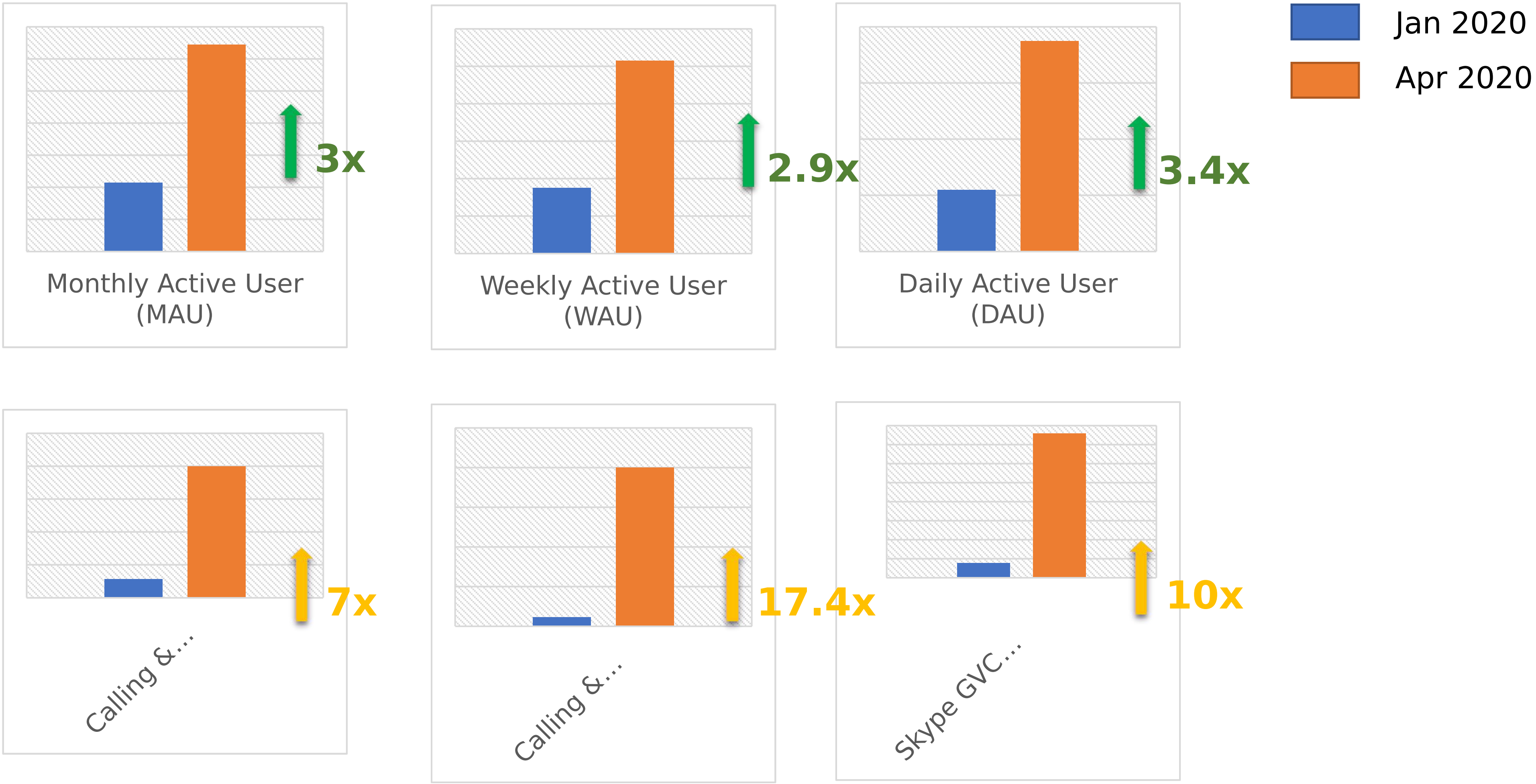
Bandwidth Estimation on OpenNetLab

Zhixiong Niu
on behalf of OpenNetLab community

