Design and Implementation of QoS Guarantee Algorithm in FPGA-based SDN (Postprint)

Authors: Wang Junkai, Zhang Xiangli, Zhang Xiangli

Date: 2023-02-15T00:00:00+00:00

Abstract

Traditional networks are increasingly incapable of addressing the growing complexity of network structures, thereby giving rise to a novel network architecture known as Software Defined Networking (SDN). The traffic flows within SDN data centers primarily comprise long flows and short flows. Long flows are characterized by extended duration, low delay sensitivity, and high bandwidth demands, whereas short flows exhibit brief duration, high delay sensitivity, and low bandwidth requirements. Although the traffic volume of short flows constitutes less than 20% of the total traffic, the number of flow entries accounts for over 80% of the total; conversely, long flows represent over 80% of the total traffic volume, yet the number of flow entries comprises less than 20% of the total. Research has demonstrated that in output port queues, long flows frequently precede short flows, causing extended waiting periods for short flows and readily inducing network congestion. Based on the distinct characteristics of these two traffic flow types, we propose a queuing mechanism and a routing optimization guarantee mechanism: short flows are designated to high-priority queues and scheduled preferentially by the SDN controller, while long flows are assigned to low-priority queues and compensated through a routing guarantee algorithm. The routing guarantee algorithm first eliminates links that fail to satisfy the bandwidth requirements of long flows, subsequently computing the path with minimal delay. To enhance the algorithmic efficiency of this design, we utilize FPGA and 10 Gigabit Ethernet to simulate traffic flows in SDN, and FPGA-based simulation validates the advantages of our design in optimizing network delay and bandwidth, as well as demonstrating the benefits of FPGA parallel computation.

Full Text

Design and Implementation of a QoS Guarantee Algorithm Based on FPGA and SDN

Journal of Guilin University of Electronic Technology, Vol. 43, No. 2, April 2023
Authors: Wang Junkai, Zhang Xiangli
Affiliation: School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
Email: 304146684@qq.com

Funding: This work was supported by the Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing Director Fund (2020GXNSFDA238001), Guangxi Cloud Computing and Big Data Collaborative Innovation Center Fund (YD1904), and Guilin University of Electronic Technology Graduate Education Innovation Program (2021YCXS021, GXKL06200104).

Abstract

Traditional networks are increasingly unable to cope with the growing complexity of network structures, giving rise to a new network architecture: Software Defined Networking (SDN). The main traffic flows in data centers consist of long flows and short flows. Long flows exhibit long duration and high bandwidth demands, whereas short flows have brief duration and high delay sensitivity. Although short flows account for less than 15% of total traffic volume, they represent approximately 80% of the total flow count. Conversely, long flows constitute over 85% of traffic volume but less than 20% of flow count. In output port queues, long flows often precede short flows, causing prolonged waiting times for delay-sensitive short flows. This paper proposes a novel queueing mechanism and routing optimization guarantee framework based on these traffic characteristics. Short flows are assigned to high-priority queues that receive preferential scheduling by the controller, while long flows are placed in low-priority queues and compensated through a routing guarantee algorithm. The routing guarantee algorithm first eliminates links that fail to meet long flow bandwidth requirements, then computes the shortest delay path. To enhance algorithm efficiency, this design leverages the parallel processing capabilities of FPGA. Experimental simulations demonstrate that the proposed approach effectively optimizes both network delay and bandwidth utilization.

Keywords: FPGA; SDN; QoS; Queue Scheduling; Routing Guarantee

---

1. Introduction

Network Quality of Service (QoS) guarantee mechanisms have attracted significant research attention for decades. With the emergence of Software Defined Networking (SDN), new solutions for QoS provisioning have become possible. Numerous scholars have contributed to applying SDN architecture to QoS guarantees [1-3].

Previous work includes Ghalwash et al. [4], who designed a parameter optimization framework for port utilization and delay in industrial networks. Guck et al. [5] addressed real-time traffic in industrial environments through rule preparation and configuration functions. Bagaa et al. [6] utilized SDN-based frameworks for medical applications. Chen et al. [7] proposed a multi-hop model for bandwidth provisioning, while Kucminski et al. [8] developed QoS-aware routing based on user demand prioritization. Li et al. [9] introduced a novel non-intrusive approach using Open vSwitches (OVSs). Yan et al. [10] combined routing and scheduling strategies, employing Equal Cost Multi-Path (ECMP) and Weighted Fair Queuing (WFQ). Sun et al. [11] presented HiQoS, an intelligent routing mechanism using machine learning for flow classification. Bahnasse et al. [12] proposed a local routing adjustment scheme that only modifies links around congested areas. Elbasheer et al. [13] developed a QoS-based routing algorithm for video streaming that selects different resolution parameters based on traffic type. Chen et al. [14] designed an energy-saving mechanism using SDN. Zhang et al. [15] improved throughput through heap sorting in OpenFlow.

However, most existing QoS guarantee methods rely on software platforms, which face processing bottlenecks [16]. They typically use soft processing approaches for queuing and multi-path forwarding, focusing only on specific scenarios without fully exploiting hardware acceleration.

