of the trace is analyzed separately.

TABLE I
Characteristics analyzed and reported by BurstVision

Characteristics Analysis	Analysis Mode	Units
(1) Number of bursts	both	Number
(2) Number of flows	Both	Number
(3) Number of bursty flows	Flow-oriented	Number
(4) Number of heavy flows	Both	Number
(5) Flows contributing to each burst	Traffic-oriented	Number
(6) Length of (all, heavy, bursty) flows	Both	μs
(7) Length of bursts	Both	μs
(8) Number of packets in bursts	Both	Number
(9) Transferred bytes in bursts	Both	Bytes
(10) Average size of packets in each burst	Both	Bytes
(11) Burst ratio of bursts	Both	Ratio
(12) Inter-burst duration	Both	μs
(13) Number of bursts in each flow	Flow-oriented	Number
(14) Number of concurrent bursty flows	Flow-oriented	Number

[1] https://github.com/toorin-lab/BurstVision

Table I represents the characteristics that are analyzed and extracted by BurstVision. BurstVision yields the number of bursts (1), flows (2), bursty flows (3), and heavy flows (4). A flow is considered bursty if it contains at least one burst. Heavy flows are flows that have an average rate above a predefined threshold, which is another input parameter of BurstVision. An important point is that the number of bursts can differ between the two analysis modes because some traffic bursts may result from a sudden increase in the number of flows rather than from a bursty flow. In other words, a burst in traffic can occur due to a significant momentary increase in the number of flows, even when there is no bursty flow at that specific moment. Some computing paradigms, such as MapReduce and partition/aggregate, can generate a high number of flows momentarily. Hence, we included the two analysis modes in BurstVision. While the traffic-oriented mode can identify any bursts in traffic, the flow-oriented mode only detects bursts in flows. When the cause of a burst is an increase in the number of flows, the number of flows contributing to the burst (5) becomes an important characteristic, which is determined by BurstVision. These different analyses help us determine whether a flow-based or traffic-based microburst mitigation solution is more suitable for a specific datacenter application. We elaborate on this point in the next section.

BurstVision also reports the length of flows (6) along with the length of bursts (7) in μs precision. Moreover, BurstVision calculates the number of packets (8), transferred bytes (9), and average packet size (10) in bursts, as well as their burst ratio (11). All these characteristics relate to the size and intensity of bursts and help us to establish the appropriate capacity for burst mitigation solutions. Another characteristic that is related to flow-based microburst mitigation solutions is the number of concurrent bursty flows (14). BurstVision determines this characteristic by counting the number of bursty flows during each burst. Inter-burst duration (12) is the time interval between bursts in either bursty flows or whole traffic. Flows or traffic with a short inter-burst duration may require special precautions and microburst mitigation solutions. BurstVision also calculates the number of bursts in each bursty flow (13). Flows that exhibit a high frequency of bursts may require unique treatment methods to be effectively managed. All these characteristics help us better understand microbursts in a datacenter and adopt an efficient solution for mitigating them.

IV. Results and Analysis

We analyzed the characteristics of microbursts in four traces, each corresponding to a different cloud infrastructure application, obtained from a cloud service provider. The applications include Apache Hadoop (as a distributed file system and MapReduce framework), Apache Cassandra (as a distributed database), Apache Kafka (as a stream-processing platform), and the ELK Stack (as a logging platform).

Our analysis began by extracting the number of bursts from the traces using both traffic-oriented and flow-oriented analyses, with a minimum burst ratio of 5. We observed distinct results for each trace between these two analytical approaches. Fig. 1 illustrates the ratio of bursts detected by traffic-oriented analysis to those detected by flow-oriented analysis. A notable finding is the significantly higher ratio for Hadoop and Cassandra. For these applications, the traffic-oriented analysis identifies more bursts than the flow-oriented analysis. This reveals that while Hadoop and Cassandra experience numerous traffic bursts, they do not generate a substantial number of bursty flows. This suggests that many of their bursts stem from a sudden increase in the number of normal flows. In contrast, Kafka and ELK show little difference in the number of bursts reported by the two analyses. This indicates that their bursts primarily result from momentary increases in the transfer rate of individual flows, i.e., bursty flows. These distinctions have important implications for the efficacy of various microburst mitigation solutions. Flow-based solutions, such as Elixir [10] and MATCP [22], are designed to detect and mitigate bursty flows. Elixir accomplishes this by adding the flows to the hardware table of switches, while MATCP sends immediate ECN signals to the sender of each bursty flow. However, these solutions may prove ineffective for Hadoop and Cassandra, as no considerable number of bursty flows are present to be detected. On the other hand, such solutions could be highly effective for Kafka and ELK, where bursts are primarily caused by bursty flows rather than an increase in the number of normal flows.