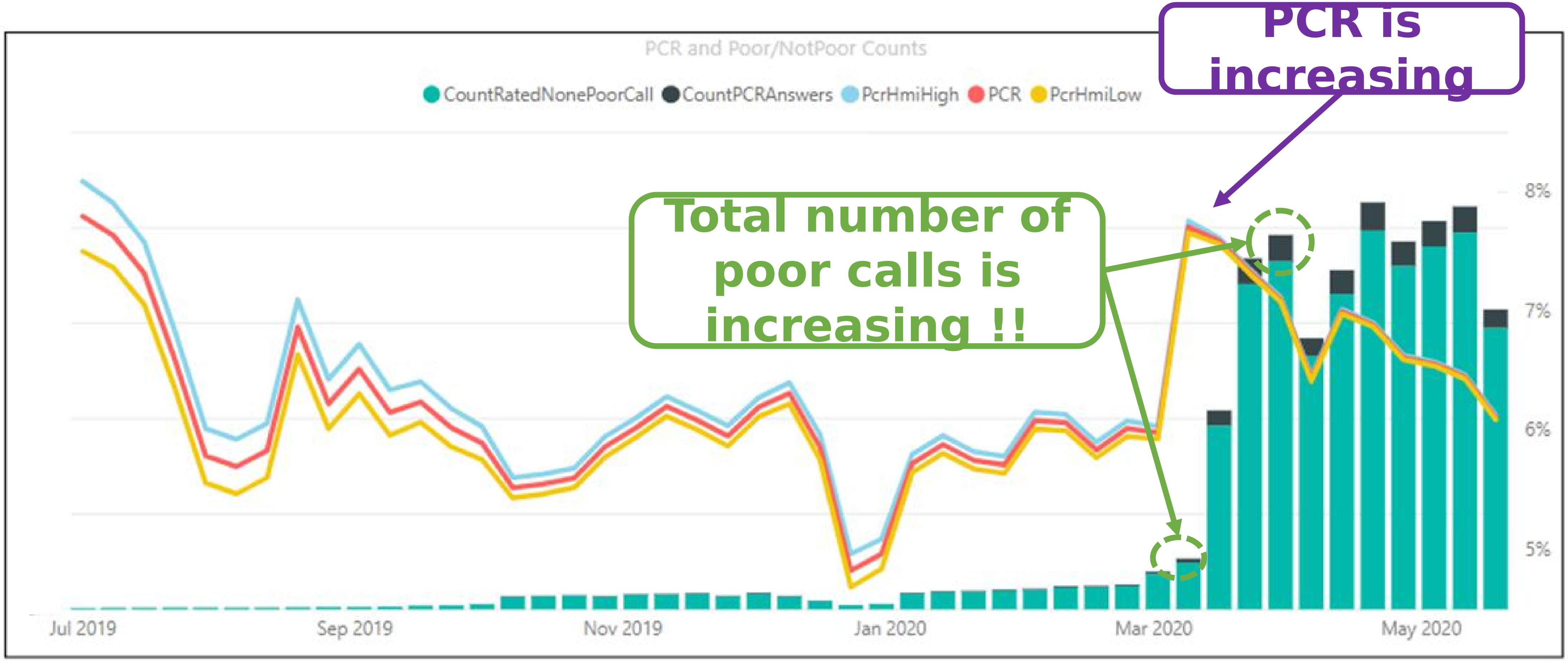
RTC is Growing Super Fast

Active User

Calling & Meeting



Most Critical KPI: Poor Call Rate (PCR)



