

SDN enabled QoS Provision for Online Streaming Services in Residential ISP Networks

Irena Trajkovska[†], Philipp Aeschlimann[‡], Christof Marti[‡], Thomas Michael Bohnert[‡], Joaquín Salvachúa[†]

[†] Universidad Politécnica de Madrid, 28040 Madrid, Spain

[‡] Zurich University of Applied Sciences, 8401 Winterthur, Switzerland

Abstract—In this paper we present an idea of a proprietary Software Defined residential Network (SDrN) and we show as a use case, a multicast streaming service that can be hosted on such networks. To verify the feasibility of the service in the context of quality of service, we offer to the providers of online streaming services (in some cases the ISPs themselves), APIs to control and validate the QoS of the users in the service. The QoS control APIs were tested on SDN based simulation environment.

Index Terms—sdn, openflow, qos, streaming

I. INTRODUCTION

Various Internet Service Providers (ISPs) today rely on their own network infrastructure to offer Internet TV services. Those services require costly dedicated servers and constant load balancing to deal with flash crowds during peak hours.

Thanks to the Software Defined Networking (SDN) [1] paradigm, the ISPs can implement more flexible infrastructure and take over the network control in order to offer better and optimal services to the users. SDN delegates the control of the network to a software component inside a switch, enabling transparent interaction with the network resources according to the application's preferences. This is enabled thanks to the OpenFlow protocol that defines a communication interface between the control and the network plane. Any software that supports OpenFlow, can be used to control the forwarding decisions in OpenFlow configured network devices. Following these directives, the ISPs can deploy online streaming services with critical quality of service (QoS) demands over dedicated SDN infrastructure, while maintaining the rest of the services run over conventional network.

In this paper we discuss SDN based QoS management for online streaming services deployed over residential networks. To demonstrate the technical feasibility, we propose APIs to adjust the users' QoS by using the OpenFlow protocol. As a use case, we base our proposal on P2P multicast scenario aimed for streaming services that may note a big benefit from the solution.

Most of the proprietary TV services charge their users a flat-rate price for unlimited real-time/on-demand videos with best-effort QoS. In a previous work, we studied a P2P streaming service with variable prices depending on the QoS levels. Motivated from this idea, we study here the SDN possibility to implement such a service in practise. The opportunity for QoS management in the OpenFlow protocol, have triggered some research efforts to approach the QoS problem from a network level. As example, a QoS driven management was

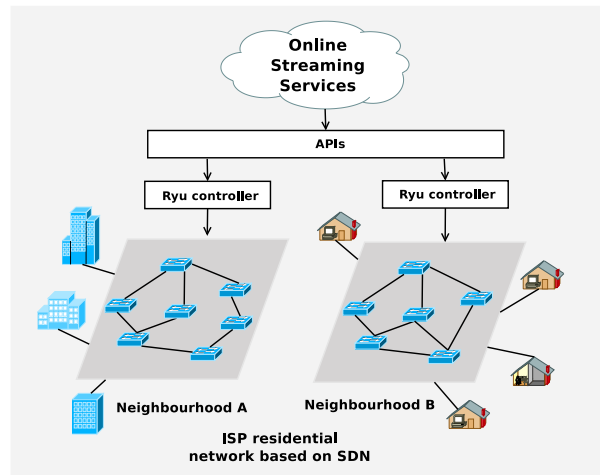


Fig. 1. SDN based residential network for dedicated streaming services

studied by Yiakoumis et al. [2]. The authors argue that the user's preferences on the application should guide the network traffic from home to the ISP network. Although similar to our idea for SDN based QoS adjustment, the proposal differs as it the focuses of direct user-ISP interaction and on-demand QoS change, while we leave the QoS maintenance responsibility to the content providers/ISPs, once the service preference has been done. Similarly, the authors in [3] propose SDN prototype architecture for future dynamic QoS provision in end-to-end applications. Next we describe the residential streaming service architecture.

II. RESIDENTIAL SDN STREAMING SERVICE

Figure 1, shows a use case scenario of SDN-based residential topology. Instead of using an already existing network infrastructure, the ISP deploys a separate SDN solution dedicated to online streaming service. Each neighborhood has SDN-enabled network, supervised by an OpenFlow-configured controller. If based on a client-server solution, a real-time service delivery requires very stable and scalable underlying topology that is expensive. As alternative, the P2P topology for media delivery is becoming increasingly present in real-time services, despite the challenges for constant QoS assurance.

Let's say that the provider offers the streaming service with three types of prices (highest to lowest): gold, silver and bronze based on the video quality. To provide scalable QoS

offer on application level, the provider can slice the video into smaller portions. Then he either pushes the sub-streams to the users directly from the video source or disseminates them in P2P multicast fashion, adapting the sent sub-streams quantity to the desired video quality. By using the SDN paradigm, the provider can achieve the same effect on a network level and provide the desired QoS by adjusting the underlying network topology between the source and the users, or between two users in the multicast. Although the P2P algorithms show significant improvement lately, when it comes to multimedia streaming, there is no standardized QoS protocol for P2P multicast. We part form this gap and the SDN advantage in order to leverage the P2P service from our previous work [4] and propose APIs for QoS management in streaming multicast services.