Given that bursts in Kafka and ELK are caused by bursty flows, it would be beneficial to have statistics on these flows. We calculated the ratio of bursty flows and heavy flows to the total number of flows. For Kafka and ELK, the bursty flow ratios are 0.02 and 0.11, while the heavy flow ratios are 0.14 and 0.09, respectively. To illustrate the significance of these findings, we revisit the flow-based solution, Elixir. Elixir's key strategy is to allocate a small portion of the switch's hardware table for storing forwarding rules of bursty flows, rather than dedicating the entire table to heavy flows. Our analysis challenges the assumption that bursty flow ratios are always significantly lower than heavy flow ratios. In the case of ELK, for instance, the bursty flow ratio (0.11) slightly exceeds the heavy flow ratio (0.09). This suggests that for scenarios similar to ELK, it may be more effective to allocate a substantial portion of Elixir's forwarding table to bursty flows.

Fig. 2a shows the distribution of burst frequency in bursty

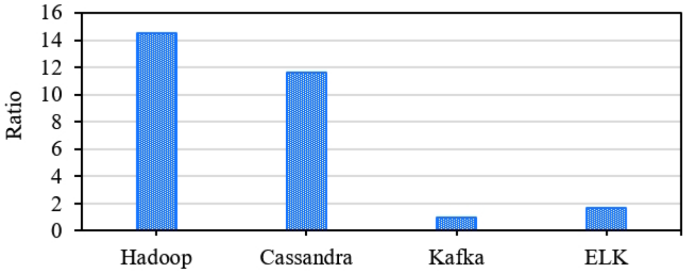
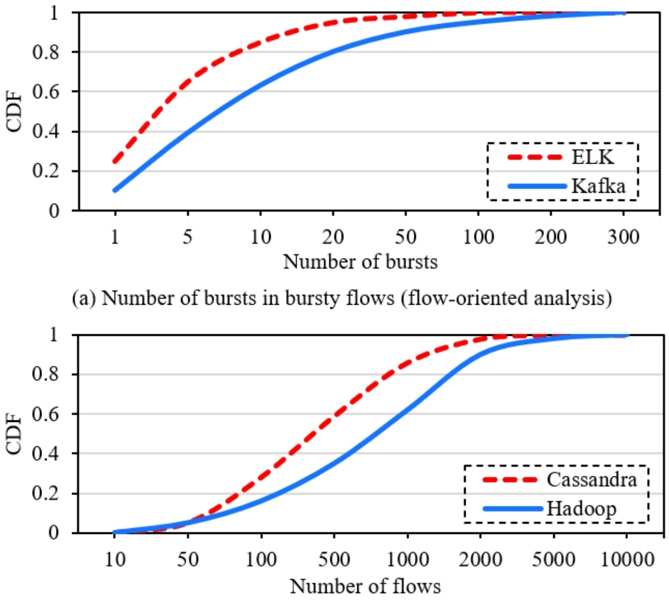


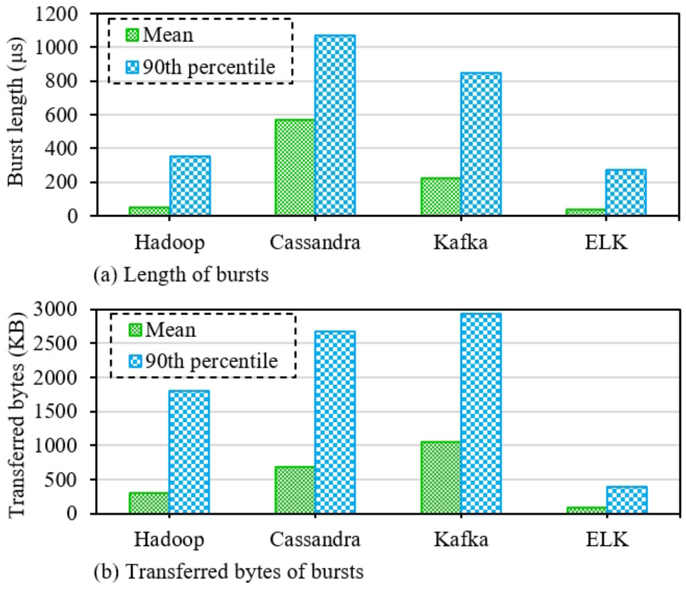
Fig. 1. Ratio of bursts detected by traffic-oriented analysis to bursts detected by flow-oriented analysis.