One of Key Reasons for PCR - Bandwidth Estimation

Poor Calls for 1:1 Call

28.9% Poor 1:1 Calls are highly related to bandwidth co

40.9% Poor 1:1 Calls are related to bandwidth control

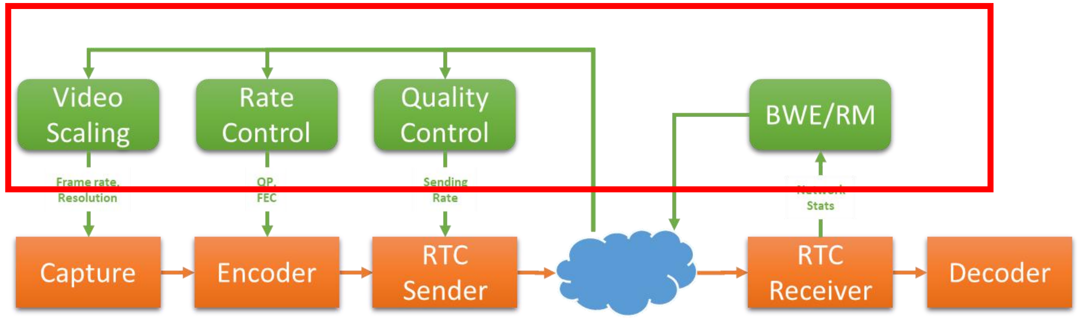
Highly-related
Related

	Problem token	% tokens	Top reasons
1	No sound	22.7%	Device selection, device issues, network loss/jitter, limited BW
2	**Distorted audio**	**14.6%**	**Network loss/jitter, limited bandwidth or control**
3	Background noise	12.8%	Background noise, mic/ADSP issues, network loss/jitter
4	Acoustic echo	8.5%	Device acoustics, non-linear loudspeaker effects, cascaded audio processing
5	Audio loudness low	6.6%	Microphone issues, lack of device gain control, device selection
6	**Audio delay**	**6.1%**	**Network RTT/jitter, bandwidth control**
7	Call dropped	5.4%	Network loss, network device lost, app crash

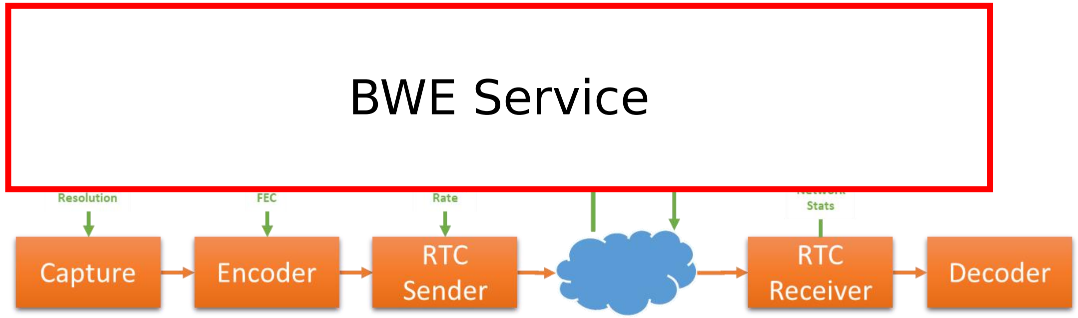
Can BWE be a service?

Traditional BWE
- Proprietary
- Single model for all users
- Hard to innovate

Standard BWE Service
- Simpler architecture
- Enable more customization
- Everyone can contribute to this service and can share the service



Traditional Bandwidth Control



BWE as a service

MMSys ‘21 BWE Challenge

Goal: Optimize QoE for real-time communications (RTC)

- video and audio quality, video frame drop rate and delay, etc.

