Codec-Aware Video Delivery Over SDNs

Obinna Izima, Ruairí de Fréin, Ali Malik
Technological University Dublin, Ireland

***Abstract*—To guarantee quality of delivery for video streaming over software defined networks, efficient predictors and adaptive routing frameworks are required. We demonstrate an agent that predicts video quality of delivery metrics in a scalable way using a bespoke codec-aware learning model. We also demonstrate the integration of this agent with an adaptive framework for centrally controlled software-defined networks that re-configures network operational paths in response to the learning agent, ensuring that good quality of delivery of video is maintained during periods of congestion. The demo scenario highlights the feasibility, scalability and accuracy of the framework.**

***Index Terms*—SDN, codec, monitoring, quality of delivery**

I. Introduction

Predicting the quality of video delivered from cloud-hosted services is challenging. The problem's complexity lies in the nature of the delivery system, which multiplexes heterogeneous user demands and services-types over a finite, shared network, compute and storage substrate. Recent attempts to model causes of adaptivity have been promising: a companion paper [1] demonstrates the efficacy of attempting to model the behaviour of the video codec in an off-line learning algorithm; the approach in [2] incorporates the effects of changing numbers of user requests on predicting real-time service-level metrics from device statistics.

This demonstration showcases a working system, which will be of interest to practitioners as it integrates recent breakthroughs in the area of data analytics for management and also programmable networks and automation. Given that video streaming is predicted to soon contribute 82% of Internet traffic [3], we consider a video-over Software-Defined Networking (SDN) scenario. Delivering video is challenging due to rapid bandwidth fluctuations and time-varying delays. The ability to estimate the Quality of Delivery (QoD) of the received video at end users in a scalable way is important if service providers are to meet service level of agreements [4]; however, deciding what to do with this information in real-time is an important question. Contributions have been made to improve video delivery [5], however, deploying more intelligence in a traditional, rigid IP Network architecture is hard. The potential gains of run-time learning approaches has not been realised.

To achieve responsive, in-network learning we take the approach of modeling the behaviour of the video codec deployed by the server. Modeling the codec, as opposed to instrumenting it, makes our approach more portable. An offline evaluation of the approach is analyzed in [1]. Modern codecs adapt based on their perceived view of the network state; they then predict future network performance and adapt the format of the video they inject into the network. However, information resulting from this codec adaptation and the prediction information is not used for path adaptation. The siloed nature of video codec adaptation algorithms and SDNs proactive management of traffic flows in [4], presents the opportunity for functionality integration, which we demonstrate here. The information that the codec has decided to adapt its output is useful because (1) the codec makes an assumption about the network and how it thinks it will behave and (2) it adapts its own behaviour. We demonstrate that an SDN framework that re-configures network operational paths based on this intelligence is better able to guarantee QoD during periods of congestion. SDN reduces the barrier to embedding intelligence inside networks. Advances from the perspective of routing, quality of service (QoS), traffic classification and prediction are reviewed in [6]

II. Scenario and Testbed

This demonstration shows our ability to predict and update flow-tables, to maintain the QoD of video content. In the testbed, a client machine streams H.264 and H.265 video from a VLC server over RTP/UDP. The QoD metric, jitter, is obtained from a client machine. The client and server communicate over an SDN which has the ability to reconfigure flow tables in response to congestion. These remedial actions act to keep the network at a desired QoD level. Due to the fact that network congestion is the primary cause of performance degradation and performance variability for time-sensitive applications like video, we focus on one of the effects of congestion, e.g. jitter, and predict future jitter values in order to inform the SDN Controller for future flow-table updates. The proposed framework is implemented and evaluated using the container-based emulator, Mininet. In the emulation environment, we employ two servers; one acts as the OpenFlow controller and the other simulates the network topology. For each server, we used Ubuntu v.14.04 LTS with Intel Core-i5 CPU and 8 GB RAM.

The Distributed Internet Traffic Generator (D-ITG) is used to generate interfering network traffic. Different levels of congestion are invoked by the D-ITG to cause the target client's QoD to change during different epoches of low to high congestion. D-ITG produces realistic, packet-based network traffic by accurately emulating the workload of real world traffic and current Internet applications. We generate different numbers of ICMP flows sequentially between the designated hosts that share the SDN with the video client and server, for the course of each epoch. The packet sizes and the inter-departure times are constant and adjusted during each epoch.

