# FlowQoS: Not Every Flow is Born the Same


Dhruv Sharma
UC San Diego, CA
dhsharma@cs.ucsd.edu

Robert Jenkins
UC San Diego, CA
rfjenkins@ucsd.edu

Frederik Nygaard
UC San Diego, CA
freddyny@ucsd.edu

Feichao Qian
UC San Diego, CA
feqian@ucsd.edu



***Abstract*—ith majority of the world's data and computation handled by cloud-based systems, cloud management stacks such as Apache's CloudStack, VMware's vSphere and OpenStack have become an increasingly important component in cloud software. However, like every other complex distributed system, these cloud stacks are susceptible to faults, whose root cause is often hard to diagnose. We present Hansel, a system that leverages non-intrusive network monitoring to expedite root cause analysis of such faults manifesting in OpenStack operations. Hansel is fast and accurate, and precise even under conditions of stress. ith majority of the world's data and computation handled by cloud-based systems, cloud management stacks such as Apache's CloudStack, VMware's vSphere and OpenStack have become an increasingly important component in cloud software. However, like every other complex distributed system, these cloud stacks are susceptible to faults, whose root cause is often hard to diagnose. We present Hansel, a system that leverages non-intrusive network monitoring to expedite root cause analysis of such faults manifesting in OpenStack operations. Hansel is fast and accurate, and precise even under conditions of stress. W**


## I. Introduction

In modern times, user devices connected to broadband access networks, run an assortment of applications that exchange network traffic with remote servers and other devices over the internet. Since the upstream and downstream throughput may be limited at most time, those traffics will compete for relatively scarce bandwidth resources.

However, traffic from one application might not share the same characteristics as the traffic emerging from another application. To a large extent, it is the end-users requirements that play a significant role in deciding the nature of such applications and hence the kind of network traffic they send and receive. For instance, a users expectation from a VoIP call, video streaming and gaming applications is that their experience remains seamless and high quality during their use of the application, requiring the traffic to be sent out at near constant rates along low delay and low loss paths over the network. Whereas, in certain use cases such as data backup to the cloud and system updates, it is expected that the operation is completed eventually even when the user is not actively using the application. Traffic from such applications, in contrast to VoIP and video streaming, does not come with any hard deadlines or network requirements.

While the nature of traffic varies so widely, the network devices today are, to a large extent, agnostic to such subtleties in the nature of network traffic and tend to handle it in the same way. Even though later kind of applications discussed above do not face any issues that directly affect the user, the former kind might be impacted severely due to the effect on network dynamics leaving the user with a sour experience.

Research over past many years has resulted in the use of various metrics such as packet delays, jitter, available bandwidth, frame rate etc. to quantify users experience in some way. One possible way to deal limited throughput is to configure the network routers to prioritize some specific applications' traffic flows (e.g. video, VoIP etc) over others (e.g. data backup, file upload etc). It will effectively improve Quality of Service (QoS). However, for some reasons, previous work on QoS mechanisms have not been deployed in broadband access networks [13]. The emergence of software-defined networking (SDN) gives more possiblities to solve this problem. One approach to deploying QoS in broadband access networks is to utilize the advantages of SDN that separates the network's control logic and forwarding planes. We can migrate the functions that perform QoS both application identification and router-level configuration to separate control logic, and design a front-end client (e.g., webpage) at high levels of abstraction to let users to assign bandwidth to each identified application according to their own preferrence. Once users set up their preference, the front-end will install the QoS configuration into the home routers [13]. This approach makes it easy for user to configure priorities and facilitates more sophisticated per-flow application based QoS.

In this paper, we focus on FlowQoS [13] solution that utilizes SDN and traffic policing in virtual switches to achieve network isolation in traffic flow types. We further provide an alternate implementation using Linux traffic control features to achieve the same goal with possible optimization in the use of available bandwidth.

Our contributions. First, we deploy the FlowQoS implementation of pair of virtual switches for improving QoS, within a virtual machine (emulating our end-host) with internet connection instead of the discussed hardware implementation. Second, we validate the results presented by FlowQoS [13] using similar tools and scenarios that they use. Finally, we intend to develop an SDN controller and agent application that utilizes Linuxs advanced routing and traffic control to implement the idea of FlowQoS while overcoming the limitation of under-utilization of available bandwidth.

Paper outline. The rest of the paper is organized in the following way. We discussed related work in QoS, as well as SDN-based solutions for home and broadband access networks in section 2. The motivation of FlowQos is presented in Section 3. Section 4 records some problems we encountered.

Section 5 describes the design of FlowQoS and Section 6 describes its implementation. Section 7 evaluates FlowQoS for video streaming and VoIP applications in the context of competing flows. We discusses future work and open research avenues in Section 8 and concludes in Section 9.

## II. Related Works

There is significant previous work for QoS in IP networks [1], [7], [10], traffic shapers, and traffic flows classifiers [12]. However, most previous approaches are different with FlowQoS either focusing on different issues or working in different scenarios. FlowQoS focuses in particular on making per-flow, application-based QoS, which is designed to deploy and configure in home networks.