Key algorithm: bandwidth estimation (BWE)

- computes a bandwidth estimate dynamically based on network stats
- passes the estimate into video codec to control the encoded bitrate

Heterogeneous real networks make data-driven approaches a good fit

- BWE can be modeled as a reinforcement learning problem

Challenge results

Rank	Score	Paper Title	Institute	Team Members
Winner	78.33	Gemini: An Ensemble Framework for Bandwidth Estimation in Web Real-Time Communications	Nanjing University	Tianrun Yin, Jiaqi Zheng, Runyu He, Shushu Yi, Hongyu Wu, Dingwei Li
Runner-up	67.96	A Hybrid Receiver-side Congestion Control Scheme for Web Real-time Communication *[accepted]*	Communication University of China	Bo Wang, Yuan Zhang, Size Qian, Zipeng Pan, Yuhong Xie
3	67.37	A Bandwidth Estimator Using Advantage Actor-Critic Algorithm	Peking University	Yunze Luo, Ting Lei
4	66.43	Bandwidth Estimation for Real-Time Communications with Reinforcement Learning	New York University	Siyuan Hong, Cheng Chen, H. Jonathan Chao, Chenyu Yen, Ke Chen, Xiaotian Li
5	62.50	Adaptive Bandwidth Estimation using Network Modeling	National University of Singapore	Yuan Li, Bingsheng He, Bryan Hool, Yuhang Chen
6	62.43	Bandwidth Estimation for Video and Audio Transfer using A2C	Peking University	Haipeng Zhang, Shenhan Zhu
Baseline	71.47	Google Congestion Control	WebRTC/Google	N/A

Can BWE as a part of the ALTO?

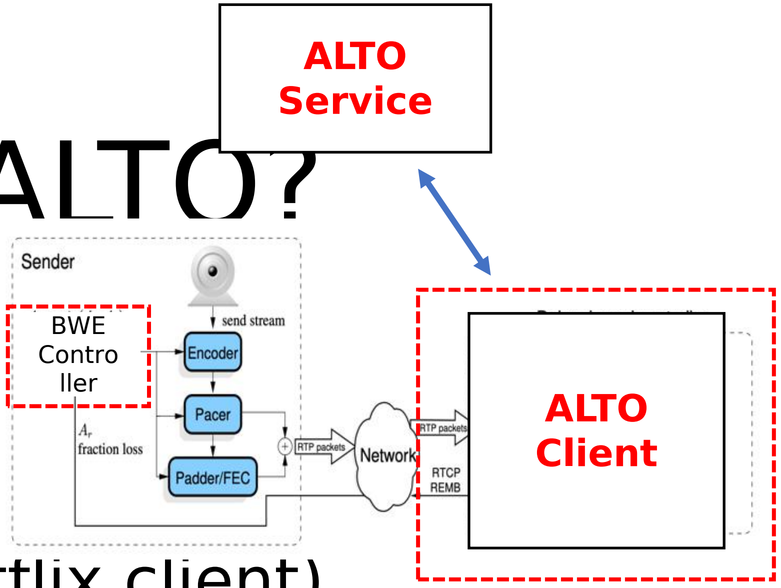


Fig. 1 ALTO in RTC

- Potential applications
  - RTC clients (Teams, Tencent Meeting, etc.)
  - Video streaming clients (Youbute client, Netflix client)
- Input
  - Packet states (send time, arrival time, seq, ssrc
  - ...
- Output
  - Estimated bandwidth to the sender

```python
class Estimator(object):
    def report_states(self, stats: dict):
        '''
        stats is a dict with the following items
        {
            "send_time_ms": uint,
            "arrival_time_ms": uint,
            "payload_type": int,
            "sequence_number": uint,
            "ssrc": int,
            "padding_length": uint,
            "header_length": uint,
            "payload_size": uint
        }
        '''
        pass

    def get_estimated_bandwidth(self)->int:
        return int(1e6) # 1Mbps
```

Fig. 2 Input and output

OpenNetLab Introduction

OpenNetLab (ONL)

The next generation platform for open and practical networking research

- **Heterogenous nodes**
  - VM, PM, desktop, laptop, smart devices
- **Real applications**
  - Real full-stack WebRTC application
  - Chrome/Edge
  - Iperf
  - Customized applications
- **Network in the wild**
  - Wired network: campus network, cloud network
  - Wireless network: Wi-Fi 5/6
  - Mobile network: 3G, 4G, 5G

