Bandwidth Estimation on OpenNetLab

Zhixiong Niu
on behalf of OpenNetLab community
RTC is Growing Super Fast

Active User

- Monthly Active User (MAU): 3x
- Weekly Active User (WAU): 2.9x
- Daily Active User (DAU): 3.4x

Calling & Meeting

- Calling & Meeting: 7x
- Skype GVC: 17.4x
- Calling & Meeting: 10x
Most Critical KPI: Poor Call Rate (PCR)

- PCR is increasing
- Total number of poor calls is increasing!!

![Graph showing increasing PCR and poor calls]
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 control

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

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

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

Potential applications
RTC clients (Teams, Tencent Meeting, etc.)
Video streaming clients (Youbute client, Netflix client)

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

Output
Estimated bandwidth to the sender

Fig. 1 ALTO in RTC

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
Platform Building

Finished 37 nodes, and building 8 nodes

<table>
<thead>
<tr>
<th>Org.</th>
<th>Location</th>
<th>Deployment Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRA</td>
<td>Beijing, China</td>
<td>Finished: 8 nodes</td>
</tr>
<tr>
<td>PKU</td>
<td>Beijing, China</td>
<td>Finished: 6 nodes</td>
</tr>
<tr>
<td>LZU</td>
<td>Lanzhou, China</td>
<td>Finished: 5 nodes Building: 1 node</td>
</tr>
<tr>
<td>NJU</td>
<td>Nanjing, China</td>
<td>Finished: 6 node</td>
</tr>
<tr>
<td>SUSTec</td>
<td>Shenzhen, China</td>
<td>Finished: 2 node Building: 1 node</td>
</tr>
<tr>
<td>SNU</td>
<td>Seoul, South Korea</td>
<td>Finished: 3 node Building: 3 nodes</td>
</tr>
<tr>
<td>KAIST</td>
<td>Daejeon, South Korea</td>
<td>Finished: 3 node</td>
</tr>
</tbody>
</table>
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
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
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

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

Fig. 2 Input and output