Effectiveness of Quality of Service Techniques for System Performance

Manikanta Nagulapalli

Abstract

Data center systems are fundamental to advancing modern technological breakthroughs. To achieve unparalleled computing power, hundreds of thousands of computing nodes are organized into a vast computing pool. Within these systems, computing resources are interconnected through a local high-speed network, the performance of which is crucial for system stability and reliability. When an application is executed, various issues such as link failures, network contention, and inefficient routing can degrade performance, leading to delays in network packet delivery and reduced link bandwidth. Quality of Service (QoS) has been demonstrated as a straightforward and effective method for enhancing system performance. In this project, we assess the effectiveness of various QoS solutions.

1 Introduction

The proliferation of digital technologies has necessitated a robust infrastructure to manage, store, and process vast amounts of data, positioning data centers as critical nodes within the global information ecosystem [1, 2]. These facilities, which house an array of networked computers and storage systems, are pivotal in ensuring that data remains accessible, secure, and efficiently managed. This introduction delineates the relationship between data center architectures, network configurations, system performance, and the role of quality of service (QoS) techniques in maintaining network stability, ultimately ensuring an optimal service delivery [3, 4, 5, 6].

Data centers are engineered to support high-demand applications across various sectors, including finance, healthcare, and telecommunications. The architecture of a data center is designed to maximize reliability and efficiency, incorporating state-of-the-art physical and cyber security measures, environmental controls, and redundant power sources. These facilities are also equipped with advanced networking hardware to support high-speed data transmission and seamless connectivity across geographic and virtual boundaries.

The backbone of a data center's operational capability lies in its network configuration, which is meticulously designed to support scalable, secure, and ultra-fast data exchange [7, 8, 9]. Network management in data centers involves the deployment of sophisticated routing and switching protocols that handle incoming and outgoing data traffic [10, 11, 12]. Effective network configuration ensures not only the swift flow of data but also addresses potential bottlenecks, thereby reducing latency and enhancing the overall user experience [13, 14].

System performance in data centers is a multi-faceted domain that involves monitoring and optimizing the efficiency of servers, storage units, and network devices. Key performance indicators (KPIs) such as server uptime, latency, throughput, and error rates are continuously assessed to gauge the health and efficiency of the data center operations [15, 16]. Performance metrics are crucial for operational management, providing insights that guide maintenance, upgrades, and scalability decisions.

The stability of a data center is paramount, directly impacting the reliability of services offered to end-users. Stability involves ensuring operational continuity even in the face of hardware failures, cyber-attacks, or power disruptions [17, 18, 19]. Techniques such as failover systems, redundant hardware, and disaster recovery plans are integral to maintaining stability. Additionally, environmental controls and regular maintenance schedules prevent equipment failure and ensure consistent performance.

Quality of Service (QoS) techniques are essential tools in the network administrator's arsenal, designed to prioritize network traffic and ensure the stability of data flows, particularly in mixed-traffic environments. QoS involves traffic shaping, bandwidth allocation, and prioritization strategies that safeguard critical data transmissions against delays and disruptions [20, 21, 22, 23]. These techniques are crucial in environments where network resources must be judiciously allocated among competing demands, such as streaming media, file transfers, and real-time communications. The integration of advanced architectural designs, meticulous network configurations, continuous performance monitoring, and strategic QoS implementations are fundamental to maintaining the stability and efficiency of data centers. These elements collectively ensure that data centers can meet the demands of an increasingly data-driven world, providing the backbone for numerous critical services across various industries [24, 7]. This report will explore these components and analyze impact on the quality of service in data center environments.

2 Background

2.1 Data center network and Fat-tree topology

Data center networks (DCNs) are complex, scalable, and highly configurable structures designed to accommodate the dynamic needs of various applications and services [25, 26]. The primary function of a DCN is to connect physical and virtual resources within the data center and to external networks efficiently [27, 28, 29]. Traditional network designs often relied on a tiered architecture consisting of the core, aggregation, and access layers, each serving a distinct role in handling data traffic [30, 31]. The core layer is the topmost layer of the network hierarchy that is responsible for fast and reliable routing between different aggregation switches and to external networks. It handles the highest level of data traffic, requiring robust, high-capacity switches. The aggregation layer, also known as the distribution layer, connects the core layer to the access layer and often incorporates security and traffic management functions to optimize data flows. Access layer is closest to the server resources, connecting end-device nodes to the network and directing traffic to the aggregation layer.