(a) Number of bursts in bursty flows (flow-oriented analysis)

(b) Contributing flows in each burst (traffic-oriented analysis)

Fig. 2. The role of flows in microbursts.



(a) Length of bursts

(b) Transferred bytes of bursts

Fig. 3. Characteristics of microbursts in each traffic trace.

flows for Kafka and ELK traces. Our analysis reveals that bursty flows in Kafka exhibit a higher number of bursts compared to those in ELK. In other words, Kafka's bursty flows demonstrate more frequent burst repetition. This finding suggests that inserting forwarding rules for bursty flows into the hardware table of switches would yield more significant benefits for Kafka than for ELK. For Hadoop and Cassandra, where their bursts are caused by sudden increases in normal flows, a different analysis should be conducted. Fig. 2b depicts the distribution of flows contributing to each traffic burst. The results indicate that Hadoop bursts typically involve a substantially higher number of contributing flows compared to Cassandra bursts. This characteristic has important implications for microburst mitigation solutions based on load balancing or traffic deflection [11]–[14]. In the event of a burst, these solutions would need to manipulate the routes of a larger number of flows in Hadoop, potentially increasing their processing load significantly.

Fig. 3 show the mean and 90th percentile of length and transferred bytes of bursts. Given our previous identification of burst types in each trace, we present flow-oriented analysis for Kafka and ELK traces, and traffic-oriented analysis for Hadoop and Cassandra traces. The traces exhibit significant variation in both length and traffic volume of bursts, which can substantially impact the effectiveness of various microburst mitigation solutions. For instance, the bursts in ELK, which are caused by bursty flows, are very short. This means that solutions that utilize feedback-based congestion protocols [19], [21], [22] may not be effective for this traffic because burst lengths are shorter than the time required for signaling feedback and reacting to a burst. However, solutions that absorb bursts in buffers [8] might prove more effective for the ELK trace due to the relatively low volume of data transferred during its bursts. Nonetheless, such buffer-based solutions may not work for the trace of Hadoop, which has short bursts but involves a large amount of data being transferred during them.

Fig. 4 presents the mean and 90th percentile of inter-burst durations for each trace. Like other burst characteristics, inter-burst durations vary significantly across different traces. Longer inter-burst periods can diminish the effectiveness of certain microburst countermeasures. For instance, load balancing-based solutions respond to bursts by adjusting traffic distribution based on current link loads. When subsequent bursts occur after prolonged intervals, the efficacy of previous load-balancing actions diminishes, necessitating repeated adjustments. This scenario highlights the challenge of maintaining optimal traffic distribution in networks with highly variable burst patterns.

Fig. 5 shows the distribution of burst ratios across different traces. Notably, Hadoop and ELK traces exhibit higher burst ratios, indicating more intense bursts compared to other traces. This heightened intensity necessitates greater caution when implementing burst mitigation solutions for traffic patterns similar to Hadoop and ELK. For instance, traffic-deflecting solutions must carefully consider the potential consequences of redirecting such intense bursty traffic. Deflecting these high-intensity bursts could inadvertently cause congestion on other links, underscoring the need for more sophisticated and adaptive mitigation strategies. Due to the page limit, we have not included the evaluation of other characteristics such as the number and size of packets in bursts. Nevertheless, we want to emphasize that these characteristics also vary across the traces.

