A Data Model-Driven Management Scheme for Quality-of-Service Enforcement in Software-Defined Networking

Hojae Kim*, Kira Vanessa Eichhorn†, Jiwon Suh†, Jaehoon (Paul) Jeong†, and Tae (Tom) Oh‡
*Department of Electrical and Electronic Engineering, Sungkyunkwan University, Suwon, Republic of Korea
†Department of Computer Science, Sungkyunkwan University, Suwon, Republic of Korea
‡School of Information, Rochester Institute of Technology, Rochester, NY, USA
Email: {jaykim817, sjw6136, pauljeong}@skku.edu, kira-vanessa.eichhorn@informatik.hs-fulda.de, thoics@rit.edu

***Abstract*—This paper presents a YANG-driven QoS enforcement framework integrated with an OpenFlow SDN controller. High-level bandwidth policies defined in a custom YANG model are translated into HTB-based shaping rules applied to a Mininet bottleneck link. Experimental results show that, unlike best-effort forwarding where background traffic collapses to 11.8 kbps, the proposed system guarantees stable 500 kbps and 250 kbps bandwidths for emergency and background flows, respectively. These results demonstrate the effectiveness of combining SDN programmability with YANG policy modeling for automated QoS control.**

***Index Terms*—Software-Defined Networking (SDN), YANG Data Modeling, NETCONF, QoS Enforcement, Bandwidth Reservation, Mininet, Traffic Shaping.**

I. Introduction

The growing diversity of network applications introduces increasingly stringent quality-of-service (QoS) requirements that cannot be satisfied by traditional best-effort forwarding. Software-Defined Networking (SDN) provides centralized control and global visibility of traffic flows [1], yet QoS enforcement in many SDN systems remains tightly coupled to controller-specific logic, limiting portability and scalability.

YANG, a data modeling language for NETCONF, offers a structured and standardized way to express configuration intent, making it suitable for policy-driven QoS management. Integrating YANG with SDN enables the decoupling of high-level QoS intent from low-level forwarding behavior. Prior work in fog and edge computing architectures, such as SDFog [2], highlights the importance of flexible and programmable QoS orchestration across distributed environments. Similarly, SDN-based QoS controllers and allocation frameworks have demonstrated the feasibility of mapping application requirements to network resource reservations [3], [4].

This paper presents a small-scale prototype that demonstrates YANG-driven QoS enforcement in an SDN-based Mininet environment.

II. Related Work

SDN has been widely studied as an enabler of programmable network control and management [1]. In particular, SDN and Network Function Virtualization (NFV) have been explored as foundations for QoS-aware traffic engineering and service orchestration [5]. These approaches leverage centralized control to improve flexibility and resource utilization in dynamic network environments.

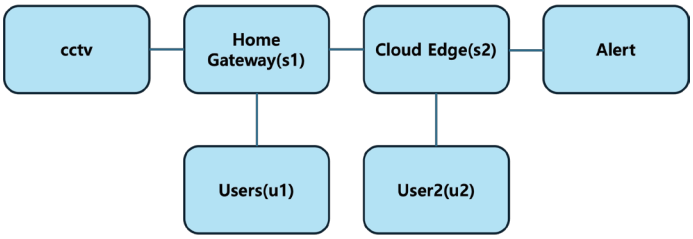

Fig. 1. Framework of YANG-driven QoS enforcement over SDN.

Several SDN-based QoS controllers and allocation schemes have been proposed to provide guaranteed bandwidth and application-aware resource management [3], [4]. While these systems demonstrate effective QoS enforcement, they often rely on controller- specific implementations. In contrast, model-driven approaches such as SDFog [2] emphasize the use of abstract policy models to decouple QoS intent from underlying mechanisms. Our work follows this direction by employing YANG as a standardized interface for expressing QoS policies and demonstrating lightweight enforcement in a Mininet-based SDN testbed.

III. Design

The proposed QoS control framework consists of four layers: the data plane, SDN control plane, YANG-based policy layer, and QoS translation layer, as shown in Fig. 1.

A. Data Plane

The data plane is implemented using two Open vSwitch instances (s1 and s2) connected through a constrained bottleneck link. Four hosts generate traffic: a CCTV emergency source, an alert receiver, and a background sender–receiver pair. Access links operate at 100 Mbps, while the bottleneck is configured at 1 Mbps with 20 ms delay, enabling controlled evaluation of QoS behavior under congestion [3].