2. Queue Scheduling and Routing Guarantee Algorithm Design

2.1 Problem Analysis

Data center traffic primarily consists of delay-sensitive short flows and bandwidth-intensive long flows. Common queue algorithms include First-Come-First-Served (FCFS) and Priority Queuing (PQ). The fatal flaw of PQ is that high-priority queues can monopolize service for extended periods, causing starvation in low-priority queues. In traditional FCFS scheduling, the network processes packets in arrival order without distinguishing traffic characteristics, leading to potential network congestion and performance degradation for delay-sensitive applications.

2.2 Proposed Mechanism

This design employs a priority queue scheduling algorithm that prioritizes short flows while guaranteeing bandwidth for long flows through routing compensa-

tion. The algorithm calculates shortest paths differently for each flow type: for short flows, it uses delay as the link weight; for long flows, it first removes links with insufficient bandwidth, then computes the shortest delay path.

Consider a simple network topology where traffic flows from source node S to destination node D. The original topology is processed differently for each flow type: short flows use Dijkstra' s algorithm directly to find the shortest delay path, while long flows first eliminate links with bandwidth below the flow' s requirements, then compute the shortest path on the modified topology.

The queue scheduling and routing guarantee algorithm provides several advantages over conventional ECMP. It adds bandwidth guarantees while reducing network overhead. Unlike ECMP, which splits high-bandwidth long flows across multiple paths—introducing flow segmentation overhead and increased inter-node communication—our algorithm selects a single optimal path that meets bandwidth requirements, thereby reducing device-to-device coordination overhead.

2.3 Algorithm Implementation Process

The implementation consists of several key modules: flow classification, priority scheduling, and routing guarantee. The algorithm module design framework processes incoming data as follows:

1. **Packet Reception and Classification**: Upon receiving data, the classification module examines the Type of Service (ToS) field. Network configuration packets bypass the algorithm module and proceed directly to encapsulation and transmission. Data packets are classified as either long flow or short flow based on ToS field information.

2. **Topology Management**: Network topology information is stored in memory (RAM1) as an adjacency matrix. Each entry represents a network link with fields for source switch, destination switch, port numbers, bandwidth weight, and delay weight. The algorithm updates this matrix dynamically by removing insufficient-bandwidth links for long flow calculations.

3. **Path Computation**: For short flows, Dijkstra' s algorithm computes the shortest delay path using delay as the link metric. For long flows, the algorithm first filters the topology to include only links satisfying the bandwidth requirement, then applies Dijkstra' s algorithm on the reduced graph.

4. **Result Encapsulation**: After path computation, results are encapsulated in a custom packet format containing source/destination nodes, path information, and QoS parameters, then transmitted to the controller for flow table installation.

The FPGA implementation leverages parallel computing capabilities and high-

speed data interfaces, making it suitable for large-scale data processing. The hardware description language implementation ensures system stability with high-precision clocking, though its complexity exceeds software implementations.

3. Experimental Validation

3.1 Testbed Setup

Experiments were conducted using a K7325T FPGA development board with a 10 Gigabit Ethernet interface. The board communicates with a server at IP address 192.168.1.110, with subnet mask 255.255.255.0. MAC addresses for ports 0-3 are configured as 58:69:6C:69:6E:78, 58:69:6C:69:6E:79, 58:69:6C:69:6E:7A, and 58:69:6C:69:6E:7B respectively.

3.2 Routing Guarantee Algorithm Verification

The Abilene network topology [18] was used for validation, with random link delays in range [0,5] and bandwidth weights in [1,10]. Simulation software generated topology and traffic request information, which was sent to the FPGA in custom packet formats. The hardware implementation, described in Verilog within the Vivado IDE, computed shortest paths and wrote results to FIFO buffers for UDP transmission.

For example, a path computation result might be encoded as `fifo_{udp}_{data}_{din}[127:119] = 0x01010201`, representing specific switch nodes and path characteristics. The hardware correctly identified shortest delay paths for short flows and bandwidth-satisfying paths for long flows.

3.3 Performance Analysis

Queueing Delay: Experiments compared queueing delay between FCFS and priority scheduling. In FCFS, long flows can precede short flows, causing excessive delay for delay-sensitive traffic. With priority scheduling, short flows receive immediate service after network topology processing, while long flows wait for queue scheduling completion. The queueing delay for short flows remains constant regardless of network load, whereas long flow delay increases with queue length. This demonstrates that the design successfully guarantees short flow delay requirements while sacrificing some long flow queueing delay.

Acceleration Performance: To validate routing algorithm correctness and efficiency, hardware execution times were compared against software implementations running on the same Abilene topology and other topologies (Poska, Atlanta). The acceleration ratio S is defined as:

$$S = \frac{\text{Software Execution Time}}{\text{Hardware Execution Time}}$$

Table 1 shows hardware and software execution times across different topologies, demonstrating significant speedup. The hardware implementation processes the routing guarantee algorithm much faster than software, with acceleration ratios varying by topology complexity.