OpenNetLab

Platform Building

Finished 37 nodes, and building 8 nodes

Org.	Location	Deployment Status
MSRA	Beijing, China	Finished: 8 nodes
PKU	Beijing, China	Finished: 6 nodes
LZU	Lanzhou, China	Finshed: 5 nodes Building: 1 nodes
NJU	Nanjing, China	Finished: 6 node
SUSTech	Shenzhen, China	Finished: 2 node Building: 1 node
SNU	Seoul, South Korea	Finished: 3 node Building: 3 nodes
KAIST	Daejeon, South	Finished: 3 node

Thank you

Backup Slides

Hard to improve in Current Bandwidth Control

10-year old technology

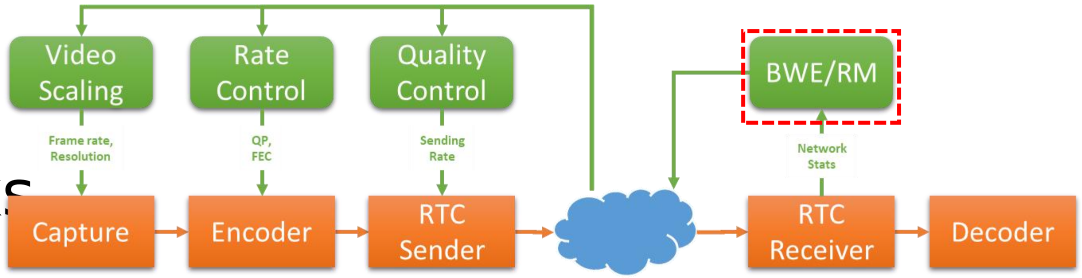
- Unscented Kalman Filter (UKF) in Resource Manager (BWE/RM)

Hard to tune

- 100's of heuristics to improve performance of Kalman filter
- Requires both network and codec experts with steep ramp-up time

Extremely hard to maintain

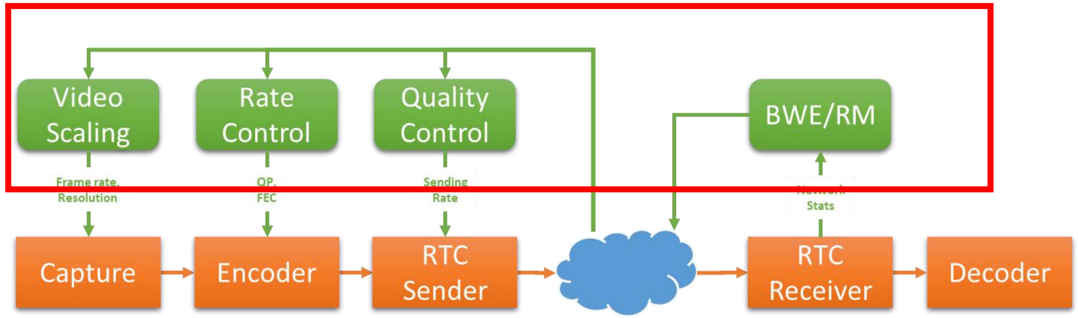
- >150K lines code for **green** blocks
- Need to be future-proof



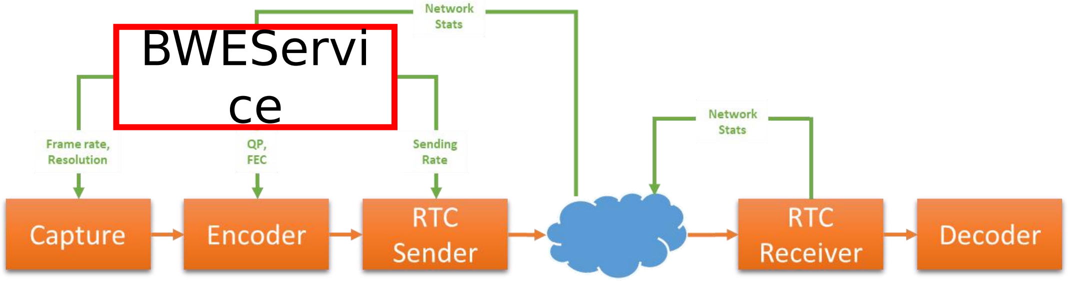
Bandwidth Control Components in Teams

Software 2.0: BWE as a Service for RTC

- Simpler architecture
- No hard-coded rules
  - Everything is automatically trained
- Much less domain expertise required
- New network/device support is automatic