The fat-tree topology, a prevalent architecture in modern data centers, is designed to overcome the limitations of traditional hierarchical designs by providing high bandwidth

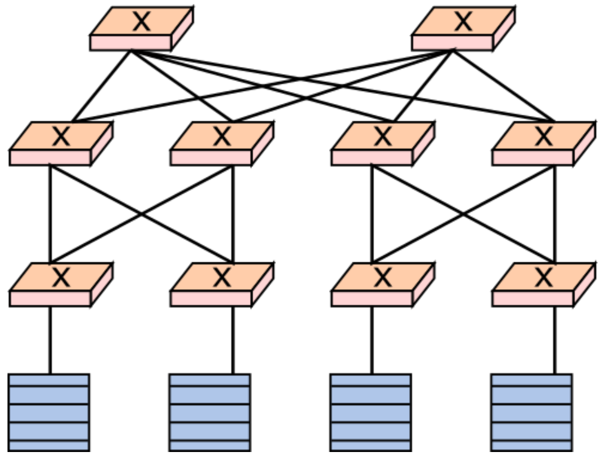

Figure 1: A three-level fat-tree system.

and scalability [32, 33]. As depicted in Figure 1, Fat-tree topology is characterized by a multi-rooted tree structure that interconnects various elements within the network through a series of leaf (edge) and spine (core) switches. In a fat-tree configuration, the network is organized in layers of switches, where the number of switches increases as one moves down from the spine to the leaf switches. This arrangement ensures that bandwidth expands as data moves closer to the servers, significantly reducing bottlenecks that typically occur in hierarchical networks. One of the defining features of the fat-tree topology is its inherent redundancy at multiple layers. Multiple paths exist between any two points in the network, providing alternate routes for data transmission in case of a link or switch failure, thereby enhancing the overall reliability of the network [34, 35, 36, 37]. The fat-tree architecture naturally supports load balancing through its multiple paths, distributing the traffic evenly across the network and optimizing resource utilization [38, 39]. This is particularly important in environments with fluctuating traffic patterns and diverse workload demands.

The fat-tree topology offers several advantages for data center networks. It can seamlessly accommodate growth in network size and traffic, making it suitable for large-scale data centers. Enhanced bandwidth and multiple routing paths help in managing high-performance applications and large volumes of data traffic without degradation in service quality. Fat-tree topology also supports various data center architectures and can be easily adapted for different service requirements and future technologies. In short, the network structure of data centers, particularly through the adoption of fat-tree topology, plays a crucial role in the operational efficiency and scalability of these facilities. By providing robust connectivity, fault tolerance, and flexible data traffic management, fat-tree topology remains a cornerstone of modern data center network design.

2.2 Quality of Service

Quality of Service (QoS) refers to the ability of a network to provide better service to selected network traffic over various technologies, including Frame Relay [40], Asynchronous Transfer Mode (ATM) [41], Ethernet and 802.1 networks [42], SONET [43], and IP-routed networks. In the data center environment, QoS is crucial for ensuring that applications perform effectively by providing priority to critical traffic, dedicated bandwidth, controlled jitter and latency, and improved loss characteristics. This is achieved by classifying and managing the traffic flows based on predefined policies that dictate how network resources are allocated.

The importance of QoS in data centers can be summarized in several key points:

1. Priority to critical applications: Data centers host a range of applications, some of which are critical and require consistent network performance. QoS mechanisms ensure that these applications receive the necessary bandwidth and latency conditions even when the network is congested.

2. Resource utilization: Effective QoS policies help in maximizing the utilization of available network resources, thereby increasing the overall efficiency of the data center.

3. Reduced latency and improved reliability: By managing traffic flows and reducing bottlenecks, QoS helps in minimizing latency for time-sensitive applications and enhances the reliability of network services.

2.2.1 Protocls

Implementing QoS in data center networks involves several techniques and mechanisms:

- Traffic Classification: traffic entering the data center network is classified into different categories based on factors such as the type of service, user identity, or data application. This classification serves as the basis for applying QoS policies.

- Traffic Shaping and Policing: these mechanisms regulate the flow and volume of traffic being transmitted, ensuring that no single service or user consumes excessive bandwidth that could impact other services.

- Congestion Management: through techniques like queue management and congestion avoidance, QoS controls how data is buffered and transmitted during periods of high congestion, ensuring that high-priority traffic maintains high levels of performance.

