Maintaining QoS-aware and Resilient Path Programmability for Metaverse in SD-WANs

Songshi Dou, Li Qi, and Zehua Guo
Beijing Institute of Technology
Beijing 100081, China
songshidou@hotmail.com,3120210950@bit.edu.cn,guolizihao@hotmail.com

ABSTRACT

To improve the performance of the metaverse, it is of great importance to maintain path programmability, which enables dynamic flow control to achieve low latency for metaverse applications in Software-Defined Wide Area Networks (SD-WANs), especially when controllers fail. In this paper, we propose a QoS-aware path programmability recovery scheme during controller failures, which utilizes the P4 Runtime to achieve fine-grained flow-controller remapping. Simulation results demonstrate that the proposed solution can recover all metaverse flows, improve the total recovery benefit by up to 375%, and substantially reduce the communication overhead of metaverse flows compared wtih baselines.

ACM Reference Format:
Songshi Dou, Li Qi, and Zehua Guo. 2023. Maintaining QoS-aware and Resilient Path Programmability for Metaverse in SD-WANs. In *ACM Turing Award Celebration Conference 2023 (ACM TURC '23), July 28–30, 2023, Wuhan, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3603165.3607407

1 INTRODUCTION

To deliver a truly immersive and seamless experience, reducing latency is essential for the metaverse, especially in situations that require intensive user interaction and real-time data processing tasks [4]. Metaverse applications require significant computation resources which are usually provided by data centers through Wide Area Network (WAN). Software-Defined Networking (SDN) offers a solution to reduce data transmission latency in WANs by separating the control plane and data plane to provide efficient network management, which is also known as Software-Defined Wide Area Networks (SD-WANs).

ACM TURC '23, July 28–30, 2023, Wuhan, China
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0233-4/23/07.
https://doi.org/10.1145/3603165.3607407

The SDN controller can realize flexible network control by routing/rerouting flows to accommodate traffic variation and further improve network performance. However, it may fail due to unforeseen issues (*e.g.*, power failure or network attacks), and the flows under its control will become offline and lose path programmability. Existing efforts on recovering path programmability during controller failures aim to establish switch-controller mappings [2, 3]. Unfortunately, these solutions suffer from poor recovery performance since the available control resources cannot be fully utilized in coarse-grained per-switch mappings.

In this paper, we present a QoS-aware and resilient path programmability recovery solution. Using the innovative control plane design of P4 programmable switches (*i.e.*, P4 Runtime), we propose to partition the control plane and enable multiple controllers to control a single switch. This leads to the possibility of fine-grained flow-controller remapping. We take QoS requirements into account to ensure optimal recovery performance for metaverse flows. We formulate the *QoS-aware Flow-Controller Remapping (QFCR)* problem, and our proposed solution solves the problem to remap offline flows to active controllers.

2 PROBLEM ANALYSIS AND SOLUTION

The development of the programmable data plane has enabled us to achieve precise and flow-level control. P4 is a language used for programming the data plane, and P4 Runtime is a new method for the control plane to manage the data plane [1]. Specifically, P4 Runtime is a control plane specification for P4 programmable switches, which allows multiple controllers to connect to the P4 Runtime server running on the switch and control it simultaneously. This allows for multiple controllers to manage a single switch, which aligns with our design goals.

With the help of the above-mentioned technique, we can achieve fine-grained flow-level control and formulate our QFCR problem as follow. In the QFCR problem, offline flows can be categorized into three types with different priorities. All metaverse flows are with the highest priority. The QFCR problem aims to maximize the benefits of recovering high and medium priority flows, which are calculated as the product of the total programmability and benefit ratios. High priority flows have a much higher benefit ratio than medium priority flows. In addition, the QFCR problem also tries to minimize the communication overhead of recovered metaverse flows, which is equal to the sum of propagation delays between switches and controllers.

To address the QFCR problem, we propose a single objective function that balances the two goals by incorporating a constant weight factor. The results of the QFCR problem is then formulated and solved using optimization solvers. However, as the network size grows, the solution space expands exponentially, leading to an unfeasible computation time. For this reason, the current approach is limited to small network sizes. In future research, we aim to develop a more efficient heuristic algorithm that balances performance and computation complexity in solving the QFCR problem.

