Streaming Scalable Video Sequences with Media-Aware Network Elements Implemented in P4 Programming Language

Guan-Ru Wang[1], Chien Chen[1], Chao-Wen Chen[2],
Li-Wen Pan[2], Yu-Rong Wang[2], Ching-Ling Fan[2], and Cheng-Hsin Hsu[2]
[1]Department of Computer Science, National Chiao Tung University, Taiwan
[2]Department of Computer Science, National Tsing Hua University, Taiwan

***Abstract*—We present the first Media-Aware Network Element (MANE) for intelligently streaming scalable video sequences in P4 programming language. Our MANE selectively drops queued scalable video packets when the queue occupancy exceeds a threshold. Three packet discarding logics are implemented: (i) tail, (ii) enhancement-layer, and (iii) rate-distortion optimized. Our P4-based MANE implementation is demonstrated in: (i) larger emulated networks in mininet with P4 software switches and (ii) a small real network with a physical P4 switch and multiple Raspberry Pis running P4 software switches.**

***Index Terms*—Scalable video streaming, media-aware network element, software-defined network, rate-distortion optimization**

I. Introduction

Over-The-Top (OTT) video streaming services, like Apple TV, Netflix, and Hulu, have become very popular, e.g., the streaming device market is projected to exceed 25 billion USD by 2024, with a 15%+ annual growth on unit shipments [1]. These OTT services offer 4k resolution video sequences, and thus incur high traffic amount on the best-effort Internet. *Scalable Video Coding (SVC)* is a promising solution to mitigate such excessive traffic: each scalable video sequence contains a base and multiple enhancement layers, while higher enhancement layers can be dropped without affecting its decodability. The dynamic decisions on which video packets to drop can be sub-optimally done by streaming servers or clients without the global knowledge on the network. A better way for making decisions is through *Media-Aware Network Elements (MANEs)*, which are switches that have access to the video-related packet headers and local network conditions, and thus can make better decisions. By collaboratively considering multiple OTT video streams sent across a set of interconnected MANEs, an even higher video quality level is possible.

Unfortunately, replacing regular switches with MANEs is quite expensive and therefore less likely to happen. However, with recent advances in *Software-Defined Networking (SDN)* and *Network Function Virtualization (NFV)*, network switches are much more *programmable*, which in turn make collaborative MANEs into a reality. A recent market report points out that the SDN/NFV investments from Internet Service Providers (ISPs) are expected to grow at an annual rate of 45% between 2017 and 2020 [3]. In other words, programmable switches are going to be deployed in the Internet scale, and *these switches can be programmed into collaborative MANEs for optimally streaming scalable video sequences.*

In this work, we prototype the very first MANE in P4 programming language [4] for scalable video streaming. P4 is designed for realizing different network protocols (in software) on packet processors that can forward packets at the line speed. We implement intelligent media-aware packet drop logics in packet processors of P4 switches, and we connect multiple P4 switches to an ONOS controller through P4 runtime for a network of collaborative MANEs. We demonstrate our prototype system using mininet or a few real P4 switches to show its practicality. Real scalable video sequences are created and used in our demonstrations, while several demonstration scenarios are used to show the merits of our system.

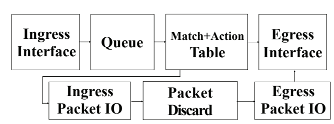

Fig. 1. Packet processing in a P4-based MANE.

II. Media-Aware Network Element

We show how to intelligently drop scalable video packets to retain streamed video quality, when bandwidth is insufficient and dynamic. We consider three *packet discarding logics*: (i) tail, (ii) enhancement-layer (EL), and (iii) rate-distortion optimized (RDO). Tail always drops the last packet in the queue, while EL only drops the enhancement-layer packets when needed. The advantage of tail logic is simplicity, while EL logic ensures the decodability of received videos. RDO logic is more comprehensive, as it takes the nonlinear nature between distortion and bitrate into considerations. By dropping the packets with the least distortion and bitrate ratio, RDO logic aims to minimize the negative impacts of dropping packets on video quality.

This work was partially supported by the Ministry of Science and Technology of Taiwan under the grant #106-2221-E-007-101.

We have implemented a MANE in P4 programming language [4]. Fig. 1 shows how the P4-based MANE processes the packets. First, we define the header formats and parsers, which allow us to understand the structure of individual packets. We use the information in headers to check if packets are valid. Next, we define tables to describe how the header fields should be matched in the match+action stage. The control program pushes the packets into one of the queues based on their header fields. It starts to throttle the ingress packets when the queue occupancy exceeds a admin-specified threshold. More specifically, the *packet discard logic* picks and drops some packets, in order to maintain high overall video streaming quality; other packets are forwarded.