III. System Architecture

The three main components of the demo architecture are shown in Fig. 1. They are the topology discovery and statistics

978-3-903176-32-4 © 2021 IFIP

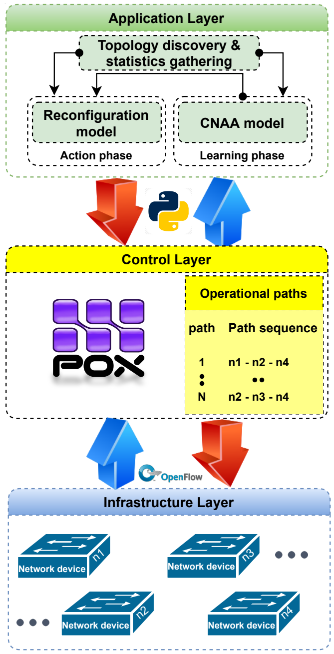

Fig. 1. System architecture and its components: the primary contribution lies in the integration of the learning and action blocks. Openflow is used on the southbound interface and POX APIs are used on the northbound interface.

gathering, the learning phase and the action phase components.

Topology Discovery and Statistics Gathering Component: Builds a topological view of the underlay infrastructure as well as periodically collecting information about the network traffic flow. The standard OpenFlow protocol [7] is used to transfer the collected data between the data and control layers.

Learning Phase Component: Typically, the transport protocol is responsible for gauging the network capacity depending on some metrics such as the congestion level and the round trip time. This network capacity is utilised by the codec as a metric to determine the most appropriate compression level for the video, so that it can adaptively relay the video stream to the destination. A Codec-aware Network Adaptation Agent (CNAA) uses a light-weight online learning strategy for estimating jitter, when the delivery system uses an adaptive video codec. CNAA achieves accurate estimation of jitter by estimating what the codec will decide to do next, as codec adaptation is often the dominant factor in the time varying nature of QoD time-series. By modeling codec behaviour, the resulting learning agent is accurate, and has linear time complexity in the number of training samples, N, which makes it ideal for real-time learning.

Action Phase Component: Exploits the information learned from the CNAA so that QoD of video is improved. Its primary objective is to reduce the effect of predictions of future jitter from all traffic by steering these flows away from elements that are expected to experience congestion. To this end, the SDN controller maintains a list of operational paths. It dynamically re-adjusts the flow-table of SDN forwarding elements according to the CNAA predictions. The action phase

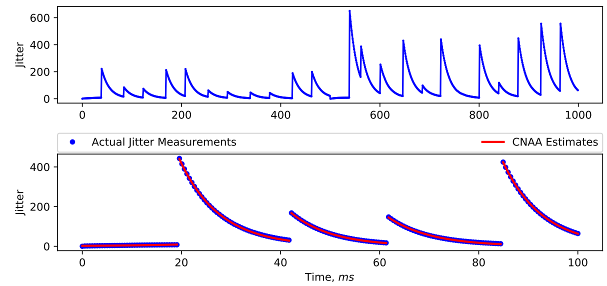

Fig. 2. Row 1 (R1) illustrate traces depicting the adaptive behaviour of the H.264 codec in response to varying levels of network congestion. Row 2 (R2) shows the accuracy of the CNAA estimation model. The estimations obtained are overlaid on the actual jitter measurements captured via Wireshark.

strives to avoid the impact of jitter by proactively updating the list of operational paths. In the setup in Fig. 1, we have developed an interface between the CNAA model outputs and the reconfiguration model. In our framework, the CNAA prediction estimates are passed to the reconfiguration model to initiate an action based on the CNAA estimates.

IV. Demonstration

The demonstration shows the predictions of QoD metrics, the accuracy of those predictions and the scalability of the predictor. The learning agent displays real-time measurements of jitter from a target VLC client. For example, in terms of the accuracy of the CNAA estimation model, Fig. 2 shows preliminary results for a client-server jitter trace. The CNAA model accurately estimates the curve heights, decay factor and the time-varying periods which are characteristic of the jitter time-series captured between a video client and server, making future jitter predictions possible. Fig. 2, R2 shows the accuracy of the CNAA estimation model. In addition, it shows the weighting of the flow-table reconfiguration system. Based on the historical jitter measurements, collected from the platform it shows whether or not a re-configuration is recommended. Predictions of the QoD metrics are given for different congestion levels for the system.

References

[1] O. Izima, R. de Fréin, and M. Davis, "Predicting quality of delivery metrics for adaptive video codec sessions," *IEEE CloudNet*, 2020.

[2] R. de Fréin, "Source separation approach to video quality prediction in computer networks," *IEEE Com. Letr.*, vol. 20, no. 7, pp. 1333–1336, 2016.

[3] "Cisco annual internet report (2018–2023) white paper," Cisco, Tech. Rep., March 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[4] A. Malik and R. de Fréin, "SLA-aware routing strategy for multi-tenant software-defined networks," in *IEEE ISCC*, pp. 1–7.

[5] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Com. Surv. & Tut.*, vol. 17, no. 1, pp. 469–492, 2014.

[6] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (sdn): Research issues and challenges," *IEEE Comm. Surv. & Tut.*, vol. 21, no. 1, pp. 393–430, 2018.

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Comp. Com. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.