OpenNetLab:
Open Platform for RL-based Congestion Control for Real-Time Communications

Jeongyoon Eo*, Zhixiong Niu†, Wenxue Cheng†, Francis Y. Yan†, Rui Gao†, Jorina Kardhashi‡, Scott Inglis‡, Michael Revow‡, Byung-Gon Chun*, Peng Cheng†, Yongqiang Xiong†

*Seoul National University †Microsoft Research ‡Microsoft
Real-Time Communications (RTC)

- Popularity and importance of RTC
  - Sharp increase during the COVID-19 pandemic
  - Video conferencing, live commerce, live streaming services

In 2025, RTC is expected to
- Take 17% of internet video traffic
- Make 217.3B $ revenue worldwide
Congestion Control (CC) for RTC

- Accurate bandwidth estimation is the key factor in achieving high QoE.

**Diagram:**
- **Sender**
  - Frames
  - Codec
  - bitstream
  - Pacer
  - Target Bitrate (= Estimated Bandwidth)
- **Concentration Controller**
- **Packet-level events (ex. loss, delay)**
- **Real-World Networks**
- **WebRTC**
- **Receiver**
Accurate bandwidth estimation is the key factor in achieving high QoE.
Congestion Control (CC) for RTC

- Rule-based CC: ex. GCC
- Hardwired mappings between predefined conditions and actions

![Diagram showing packet-level events, state, conditions, action, and target bitrate.](image-url)
Congestion Control (CC) for RTC

- Rule-based CC: ex. GCC
- **Limitation** of the hardwired mapping-based congestion control
  - Cannot adapt to patterns that aren’t predefined
  - Severe performance degradation when underlying assumptions are violated

Packet-level events (ex. loss, delay) → **Congestion Controller**

- if loss > 10%
- if one-way delay interval > threshold for 100ms,
  ...

**State**

Target Bitrate (= Estimated Bandwidth)

**Action**

**Conditions**
Congestion Control (CC) for RTC

- RL-based CC aims to overcome the limitations of rule-based CC
- Learns to adapt actions based on observed network performance statistics

Packet-level stat. history (ex. loss, delay) → NN or ML model → Target Bitrate (= Estimated Bandwidth)
Congestion Control (CC) for RTC

- RL-based CC aims to overcome the limitations of rule-based CC
- **Benefits** of learning to adapt congestion control decisions
  - Flexible: can adapt to patterns that aren’t predefined

![Diagram of Congestion Controller]

- **Packet-level stat. history** (ex. loss, delay)
- **State**
- **Policy**
  - NN or ML model
- **Action**
  - Target Bitrate (= Estimated Bandwidth)
## Status Quo: Training/Evaluation

- No common, open platform for training & evaluation
  - Trained with in-house simulators or live traffic

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Approach</th>
<th>Target Scenario</th>
<th>Online vs. Offline Training</th>
<th>Training/ Evaluation Data</th>
<th>Validation Environment</th>
<th>Open Sourced</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoRR’19 R3Net</td>
<td>RL-Based</td>
<td>Audio/video call</td>
<td>Offline</td>
<td>Synthetic traces</td>
<td>Custom testbed</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’19 Concerto</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Offline, Simulator</td>
<td>Closed traces</td>
<td>Production traffic</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’20 OnRL</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Online</td>
<td>Production traffic</td>
<td>Production traffic</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’21 Loki</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Offline + Online</td>
<td>Production traffic</td>
<td>Production traffic</td>
<td>X</td>
</tr>
</tbody>
</table>
Status Quo: Validation

- No common, open testbed for fair comparison
  - Validation in closed testbeds or in the wild

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Approach</th>
<th>Target Scenario</th>
<th>Online vs. Offline Training</th>
<th>Training/Evaluation Data</th>
<th>Validation Environment</th>
<th>Open Sourced</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoRR’19 R3Net</td>
<td>RL-Based</td>
<td>Audio/video call</td>
<td>Offline</td>
<td>Synthetic traces</td>
<td>Custom testbed</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’19 Concerto</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Offline, Simulator</td>
<td>Closed traces</td>
<td>Production traffic</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’20 OnRL</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Online</td>
<td>Production traffic</td>
<td>Production traffic</td>
<td>X</td>
</tr>
<tr>
<td>MobiCom’21 Loki</td>
<td>RL-Based</td>
<td>Live commerce</td>
<td>Offline + Online</td>
<td>Production traffic</td>
<td>Production traffic</td>
<td>X</td>
</tr>
</tbody>
</table>

Status: No common, open testbed for fair comparison.
- Validation in closed testbeds or in the wild.
OpenNetLab

- What is OpenNetLab?
  - Next-generation data-driven open networking research platform
  - Community effort including 10+ universities in Asia
  - More information on APNet’21 Keynote talk

- This talk covers one of projects on OpenNetLab for RL-based CC for RTC
OpenNetLab for RL-based CC for RTC