Traditional Bandwidth Control



Ideal RL-based Bandwidth Control

Challenge framework

Simple interface to implement

- participants are only required to fill in a Python class
- executed as WebRTC's bandwidth estimator in AlphaRTC
- containerized runtime environment

Simulated environment to facilitate ML solutions

- AlphaRTC-Gym

Real-world testbed with automated evaluation

- OpenNetLab

```python
class Estimator(object):
    def report_states(self, stats: dict):
        '''
        stats is a dict with the following items
        {
            "send_time_ms": uint,
            "arrival_time_ms": uint,
            "payload_type": int,
            "sequence_number": uint,
            "ssrc": int,
            "padding_length": uint,
            "header_length": uint,
            "payload_size": uint
        }
        '''
        pass

    def get_estimated_bandwidth(self)->int:
        return int(1e6) # 1Mbps
```

Evaluation setup

405 runs per scheme on OpenNetLab

9 videos

- online video chat, remote desktop, etc.

3 networks

- High bandwidth (300–400 Mbps)
  - Lanzhou → Hong Kong; wired network
- Medium bandwidth (2–3 Mbps)
  - Beijing → Hong Kong; 4G network with competing flows
- Low bandwidth (<1 Mbps)
  - Beijing → Hong Kong; Wi-Fi in an isolation box

3 series of 5 runs per scheme in round robin

final score = average weighted sum of video score, audio score, and network score

Standardize the BWE Service

Location

Receiver side (Fig. 1)

Input

Packet states (send time, arrival time, seq, ssrc, etc.)

Output

Estimated bandwidth to the sender

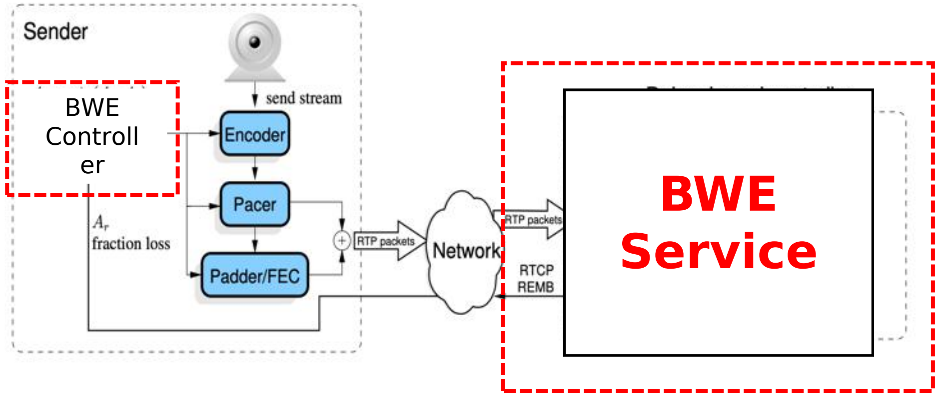


Fig. 1 BWE Model in AlphaRTC

```python
class Estimator(object):
    def report_states(self, stats: dict):
        '''
        stats is a dict with the following items
        {
            "send_time_ms": uint,
            "arrival_time_ms": uint,
            "payload_type": int,
            "sequence_number": uint,
            "ssrc": int,
            "padding_length": uint,
            "header_length": uint,
            "payload_size": uint
        }
        '''
        pass

    def get_estimated_bandwidth(self)->int:
        return int(1e6) # 1Mbps
```

Fig. 2 Input and output