Assessing Video Quality in AlphaRTC via VMAF and Throughput Analysis

Xinnian Li*, Zijun Peng†, Bairui Li‡, Sijie Li§, Zhang Zhang¶
*2200012910@stu.pku.edu.cn, †2200012909@stu.pku.edu.cn, ‡2300011094@stu.pku.edu.cn,
§2200012943@stu.pku.edu.cn, ¶2200012902@stu.pku.edu.cn

***Abstract*—This report evaluates the quality of real-time video communication over AlphaRTC, an open-source platform leveraging reinforcement learning for congestion control. We analyze the system's video transmission fidelity using VMAF (Video Multi-Method Assessment Fusion) and measure throughput from packet-level logs.**

I. Introduction

AlphaRTC [1] provides a flexible environment for real-time video communication research. It integrates learning-based congestion control algorithms and supports custom packet-level instrumentation. To assess the subjective and objective quality of AlphaRTC's video transmission, we apply the VMAF metric [2] and compute throughput from transport logs.

II. Methodology

Our evaluation involves preparing the video input, running AlphaRTC in a containerized environment, and computing two metrics: VMAF score for video quality and throughput in KB/ms.

A. Preprocessing Video Input

The source video is preprocessed using `ffmpeg`:

- Resize to 320×240:

```
ffmpeg -i usain_bolt.mp4 -vf
↪   "scale=320:240,setdar=4/3"
↪   usain_bolt_0.mp4
```

- Extract seconds 11–26:

```
ffmpeg -ss 00:00:11 -to 00:00:26 -i
↪   usain_bolt_0.mp4 usain_bolt_1.mp4
```

- Reduce frame rate to 10 fps:

```
ffmpeg -i usain_bolt_1.mp4 -filter:v fps=10
↪   usain_bolt_10.mp4
```

- Convert to YUV and WAV:

```
ffmpeg -i usain_bolt_10.mp4 usain_bolt.yuv
ffmpeg -i usain_bolt_10.mp4 usain_bolt.wav
```

B. AlphaRTC Execution

The platform requires JSON configs for sender and receiver. We launch AlphaRTC using Docker and log packet statistics by modifying the bandwidth estimator:

```python
class Estimator(object):
    def report_states(self, stats: dict):
        with open("stats.txt", "a") as f:
            f.write(str(stats) + "\n")
```

Launch commands:

```
sudo docker run -d --rm -v <abs_path>:/app -w /app
↪   \
  --name alphartc --platform linux/amd64 \
  alphartc peerconnection_serverless receiver.json

sudo docker exec alphartc peerconnection_serverless
↪   sender.json
```

C. Video Quality Evaluation

Since AlphaRTC may repeat video frames, we select a 30-frame segment for comparison using VMAF:

```
vmaf --reference usain_bolt.yuv --distorted
↪   out_usain_bolt.yuv \
  --width 320 --height 240 --pixel_format 420
  ↪   --bitdepth 8 \
  --model version=vmaf_v0.6.1 --output output.xml
  ↪   --frame_cnt 30
```

D. Throughput Computation

From `stats.txt`, we compute the effective transmission throughput:

```python
total_time_ms = 0
total_payload_size = 0
last_time = None
stats = [eval(x) for x in open("stats.txt")]

for entry in stats:
    t = entry["arrival_time_ms"]
    if last_time is not None:
        total_time_ms += t - last_time
    total_payload_size += entry["payload_size"]
    last_time = t

print(total_payload_size * 1000 / total_time_ms)
```

III. Results

The VMAF score averaged approximately 8.27 across the sampled segment, indicating notable degradation due to compression and frame skipping. The throughput measured was around 21.89 KB/ms.

IV. Conclusion

By integrating AlphaRTC with VMAF and custom throughput tracking, we demonstrated a repeatable methodology for analyzing real-time video delivery performance. The low VMAF suggests trade-offs in resolution and bitrate need further tuning for better perceptual quality.

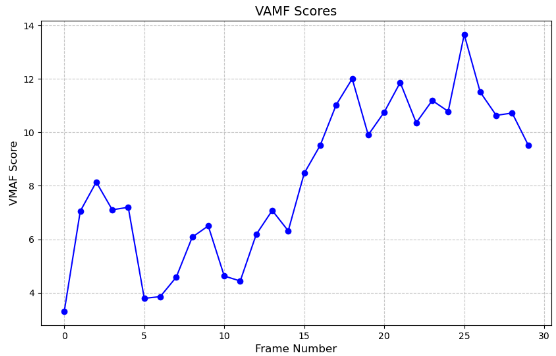

Fig. 1: Frame-wise VMAF scores from output.xml

REFERENCES

[1] J. Eo, Z. Niu, W. Cheng, F. Y. Yan, R. Gao, J. Kardhashi, S. Inglis, M. Revow, B.-G. Chun, P. Cheng, and Y. Xiong, "Opennetlab: Open platform for rl-based congestion control for real-time communications," in *Proceedings of the 6th Asia-Pacific Workshop on Networking*, 2022, pp. 70–75.

[2] N. Technology, "Toward a practical perceptual video quality metric," https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, 2016.