Processing Delay: After queue scheduling, the system computes paths and installs flow tables. Initial flow table installation requires several tens of microseconds for the first route computation. Subsequent flows between the same source-destination pairs with identical bandwidth requirements only need table lookup operations, incurring minimal processing delay.

4. Conclusion

This paper presents a queue scheduling and routing guarantee algorithm that addresses the conflicting requirements of short and long flows in SDN environments. By assigning high priority to delay-sensitive short flows and compensating bandwidth-intensive long flows with intelligent path selection, the design simultaneously guarantees QoS for both flow types. The FPGA-based hardware implementation exploits parallel processing to achieve significant performance acceleration over software approaches. Experimental results demonstrate that the algorithm effectively reduces short flow delay while providing adequate bandwidth for long flows.

Future work will focus on optimizing the algorithm for more complex network topologies and further enhancing its efficiency through refined hardware architecture design.

References

[1] TOMOVIC S, PRASAD N, RADUSINOVIC I. SDN control framework for QoS provisioning[C]//IEEE Conference on Telecommunications Forum. Piscataway, NJ: IEEE Press, 2014: 111-114.

[2] BARI M F, CHOWDHURY S R, AHMED R. PolicyCop: an autonomic software defined networking framework for policy enforcement[C]//IEEE Conference on Future Networks and Services. Piscataway, NJ: IEEE Press, 2013: 1-7.

[3] VENKATESH K, SRINIVAS B, KRISHNAN M K, et al. QoS improvisation for delay sensitive communication in SDN-based networks[C]//2018 International Conference on Internet Computing. Piscataway, NJ: IEEE Press, 2018: 98-105.

[4] GHALWASH A, GUCK J, HUANG H. DET-SERV: network actions for medical applications[J]. Future Generation Computer Systems, 2019, 93: 256-265.

[5] GUCK J, VAN BEMTEN A, KELLERER W. QoS-aware routing discovery using multiple optimization parameters[J]. IEEE Transactions on Communications, 2020, 19(7): 4700-4714.

[6] BAGAA M, DUTRA C, TALEB T. On driven network provisioning in industrial environments[J]. IEEE Transactions on Service Management, 2017, 14(4): 1003-1017.

[7] CHEN H, HUANG H. SDN-based framework for real-time traffic in industrial networks[C]//2018 International Conference on Collaboration Computing. Piscataway, NJ: IEEE Press, 2018: 98-105.

[8] KUCMINSKI A, AL-JAWAD A, SHAH S. QoS-aware source routing in software defined networks[C]//International Symposium on Broadband Multimedia Systems. Piscataway, NJ: IEEE Press, 2017: 1-6.

[9] LI L, CAO N, WANG K. QoS-aware top-K paths in SDN[J]. KSII Transactions on Internet and Information Systems, 2018, 12(6): 2534-2553.

[10] YAN Y, ZHANG L, SHUAI J. HiQoS: an SDN-based multipath solution[J]. China Communications, 2015, 12(5): 123-133.

[11] SUN F, WANG Z, ZHANG Y. QoS-guaranteed routing mechanism in software-defined networks[J]. Computer Networks, 2021, 185: 107709.

[12] BAHNASSE A, LOUHAB K, OULAHYANE A, et al. Novel smart traffic management architecture based on DS-TE and OpenFlow[J]. Future Generation Computer Systems, 2018, 87: 115-126.

[13] ELBASHEER O, ALDEGHEISHEM A, LLORET J. QoS-Based routing algorithm for video streaming in software defined networks[J]. Journal of Network and Computer Applications, 2021, 194: 103215.

[14] CHEN H, WANG J, SHEN C. Achieving energy saving guarantee in cloud applications using software defined networking[C]//2016 International Conference on Computational Intelligence and Security. Piscataway, NJ: IEEE Press, 2016: 1-7.

[15] ZHANG Q, WANG H, DENG J. Implementation of heap sorting processing structure in QoS guarantee mechanism[J]. Journal of Guilin University of Electronic Technology, 2018, 38(1): 12-15.

[16] DENG H, WANG J, SHEN C. An inter-domain multi-control transmission selection network based on FPGA[C]//2018 International Conference on Computational Intelligence and Security. Piscataway, NJ: IEEE Press, 2018: 66-70.

[17] WANG J, ZHANG X. Design and implementation of 10 Gigabit Ethernet hardware protocol stack[J]. Computer Applications Research, 2022, 38(4): 11-14.

[18] ORLOWSKI S, WESSALY R, TOMASZEWSKI A. SNDlib library[EB/OL]. (2017-12-17)[2022-11-15]. http://sndlib.zib.de/home.

[19] BENSON T, AKELLA A, MALTZ D A. Network traffic characteristics of data centers in the wild[C]//Proceedings of SIGCOMM Conference on Internet Measurement. New York, NY: ACM Press, 2010: 267-280.

[20] AFAQ M, REHRMAN U, SONG H. SDN-based framework for elephant flow classification and visualization[J]. Procedia Computer Science, 2015, 65: 672-681.

Editor: Liang Wanghuan

Note: Figure translations are in progress. See original paper for figures.