- Service Level Agreements (SLAs): QoS parameters are often defined within SLAs that specify the performance characteristics guaranteed by the data center provider. Compliance with SLAs is critical for maintaining client trust and satisfaction.

Quality of Service is a foundational element in managing the complex and dynamic environment of data center networks. By intelligently prioritizing and managing network traffic, QoS ensures that data center resources are allocated efficiently and effectively, enhancing the performance of critical applications and improving the overall operational efficiency of data centers. As network demands continue to evolve, the role of QoS in maintaining stable and reliable network operations will only grow in significance.

3 Evaluation

In this project, we evaluate QoS techniques with ns-3 network simulator. The ns-3 simulator is a discrete-event network simulator, widely used in research and educational contexts for the design and performance analysis of network protocols and network architectures [44, 45].

3.1 Simulation Setup

The simulation studies are based on a system comprising 1,296 nodes interconnected via a three-tier, fat-tree network. The topology is featured by 25 GB/s links with 1000ns transmission latency based on the recent network technology. Each node links to one of 72 36-port level-1 (leaf) switches as shown in Figure 1.

3.2 Workloads

Two workloads, AMD [46] and Random are studied in this project:

- AMG: AMG (Algebraic MultiGrid) benchmark is a performance test used to evaluate the efficiency of high-performance computing systems, particularly their capability in solving large-scale algebraic equations. It is based on the Algebraic MultiGrid method, which is a numerical solution technique designed to efficiently solve linear systems of equations, especially those arising from discretized partial differential equations. This benchmark is important in scientific computing where solving large, sparse systems of equations is common and can be demanding on both computational resources and system network bandwidth.

- Random permutation: random traffic is a traffic pattern used in network system studies where each node communicates with a destination node chosen at random. This pattern is considered representative of realistic system usage, as it simulates an environment where multiple applications, each characterized by different traffic patterns, coexist and compete for network bandwidth. The randomness in destination selection helps to model the unpredictable nature of real-world network traffic, providing insights into network behavior under varied and complex load conditions.

Both workloads are assessed under various Quality of Service (QoS) priority settings, which allocate different percentages of the network link bandwidth. This evaluation helps to determine how effectively the network can manage and prioritize traffic from diverse applications, ensuring that bandwidth distribution aligns with the specific needs and priority levels of each workload.

3.3 Result Analysis

Figure 2 shows the application performance when AMG or Random is the only application on the system (standalone) or when both applications run at the same time (co-exist).

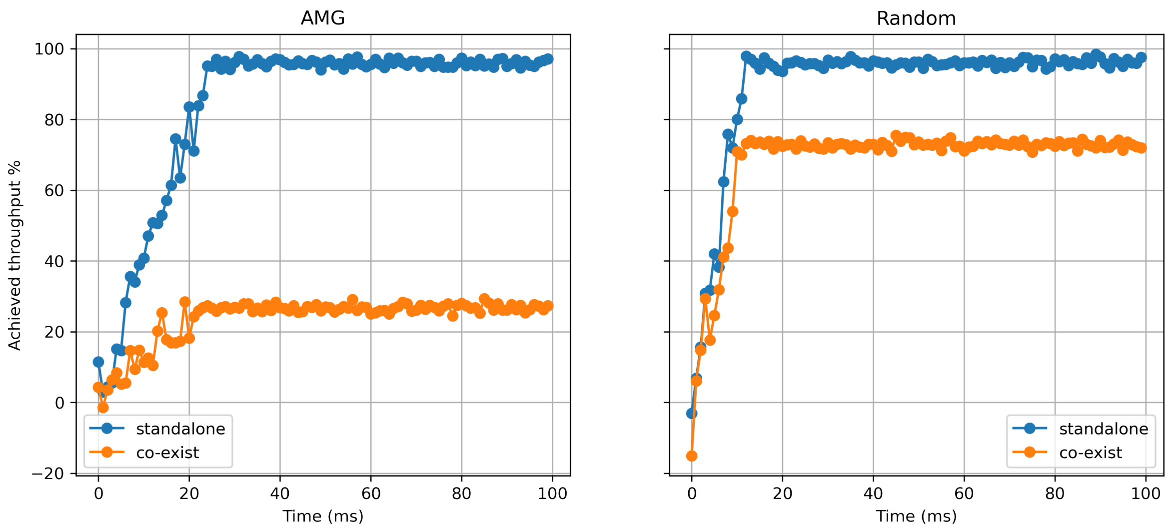

Figure 2: Achieved network bandwidth by AMG and Random without QoS. Both workloads are launched as the single job on the system (standalone case) and co-existing cases.