V. Conclusion

In this paper, we first presented BurstVision, a comprehensive tool for extracting the characteristics of microbursts. The

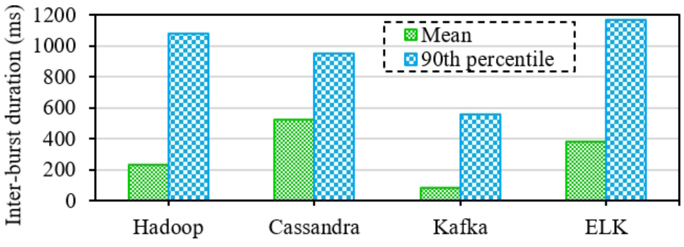


Fig. 4. Inter-burst duration.

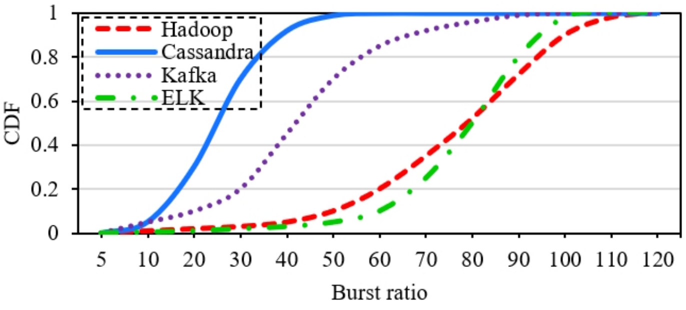


Fig. 5. Burst ratio of microbursts.

tool can help gain a better understanding of burst patterns and behaviors. Then, we analyzed four different traffic traces using BurstVision and showed that the microburst characteristics vary across different applications. Furthermore, we discussed how these differences can impact the performance of various microburst mitigation methods. In future work, we plan to practically evaluate various microburst mitigation solutions for traffic with different microburst characteristics. We will also work on tuning and optimizing the solutions by considering the characteristics of the bursts.

Acknowledgment

Work supported by in parts by Swiss National Science Foundation projects 192121 (FORWARD) and 197353 (BASIS)

References

[1] T. Benson, A. Akella, and D. a. Maltz, “Network traffic characteristics of data centers in the wild,” in 10th ACM Internet Measurement Conference (IMC), pp. 267–280, 2010.

[2] T. Benson, A. Anand, A. Akella, and M. Zhang, “Understanding data center traffic characteristics,” ACM SIGCOMM Computer Communication Review, vol. 40, no. 1, pp. 92–99, 2010.

[3] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, “Inside the social network’s (datacenter) Network,” in 29th ACM SIGCOMM Conference, pp. 123–137, 2015.

[4] Q. Zhang, V. Liu, H. Zeng, and A. Krishnamurthy, “High-resolution measurement of data center microbursts,” in 17th ACM Internet Measurement Conference (IMC), pp. 78–85, 2017.

[5] R. Kapoor, A. C. Snoeren, G. M. Voelker, and G. Porter, “Bullet trains: A study of NIC burst behavior at microsecond timescales,” in 9th ACM conference on Emerging networking experiments and technologies (CoNEXT), pp. 133–138, 2013.

[6] M. Marty, M. de Kruijf, J. Adriaens, C. Alfeld, S. Bauer, C. Contavalli, M. Dalton, N. Dukkipati, W. C. Evans, S. Gribble, N. Kidd, R. Kononov, G. Kumar, C. Mauer, E. Musick, E. Rubow, M. Ryan, K. Springborn, P. Turner, V. Valancius, X. Wang, and A. Vahdat, “Snap: a microkernel approach to host networking,” in 27th ACM Symposium on Operating Systems Principles (SOSP), pp. 399–413, 2019.

[7] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, “Data center TCP (DCTCP),” in 24th ACM SIGCOMM Conference, pp. 63–74, 2010.

[8] D. Shan, W. Jiang, and F. Ren, “Absorbing micro-burst traffic by enhancing dynamic threshold policy of data center switches,” in 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 118–126, 2015.

[9] D. Shan, W. Jiang, and F. Ren, “Analyzing and enhancing dynamic threshold policy of data center switches,” IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 9, pp. 2454–2470, 2017.

[10] Y. Wang, D. Li, J. Wu, S. Hua, “Elixir: a high-performance and low-cost approach to managing hardware/software hybrid flow tables considering flow burstiness,” in 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 535–550, 2022.