3 PRELIMINARY SIMULATION

3.1 Simulation Setup

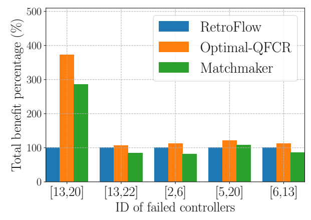
We evaluate the effectiveness of our solution by using a real-world topology called AT&T from the Topology Zoo. We use the Haversine formula and describe the propagation delay between nodes as the distance divided by the propagation speed (*i.e.*, 2×10^8 m/s). According to the results in [5], 50% of flows in real-world situations are with high priority. Thus, we set 50% as metaverse flows with high priority, 25% with medium priority, and 25% with low priority in this paper.

3.2 Comparison Algorithms

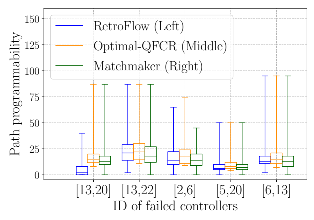
(1) RetroFlow [3]: this scheme alleviates the recovery pressure by using the hybrid SDN/legacy routing mode.
(2) Optimal-QFCR: it is the optimal results of our proposed QFCR problem.
(3) Matchmaker [2]: this scheme adjusts the control cost of switches to match the available control resources of active controllers by dynamically rerouting flows.

3.3 Simulation Results

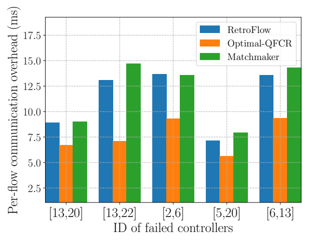
Figure 1 displays the results of two of the six controllers failing. Due to limited space, we only show 5 of 15 failure scenarios in this paper. Figure 1(a) illustrates the total recovery benefit. For the controller failure case {13, 20}, Optimal-QFCR outperforms RetroFlow by 375%. This is because the available control resources of active controllers cannot be perfectly matched to any of the active controllers' resources for RetroFlow and Matchmaker. Figure 1(b) displays the recovered path programmability of metaverse flows and medium priority flows. For all cases, the medians of RetroFlow and Matchmaker are lower than those of Optimal-QFCR. Figure 1(c) shows the per-flow remapping communication overhead of metaverse flows. Optimal-QFCR has the least communication overhead because it takes the propagation delay of controlling metaverse flows into account.



(a) Percentage of the total recovery benefit to RetroFlow.



(b) Path programmability of metaverse and medium priority flows.



(c) Per-flow mapping communication overhead of metaverse flows.

Figure 1: Simulation results under AT&T topology.

4 CONCLUSION AND FUTURE WORK

In this work, we present a QoS-aware approach to maintain robust path programmability and minimize communication overhead for metaverse flows. Thanks to the P4 Runtime supported by P4 programmable switches, our solution realizes accurate per-flow control by intelligently remapping offline flows from their traversing switches to active controllers.

In future efforts, we plan to analyze the complexity of the QFCR problem and design an efficient heuristic algorithm to solve it efficiently for large-scale network topologies. Additionally, we will conduct further simulations on various topologies to assess the scalability of our proposed solution.

ACKNOWLEDGMENTS

The corresponding author is Zehua Guo. This work was supported by the National Natural Science Foundation of China under Grant 62002019, Zhejiang Lab Open Research Project under Grant K2022QA0AB02, and SongShan Laboratory Fund under Grant YYJC022022009.

REFERENCES

[1] 2023. P4Runtime Specification version 1.3.0. https://p4.org/p4-spec/p4runtime/main/P4Runtime-Spec.html.
[2] Songshi Dou et al. 2021. Matchmaker: Maintaining network programmability for Software-Defined WANs under multiple controller failures. *Computer Networks* 192 (2021), 108045.
[3] Zehua Guo et al. 2019. RetroFlow: Maintaining control resiliency and flow programmability for software-defined WANs. In *IWQoS'19*. 1–10.
[4] Penny Sweetser, Zane Rogalewicz, and Qingyang Li. 2019. Understanding enjoyment in VR games with GameFlow. In *VRST'19*. 1–2.
[5] Zhaohua Wang et al. 2021. Examination of WAN traffic characteristics in a large-scale data center network. In *IMC'21*. 1–14.