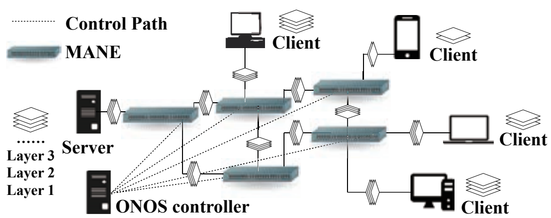

Fig. 2. High-level system architecture with a network of MANEs.

III. System Architecture

The considered system is composed of an ONOS controller, a video server, multiple video clients, and several MANEs as shown in Fig. 2. The ONOS controller forms a control plane disassociated from data plane among P4 switches. Using the knowledge on the network topology and available resources, the ONOS controller determines the best routes from servers through MANEs to clients. Once the streaming starts, each MANE drops the scalable video packets when necessary. The life cycle of the video streaming sessions is as follows. First, the receiver sends a request to the server, and the server sends the encoded video streams through the MANEs. During transmissions, the ONOS controller communicates with all MANEs to set up the outgoing port for packets. When the network topology changes, the ONOS controller informs every MANE for the alternative paths immediately. The ONOS controller may also fine-tune the video quality levels of different users for optimal overall quality.

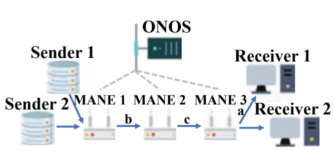

Fig. 3. Mininet testbed topology (scenarios 1 and 2).

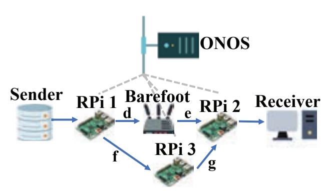

Fig. 4. Real testbed topology (scenario 3).

IV. Demonstration Setup

We use real H.264/SVC video sequences in our demonstrations. We develop three scenarios, where the first two scenarios are done in mininet with software MANEs (running the reference P4 software switch, called *bmv2* [2]) and virtual hosts. We deploy real MANEs: a Barefoot switch and three Raspberry Pi ones (with bmv2) in the last scenario, in order to show the practicality of our prototype system. We describe the three scenarios below:

- **Intelligent packet drops of a single video stream.** We construct a simple network with a sender, a controller, a MANE, and a receiver. The network is similar to Fig. 3 but with only a pair of sender/receiver and a MANE. The mininet bandwidth on the link is varied over time by scripts. In the MANE, we implement tail, EL, and RDO logics, and compare their performance.
- **Optimal packet drops across multiple video streams.** We construct a network with two pairs of senders and receivers, and three MANEs, as shown in Fig. 3. We vary the bandwidth of link c over time. MANE 2 determines which packets to drop to maximize the overall quality. Moreover, MANE 2 notifies MANE 1 to drop the packets (early) to avoid bandwidth wastes on link b. We evaluate different packet-discarding logics and video sequences.
- **Error resilience with real P4 switches.** The network we construct includes a sender, a controller, one Barefoot switch, three Raspberry Pi switches (with bmv2), and a receiver, as revealed in Fig. 4. All the links are Fast Ethernet cables. The bandwidths of links d and e are higher than the bandwidths of links f and g. We physically remove link e during streaming, and we expect to see that the ONOS controller reroutes the video stream through the Raspberry Pi 3 at the bottom. Because we configure links f and g to have lower bandwidth, the video quality goes down. When we plug the Ethernet cable back, the ONOS controller reroutes the video stream over the Barefoot switch, and the quality of video stream goes up.

V. Conclusion and Outlook

We have demonstrated the feasibility of realizing MANE for scalable video streaming using P4 programming language. Through the high programmability offered by software-defined networks, we show how optimally dropping scalable video packets improves the streaming quality, while many other optimization approaches are possible in the future. For example, video and non-video Internet services can be managed in a Quality-of-Service fair way, and popular video packets can be proactively or reactively cached.

References

[1] Streaming devices market to hit $25bn by 2024: Global market insights, inc. https://www.gminsights.com/industry-analysis/streaming-devices-market, December 2017.

[2] Github repository of the second version of the P4 software switch (a.k.a. behavioral model), nicknamed bmv2. https://github.com/p4lang/behavioral-model, January 2018.

[3] World SDN, NFV & Network Virtualization ecosystem research report 2017-2030 - $22 billion market opportunities, challenges, strategies & forecasts. http://www.snstelecom.com/sdn-nfv, January 2018.

[4] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95, July 2014.