B. SDN Control Plane

A Ryu controller provides OpenFlow 1.3 connectivity and performs basic L2 learning for forwarding without embedding QoS logic. This separation allows the controller to focus on forwarding while QoS enforcement is handled externally [1].

C. YANG-Based Policy Layer

A custom YANG module, *hsh-qos*, defines two configuration parameters: *emergency-bw*, specifying reserved bandwidth for emergency traffic, and *background-bw*, specifying bandwidth for background flows. These parameters are supplied via an XML configuration file (*qos-config.xml*), decoupling QoS intent from controller implementation. This model-driven approach aligns with the use of YANG and NETCONF in fog architectures such as SDFog [2].

D. QoS Translation Layer

The QoS Translator enforces YANG-defined policies by installing an HTB root qdisc on the bottleneck interface (*s1-eth3*) and creating two child classes for emergency and background traffic with rates derived from the YANG configuration. Packet classification is performed using *u32* filters, mapping emergency traffic (10.0.0.1 to 10.0.0.2) to the high-priority class and all remaining flows to the background class. This design enables lightweight, intent-based QoS automation in a Mininet-based SDN environment.

IV. Emulation Results

Two scenarios are evaluated to examine the effectiveness of the proposed QoS framework: (1) best-effort forwarding without QoS enforcement and (2) YANG-driven QoS enforcement using HTB shaping. UDP traffic is generated using a 500 kbps emergency stream and a 250 kbps background stream.

A. Best-Effort Forwarding

Under best-effort forwarding, no traffic shaping is applied at the bottleneck link. While the emergency stream sustains its sending rate of approximately 500 kbps, the background flow experiences severe starvation and achieves only about 11.8 kbps due to FIFO queuing under congestion.

B. YANG-Driven QoS Enforcement

With YANG-driven QoS enforcement enabled, emergency and background flows are isolated into separate HTB classes on the bottleneck interface. Both flows achieve their configured bandwidth levels with stable throughput over time.

C. Summary

Overall, the emulation results demonstrate that YANG-driven shaping eliminates starvation and provides predictable, policy-compliant bandwidth allocation compared to best-effort forwarding.

V. Performance Evaluation

We evaluate the hybrid QoS approach in terms of throughput stability, fairness, and policy responsiveness, focusing on the benefits of decoupling policy expression from enforcement.

A. Throughput Stability

Under YANG-driven shaping, both emergency and background flows achieve throughputs close to their configured rates with low variability, demonstrating effective class isolation by HTB in a constrained environment.

B. Fairness and Resource Allocation

Compared to best-effort forwarding, where the background flow achieves only 11.8 kbps, the shaped configuration improves background throughput by more than a factor of 20 while preserving higher priority for emergency traffic, ensuring policy-compliant resource allocation.

C. Policy Responsiveness

QoS parameters are defined in YANG and provided via XML, allowing bandwidth allocation to be modified without changes to SDN controller logic. The translation layer applies updates dynamically, reducing operational overhead and supporting intent-based network management. The same YANG model can be extended to additional QoS mechanisms or NFV-based service chains [5].

VI. Conclusion

This paper presented a YANG-driven QoS enforcement framework operating over an SDN-controlled Mininet environment. Declarative bandwidth policies are translated into HTB shaping rules, achieving stable and fair bandwidth allocation and eliminating starvation observed under best-effort forwarding. The results demonstrate that combining SDN programmability with structured YANG policy models enables lightweight QoS automation. Future work includes integrating a full NETCONF server and extending the approach to additional QoS parameters and larger or multi-hop topologies.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2024-00398199) and the National Research Foundation of Korea (NRF) grant (No. 2023R1A2C2002990) funded by the Ministry of Science and ICT (MSIT), South Korea. Jaehoon (Paul) Jeong is the corresponding author.

References

[1] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[2] H. Gupta, S. B. Nath, S. Chakraborty, and S. K. Ghosh, "SDFog: A software defined computing architecture for QoS-aware service orchestration over edge devices," *arXiv preprint arXiv:1609.01190*, 2016.

[3] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *APSIPA ASC*, 2012, pp. 1–8.

[4] A. V. Akella and K. Xiong, "QoS-guaranteed network resource allocation via software defined networking (SDN)," in *IEEE 12th Int. Conf. on Dependable, Autonomic and Secure Computing (DASC)*, 2014, pp. 7–13.

[5] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.