Kim et al. provides a solution [6], which sets rate limiters at the edge switches and priority queues for flow at each path hop. It uses a QoS control framework to manage automatically OpenFlow networks with multiple switches. On the contrary, FlowQoS provides similar automated trafiic shaping at a single gateway. Ishimori et al. developed QoSFlow [5]. It is a system that provides QoS in OpenFlow networks. QoSFlow shares similar shaping mechanism with FlowQoS. However, it does not focus on providing usable QoS for broadband access networks and is still under development. Ko et al. proposed a two-tier flow-based QoS management framework [9], which needs multicore processors and is not designed for home networks like FlowQoS. Ferguson et al. developed PANE, a system that allows a user to reserve guaranteed minimum bandwidth between two hosts [3]. PANE addresses on more issues than FlowQos, which makes it do not focuse on QoS in broadband access networks or application identification. Williams et al. [14] developed an automated IP traffic classification algorithm based on statistical flow properties. This approach limited the throughout of commodity home routers to 28 Mbps. In contrast, FlowQoS faces no such limitations.

The emergence of SDN provides more possible solutions for QoS. A lot previous work is based on SDN. Risso et al. [11] developed an OpenFlow-based mechanism for customizing data-plane processing in home routers, but the architecture is focused on more general data-plane modifications, not QoS. Georgopoulos et al. [4] proposed an OpenFlow-assisted framework that improves users quality of experience (QoE) in home networks for multimedia flows. However, it performs per-device QoS. Mortier et al. [8] developed Homework and Carbone et al. [2] developed a port of Dummynet for OpenWrt. Both of them do not perform any application classification.

## III. Motivation

Consider the case of physical network connection for a host that provides maximum total bandwidth B Kbps. Now, consider two different traffic types being forwarded through this link say, video (UDP) traffic and file transfer (TCP) traffic. Let the video traffic be running at some bit-rate that produces an equivalent traffic at rate x Kbps. Now the TCP traffic starts and as there is (B-x) Kbps of bandwidth still available, it increases it window size by maximum segment size MSS every RTT msecs. It linearly increases its rate until it tries to send traffic more than (B-x) Kbps, thereby increasing the total traffic exceeds the threshold of B Kbps that the link is capable of sending. As it is not possible to send traffic at that rate, both video and TCP traffic face packet losses. TCP based on its congestion control algorithm, halves its window size, and hence the rate. But, it again tries to linearly increase the rate and crosses the (B-x) Kbps available bandwidth threshold. This happens continuously until the video streaming service adapts itself to a lower bitrate and therefore a lower datarate y (¡ x) Kbps. But, this only gives TCP more room to expand and repeating the same process over again, thereby reducing the bitrate even further, until an equilibrium is reached.

However, if we isolate the TCP and video traffic in separate queues/channels that put a hard rate limit on the either type of traffic, we can ensure that TCP can never try to consume the bandwidth required/used by the video traffic. FlowQoS helps us achieve the same effect, thereby providing the required Quality of Service to the sensitive traffic types.

## IV. Problem

## V. Design

## VI. Implementation

## VII. Evaluation

## VIII. Discussion

## IX. Conclusion

## References


[1] C. Aurrecoechea, A. T. Campbell, and L. Hauw. A survey of qos architectures. *Multimedia systems*, 6(3):138–151, 1998.

[2] M. Carbone and L. Rizzo. Dummynet revisited. *ACM SIGCOMM Computer Communication Review*, 40(2):12–20, 2010.

[3] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Participatory networking: An api for application control of sdns. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 327–338. ACM, 2013.

[4] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards network-wide qoe fairness using openflow-assisted adaptive video streaming. In *Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking*, pages 15–20. ACM, 2013.

[5] A. Ishimori, F. Farias, I. Furtado, E. Cerqueira, and A. Abelém. Automatic qos management on openflow software-defined networks. *IEEE Journal*, 2012.

[6] W. Kim, P. Sharma, J. Lee, S. Banerjee, J. Tourrilhes, S.-J. Lee, and P. Yalagandula. Automated and scalable qos control for network convergence. *Proc. INM/WREN*, 10:1–1, 2010.

[7] D. McDysan. *QoS and traffic management in IP and ATM networks*. McGraw-Hill, Inc., 1999.

[8] R. Mortier, B. Bedwell, K. Glover, T. Lodge, T. Rodden, C. Rotsos, A. W. Moore, A. Koliousis, and J. Sventek. Supporting novel home network management interfaces with openflow and nox. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 464–465. ACM, 2011.

[9] K. Nam-Seok, H. Hwanjo, P. Jong-Dae, and P. Hong-Shik. Openqflow: Scalable openflow with flow-based qos. *IEICE transactions on communications*, 96(2):479–488, 2013.

[10] P. Newman, W. Edwards, R. Hinden, E. Hoffman, F. C. Liaw, T. Lyon, and G. Minshall. Ipsilon flow management protocol specification for ipv4 version 1.0. Technical report, 1996.

[11] F. Risso and I. Cerrato. Customizing data-plane processing in edge routers. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 114–120. IEEE, 2012.

[12] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield. Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 135–148. ACM, 2004.

[13] M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, and Y.-q. Song. FlowQoS: Per-Flow Quality of Service for Broadband Access Networks. *Proceedings of the third workshop on Hot topics in software defined networking - HotSDN '14*, pages 207–208, 2014.

[14] N. Williams and S. Zander. Real time traffic classification and prioritisation on a home router using diffuse. Technical report, CAIA Technical Report 120412A, 2011.