[11] S. Ghorbani, Z. Yang, P. B. Godfrey, Y. Ganjali, and A. Firoozshahian, “Drill: Micro load balancing for low-latency data center networks,” in 31st ACM SIGCOMM Conference , pp. 225–238, 2017.

[12] S. Abdous, E. Sharafzadeh, and S. Ghorbani, “Burst-tolerant datacenter networks with Vertigo,” in 17th International Conference on emerging Networking Experiments and Technologies (CoNEXT), pp. 1–15, 2021.

[13] K. Zarifis, R. Miao, M. Calder, E. Katz-Bassett, M. Yu, and J. Padhye, “DIBS: Just-in-time congestion mitigation for data centers,” in 9th European Conference on Computer Systems, pp. 1–14, 2014

[14] X. Shi, L. Wang, F. Zhang, K. Zheng, and Z. Liu, “PABO: Congestion mitigation via packet bounce,” in 2017 IEEE International Conference on Communications (ICC), pp. 1–6, 2017.

[15] S.M. Hosseini, A.H. Jahangir, and S. Daraby, “Session-persistent load balancing for clustered web servers without acting as a reverse-proxy,” in 17th International Conference on Network and Service Management (CNSM), pp. 360-364, 2021.

[16] M. Hosseini, S. Darabi, A.H. Jahangir, A. Movaghar, “Yuz: Improving Performance of Cluster-Based Services by Near-L4 Session-Persistent Load Balancing,” IEEE Transactions on Network and Service Management, pp. 1929-1942, 2023.

[17] M. Handley, C. Raiciu, A. Agache, A. Voinescu, A. W. Moore, G. Antichi, and M. Wójcik, “Re-architecting datacenter networks and stacks for low latency and high performance,” in 31st ACM SIGCOMM Conference, pp. 29–42, 2017.

[18] G. Kumar, N. Dukkipati, K. Jang, H. M. G. Wassel, X. Wu, B. Montazeri, Y. Wang, K. Springborn, C. Alfeld, M. Ryan, and A. Vahdat, “Swift: delay is simple and effective for congestion control in the datacenter,” in 34th ACM SIGCOMM Conference, pp. 514–528, 2020.

[19] D. Shan, F. Ren, P. Cheng, and C. Guo, “Micro-Burst in data centers: observations, analysis, and mitigations,” in 26th IEEE International Conference on Network Protocols (ICNP), pp. 88–98, 2018.

[20] Y. Li, R. Miao, H. H. Liu, Y. Zhuang, F. Feng, L. Tang, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, “HPCC: High precision congestion control,” in 33rd ACM SIGCOMM Conference, pp. 44–58, 2019.

[21] D. Shan and F. Ren, “Improving ECN marking scheme with microburst traffic in data center networks,” in IEEE Conference on Computer Communications (INFOCOM), pp. 1–9, 2017

[22] D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, “Observing and mitigating micro-burst traffic in data center networks,” IEEE/ACM Transactions on Networking, vol. 28, no. 1, pp. 98–111, 2020.

[23] E. Sharafzadeh, S. Abdous, and S. Ghorbani, “Understanding the impact of host networking elements on traffic bursts,” in 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp. 237–254, 2023.

[24] R. Joshi, T. Qu, M. C. Chan, B. Leong, and B. T. Loo, “BurstRadar: practical real-time microburst monitoring for datacenter networks,” in 9th Asia-Pacific Workshop on Systems, pp. 1–8, 2018.

[25] K. Gao, D. Li, and S. Wang, “Bandwidth-efficient microburst measurement in Large-scale datacenter networks,” in 6th Asia-Pacific Workshop on Networking (APNet), pp. 1–7, 2022.

[26] E. Ghabashneh, Y. Zhao, C. Lumezanu, N. Spring, S. Sundaresan, and S. Rao, “A microscopic view of bursts, buffer contention, and loss in data centers,” in 22nd ACM Internet Measurement Conference (IMC), pp. 567–580, 2022.

[27] “ipfixprobe - IPFIX flow exporter.” [Online]. Available: https://github.com/CESNET/ipfixprobe.

[28] Z. Tropkova, K. Hynek, and T. Cejka, “Novel HTTPS classifier driven by packet bursts, flows, and machine learning,” in 17th International Conference on Network and Service Management (CNSM), pp. 345–349, 2021.