- An end-to-end open platform for RL-based CC for RTC as a community resource for researchers

- Fast training & Reproducible evaluation
- Validation under real Internet
OpenNetLab for RL-based CC for RTC

- Easy to use interface
- High fidelity: WebRTC + ns-3
- Fast & reproducible

- Diverse network types
  - Cellular, wired, wireless
- Geo-distributed nodes across 9 universities in Asia

Simulator

Fast training & Reproducible evaluation

Testbed

Validation under real Internet
Fast Training & Reproducible Evaluation with Simulator

- Python-based easy-to-use interface
- Can port tens of widely used RL libraries
  - ex. OpenAI gym or Ray RLlib
Fast Training & Reproducible Evaluation with Simulator

- Fast & reproducible simulation with ns-3
- High-fidelity simulation via real WebRTC sender/receiver
Fast Training & Reproducible Evaluation with Simulator

- Learn to adapt actions based on observed performance
Validation & Fair Comparison with Real Internet Testbed

- Diverse network types: Cellular, wired, wireless
- Geo-distributed nodes across 9 universities in Asia
Validation & Fair Comparison with Real Internet Testbed
Validation & Fair Comparison with Real Internet Testbed

Researchers

Job submission

Web Frontend

DeployManager

Trained RL policy
SW packages

Job Info Table

Job Status Table

ResourceManager

Request/return VMs

Allocate/release VMs

PM Table

VM Table

Testbed Nodes
Validation & Fair Comparison with Real Internet Testbed

1. Researchers
   - Job submission
   - Web Frontend
   - Trained RL policy
   - SW packages

2. DeployManager
   - Job Info Table
   - Job Status Table
   - Schedule jobs
   - Job running

3. ResourceManager
   - Request/return VMs
   - Allocate/release VMs

4. Testbed Nodes

   - Job finished
Validation & Fair Comparison with Real Internet Testbed

Researchers → Trained RL policy SW packages

Job submission → Web Frontend

DeployManager

- Job Info Table
- Job Status Table

ResourceManager

- Request/return VMs
- Allocate/release VMs

Periodic health check → ACK

Testbed Nodes

Map of East Asia showing countries such as China, South Korea, Japan, and others.
OpenNetLab: Contribution to Research Community

- Three network scenarios with different bandwidth and latency characteristics
- Scoring: weighted sum of network score & media quality score
  - Network score: weighted sum of delay, loss, receiver-side throughput
  - Media quality score: video score (VMAF), audio score (DNSMOS)

### Grand Challenge on

**Bandwidth Estimation for Real-Time Communications**

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Bandwidth range</th>
<th>Mean RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beijing (Wi-Fi with weak signal)</td>
<td>Hong Kong (Wired)</td>
<td>&lt;1Mbps</td>
<td>55ms</td>
</tr>
<tr>
<td>Beijing (Mobile)</td>
<td>Hong Kong (Wired)</td>
<td>2 - 3Mbps</td>
<td>62ms</td>
</tr>
<tr>
<td>Lanzhou (Wired)</td>
<td>Hong Kong (Wired)</td>
<td>&gt;10Mbps</td>
<td>30ms</td>
</tr>
</tbody>
</table>
OpenNetLab: Contribution to the Research Community

- OpenNetLab contributed to training & validation of state-of-the-art RL-based CC
  - Baseline: widely used rule-based GCC
  - HRCC (published in MMSys’21), Gemini (challenge winner)

RL-based CC outperforms GCC on video quality under low bandwidth network path
OpenNetLab: Contribution to the Research Community

- OpenNetLab contributed to training & validation of state-of-the-art RL-based CC
  - Baseline: widely used rule-based GCC
  - HRCC (published in MMSys’21), Gemini (challenge winner)

RL-based CC outperforms GCC on network performance under low & mid bandwidth network path
OpenNetLab: Contribution to the Research Community

- OpenNetLab contributed to training & validation of state-of-the-art RL-based CC
  - Baseline: widely used rule-based GCC
  - HRCC (published in MMSys’21), Gemini (challenge winner)
Future Work

- **Open dataset and model zoo**
  - Trace collection under heterogeneous network types of real Internet

- **Common reference set of 7 state-of-the-art CC algorithms**
  - Rule-based: MMSys’16 GCC
  - CC-codec co-opt: NSDI’18 Salsify
  - Online learning-based: NSDI’18 PCC-Vivace, ICML’19 PCC-RL
  - RL-based: MobiCom’19 Concerto, MobiCom’20 Loki, MMSys’21 HRCC
Conclusion

- **OpenNetLab**: an end-to-end platform for RL-based CC for RTC.
  - Fast training
  - Reproducible evaluation
  - Validation under real Internet

- Try OpenNetLab RTC tools: [https://github.com/OpenNetLab/](https://github.com/OpenNetLab/)

- More information on [https://opennetlab.org/](https://opennetlab.org/)
Q & A