For AMG, in the standalone case, the performance sharply increases until it reaches 96% system network throughput. This suggests that without competition from Random, the system can rapidly achieve its full potential for serving AMG. However, when AMG is running with Random, the performance significantly struggles to rise and then stabilizes at a much lower level (27%) compared to the standalone scenario. This indicates that competition impacts the system's ability to perform optimally, possibly due to shared resources and increased load. For Random, in the standalone case, similar to AMG, performance rapidly increases and plateaus, showing stable and optimal performance without interference. The co-exist condition shows a different trend for Random than the case for AMG. Here, the performance closely matches the "standalone" condition throughout (70%), suggesting that Random application is more aggressive in bandwidth requirement.

Figure 3 shows the achieved network throughput of each application with QoS policy. The QoS policy is tuned to guarantee AMG being served with at least 80% system network bandwidth. For AMG in the standalone case, the system reaches its peak performance quickly and maintains a stable high level. When AMG is co-exist case, AMG's network throughput increases slower due to the network system sharing resources with Random. However, AMG is guaranteed with 80% system network bandwidth sharing. Since AMG's peak network requirement is greater than 80%, it maintains an 80% network throughput utilization, and sharing the rest system network bandwidth with Random. For Random in Standalone case, it maintains a consistent high performance throughout due to it is the only application on the system. QoS policy guarantees AMG network usage, which is missing from the system. When Random co-exists with AMG, Random starts lower and after initial variability, it becomes stable but significantly underperforms relative to the standalone scenario. This is caused by QoS allocating more network bandwidth to AMG as instructed.

Figure 3 illustrates the achieved network throughput for each application under a QoS policy. The QoS policy is configured to ensure that AMG receives no less than 80%

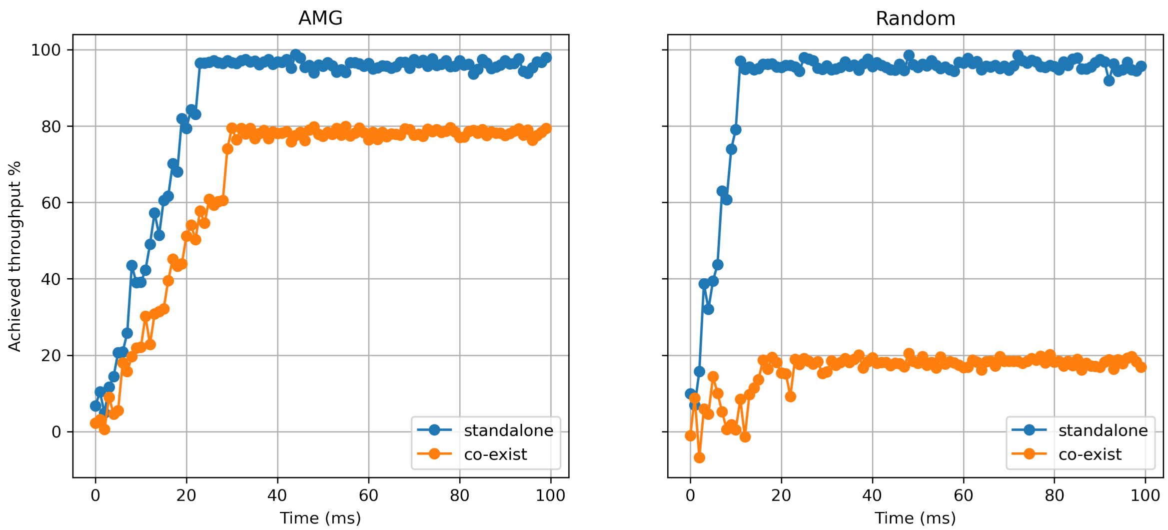

Figure 3: Achieved network bandwidth by AMG and Random with QoS. AMG is set to be satisfied for 80% of system bandwidth.

of the system's network bandwidth. In the standalone scenario, AMG quickly reaches peak performance and sustains a high level consistently. When AMG is in a co-existing scenario, its network throughput climbs more slowly due to shared network resources with Random. Nevertheless, AMG is assured 80% of the system network bandwidth. Since AMG's peak network requirement exceeds 80%, it consistently utilizes 80% of the network throughput, while the remaining bandwidth is shared with Random. In Random's standalone case, it consistently achieves high performance since it is the sole application on the system. Due to the absent of AMG application, the QoS policy is not triggered in the system. When Random operates concurrently with AMG, it starts lower and, despite some initial variability, stabilizes at a performance level significantly below that of the standalone scenario, due to QoS policies favoring AMG by allocating it more network bandwidth.