The advantage of the SDN based QoS control, is the possibility to offer end-to-end quality for real-time video delivery that is tuned down on a network level. This approach abstracts from the complex P2P overlay algorithms on the top of the network infrastructure. Instead, it shifts the interconnection decisions to the SDN control panel, thus enabling a direct mapping between the users' virtual and physical connections. Moreover it provides a better management of the physical topology with direct QoS supervision. The drawback of this solution is its current limitation to work only in a proprietary SDN infrastructure – enclosed in a residential ISP network or a private cloud deployment. For the moment, there is undergoing initiative of the switch producers to facilitate their solutions with OpenFlow support. Yet a global coordination and control interface across multiple SDN domains is about to come.

III. APIs FOR QoS MANAGEMENT

We propose QoS control APIs for residential SDN networks. As previously discussed, these APIs can be adapted by the streaming providers for QoS management in online streaming services or used by the ISPs for QoS adjustment on a network level. We describe the APIs in the context of streaming service scenario. When a user connects to online service to watch a streaming video, he sets up the desired QoS and his willingness to participate in a P2P multicast.

- (1) `get_connected_users()`
- (2) `get_closest_nodes_list()`
- (3) `get_current_bw(newNode, [closestNodes])`
- (4) `connect(newNode, hostNode)`
- (5) `adjust_bw(newNode, closestNode)`

Using the *Mininet 2.1* [5] emulator, we simulated a network of nodes connected to an OpenFlow 1.3 Software Switch. The switch implements a *Ryu* [6] controller application. Ryu is SDN framework that supports various protocols for managing network devices.

Upon this negotiation, the provider invokes API (1) to retrieve a list of the connected users. Afterwards, the candidate nodes are calculated, that are close to the new node by using the *ping* tool in Mininet for the *RTT* distance between the new node and the rest of the nodes, API (2). All the nodes that respond within a delay specified in the negotiated interval, are chosen as potential host nodes for the new node. The API

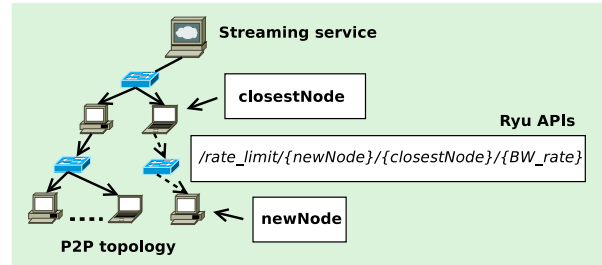


Fig. 2. Adjusting BW between nodes in a multicast tree using OpenFlow

(3) calculates the bandwidth between the new node and the potential nodes. The node that has a BW value that fits within the requested interval, becomes a host for the new node, API (4). Otherwise with API (5), the algorithm adjusts the BW between the new node and the closest node to match with the requested value, as shown on Figure 2.

The BW control algorithm bases on the OpenFlow *Meter Bands* [1]. The basic idea is to slice the maximal bandwidth a device can offer into one or more rate-limiters. The controller application detects any new device connected to the network and adds in its initial state, a rate-limiter associated to that device. This rate limiter is equal to the total amount of bandwidth that the device can offer. We validated the correctness of the API calls by invoking REST calls to retrieve the current QoS state. Testing the APIs with the use case scenario, showed the SDN approach to be convenient for QoS control in P2P multicast environment. The described proposal is aimed to motivate the ISPs to deploy a complete SDN solution within different neighborhoods of a city, as a host infrastructure for dedicated online streaming services. The SDN approach optimizes the use of the infrastructure resources, enabling the ISPs to host QoS critical and elastic services.

IV. CONCLUSIONS

We proposed an idea of SDN based residential network and APIs for QoS management in a streaming service to be implemented in similar SDN-enabled networks. As a use case, we applied the APIs to a multicast service and proved the feasibility to offer a network level QoS adjustment for P2P streaming services.

REFERENCES

- [1] "OpenFlow 1.4.0 Specification," Apr. 2014. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf>
- [2] Y. Yiakoumis, S. Katti, T.-Y. Huang, N. McKeown, K.-K. Yap, and R. Johari, "Putting home users in charge of their network," in *Proc. of 2012 ACM Conference on Ubiquitous Computing*, ser. UbiComp '12. New York, NY, USA: ACM, 2012, pp. 1114–1119.
- [3] I. Bueno, J. Aznar, E. Escalona, J. Ferrer, and J. Antoni Garcia-Espin, "An opennaas based sdn framework for dynamic qos control," in *Future Networks and Services, 2013 IEEE SDN for*, Nov 2013, pp. 1–7.
- [4] I. Trajkovska, P. Rodríguez, J. Cerviño, P. Harsh, and J. Salvachúa, "P2P Incentive Model for QoS based Streaming Systems," in *Proc. of the Int. conference on Consumer Communications and Networking*, ser. CCNC '14. IEEE, 2014.
- [5] "Mininet," Apr. 2014. [Online]. Available: <http://mininet.org>
- [6] "Ryu," Apr. 2014. [Online]. Available: <http://osrg.github.io/ryu/>