4 Conclusion

In summary, Quality of Service (QoS) is an essential element in the management of contemporary data center networks. It boosts network performance through efficient traffic and resource management, ensuring that all applications hosted on the network receive adequate support to operate effectively. As data traffic volume and variety expand, the importance of QoS in sustaining stable, reliable, and efficient network operations intensifies. Therefore, investing in sophisticated QoS strategies and technologies is vital for any data center committed to delivering high-quality, reliable services in a rapidly changing digital environment.

References

[1] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. "A scalable, commodity data center network architecture". In: *ACM SIGCOMM computer communication review* 38.4 (2008), pp. 63–74.

[2] Albert Greenberg et al. "VL2: A scalable and flexible data center network". In: *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 2009, pp. 51–62.

[3] Andrew Campbell, Geoff Coulson, and David Hutchison. "A quality of service architecture". In: *ACM SIGCOMM Computer Communication Review* 24.2 (1994), pp. 6–27.

[4] Narasimhaiah Gorla, Toni M Somers, and Betty Wong. "Organizational impact of system quality, information quality, and service quality". In: *The journal of strategic information systems* 19.3 (2010), pp. 207–228.

[5] Yao Kang, Xin Wang, and Zhiling Lan. "Q-adaptive: A multi-agent reinforcement learning based routing on dragonfly network". In: *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*. 2021, pp. 189–200.

[6] Jorge Cardoso et al. "Quality of service for workflows and web service processes". In: *Journal of web semantics* 1.3 (2004), pp. 281–308.

[7] Miyuru Dayarathna, Yonggang Wen, and Rui Fan. "Data center energy consumption modeling: A survey". In: *IEEE Communications surveys & tutorials* 18.1 (2015), pp. 732–794.

[8] Mohammad Alizadeh et al. "Data center tcp (dctcp)". In: *Proceedings of the ACM SIGCOMM 2010 Conference*. 2010, pp. 63–74.

[9] Theophilus Benson et al. "Understanding data center traffic characteristics". In: *ACM SIGCOMM Computer Communication Review* 40.1 (2010), pp. 92–99.

[10] Roger P Roess, Elena S Prassas, and William R McShane. *Traffic engineering*. Pearson/Prentice Hall, 2004.

[11] Sugam Agarwal, Murali Kodialam, and TV Lakshman. "Traffic engineering in software defined networks". In: *2013 Proceedings IEEE INFOCOM*. IEEE. 2013, pp. 2211–2219.

[12] Yao Kang et al. "Modeling and analysis of application interference on dragonfly+". In: *Proceedings of the 2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 2019, pp. 161–172.

[13] Michael Welzl. *Network congestion control: managing internet traffic*. John Wiley & Sons, 2005.

[14] Xiaolei Ma et al. "Large-scale transportation network congestion evolution prediction using deep learning theory". In: *PloS one* 10.3 (2015), e0119044.

[15] Peter Bodik et al. "Fingerprinting the datacenter: automated classification of performance crises". In: *Proceedings of the 5th European conference on Computer systems*. 2010, pp. 111–124.

[16] Xin Wang et al. “Union: An automatic workload manager for accelerating network simulation”. In: *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2020, pp. 821–830.

[17] Mohammad Alizadeh, Adel Javanmard, and Balaji Prabhakar. “Analysis of DCTCP: stability, convergence, and fairness”. In: *ACM SIGMETRICS Performance Evaluation Review* 39.1 (2011), pp. 73–84.

[18] M Farhan Habib et al. “Design of disaster-resilient optical datacenter networks”. In: *Journal of Lightwave Technology* 30.16 (2012), pp. 2563–2573.

[19] Carlos Colman-Meixner et al. “A survey on resiliency techniques in cloud computing infrastructures and applications”. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2244–2281.

[20] Leonidas Georgiadis et al. “Efficient network QoS provisioning based on per node traffic shaping”. In: *IEEE/ACM transactions on networking* 4.4 (1996), pp. 482–501.

[21] Chadi M Assi et al. “Dynamic bandwidth allocation for quality-of-service over Ethernet PONs”. In: *IEEE Journal on selected Areas in Communications* 21.9 (2003), pp. 1467–1477.

[22] Yao Kang, Xin Wang, and Zhiling Lan. “Study of workload interference with intelligent routing on dragonfly”. In: *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE. 2022, pp. 1–14.

[23] Aggeliki Sgora and Dimitrios D Vergados. “Handoff prioritization and decision schemes in wireless cellular networks: a survey”. In: *IEEE Communications Surveys & Tutorials* 11.4 (2009), pp. 57–77.

[24] Justin Moore et al. “Data center workload monitoring, analysis, and emulation”. In: *Eighth workshop on computer architecture evaluation using commercial workloads*. 2005, pp. 1–8.

[25] Greg Faanes et al. “Cray cascade: a scalable HPC system based on a Dragonfly network”. In: *SC’12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE. 2012, pp. 1–9.

[26] John Kim et al. “Technology-driven, highly-scalable dragonfly topology”. In: *ACM SIGARCH Computer Architecture News* 36.3 (2008), pp. 77–88.

[27] Lucian Popa et al. “A cost comparison of datacenter network architectures”. In: *Proceedings of the 6th International COnference*. 2010, pp. 1–12.

[28] Kashif Bilal et al. “A Comparative Study Of Data Center Network Architectures.” In: *ECMS*. 2012, pp. 526–532.

[29] Yao Kang, Xin Wang, and Zhiling Lan. “Workload interference prevention with intelligent routing and flexible job placement on dragonfly”. In: *Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 2023, pp. 23–33.

[30] Mung Chiang et al. “Layering as optimization decomposition: A mathematical theory of network architectures”. In: *Proceedings of the IEEE* 95.1 (2007), pp. 255–312.

[31] Sean W O'Malley and Larry L Peterson. "A dynamic network architecture". In: *ACM Transactions on Computer Systems (TOCS)* 10.2 (1992), pp. 110–143.

[32] Charles E Leiserson. "Fat-trees: Universal networks for hardware-efficient supercomputing". In: *IEEE transactions on Computers* 100.10 (1985), pp. 892–901.

[33] Sabine R Ohring et al. "On generalized fat trees". In: *Proceedings of 9th international parallel processing symposium*. IEEE. 1995, pp. 37–44.

[34] SA Ghozati and HC Wasserman. "The k-ary n-cube network: modeling, topological properties and routing strategies". In: *Computers & electrical engineering* 25.3 (1999), pp. 155–168.

[35] Crispín Gomez et al. "Deterministic versus adaptive routing in fat-trees". In: *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE. 2007, pp. 1–8.

[36] Xuan-Yi Lin, Yeh-Ching Chung, and Tai-Yi Huang. "A multiple LID routing scheme for fat-tree-based InfiniBand networks". In: *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* IEEE. 2004, p. 11.

[37] Yao Kang. "Workload Interference Analysis and Mitigation on Dragonfly Class Networks". PhD thesis. Illinois Institute of Technology, 2022.

[38] Yu Li and Deng Pan. "OpenFlow based load balancing for fat-tree networks with multipath support". In: *Proc. 12th ieee international conference on communications (icc'13), Budapest, Hungary*. 2013, pp. 1–5.

[39] Shavan Askar. "SDN-Based Load Balancing Scheme for Fat-Tree Data Center Networks". In: *Al-Nahrain Journal for Engineering Sciences* 20.5 (2017), pp. 1047–1056.

[40] Uyless D Black. *TCP/IP and Related Protocols: IPv6, Frame Relay, and ATM*. McGraw-Hill School Education Group, 1998.

[41] Martin De Prycker. *Asynchronous Transfer Mode solution for broadband ISDN*. Prentice Hall International (UK) Ltd., 1995.

[42] Silviu S Craciunas et al. "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks". In: *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 2016, pp. 183–192.

[43] Ori Gerstel, Philip Lin, and Galen Sasaki. "Combined WDM and SONET network design". In: *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*. Vol. 2. IEEE. 1999, pp. 734–743.

[44] Thomas R Henderson et al. "Network simulations with the ns-3 simulator". In: *SIGCOMM demonstration* 14.14 (2008), p. 527.

[45] George F Riley and Thomas R Henderson. "The ns-3 network simulator". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[46] Klaus Stüben. "Algebraic multigrid (AMG). An introduction with applications". In: (1999).