A possible attempt to use sconeupro for RTC scenario

IETF 120
20–21 July 2024
Vancouver, Canada
Latency of Real-time Communication

- Long Latency
- User End
- Access Point
- Sender
- Bandwidth Jitter

Bandwidth detection is too slow. A feedback pathway is needed. Reduce detection time and latency.
sconepro

- sconepro: Secure Communication of Network Properties
  - SCONEPRO is addressing a use-case wherein the network element does “intentional throttling”.

- Our Attempt
  - Identify feedback information
  - Flow Control Strategy of sender
  - Latency reduction effect evaluation
Sconepro-RTC Hackathon Plan

• Plan
  – Testing the latency of sconepro in RTC applications
• Software
  – XQUIC: a commonly used QUIC protocol stack
  – Docker and Linux tc: simulated network environment
• Test Repository
  – https://github.com/zhangjiaxingict/sconepro-rtc
  – https://github.com/zhangjiaxingict/sconepro-xquic
SconePro-RTC Implementation

Receiver

Docker-bridge

TC0

Fixed Latency

Dynamically Changing Bandwidth

Container1: client

193.167.1.100

193.167.1.2

Container2: sim

sconePro

Container3: server

Sender

Docker-bridge

193.167.10.2

193.167.10.100

RTC

XQUIC
Feedback and Flow Control

• Feedback
  – Capacity: bandwidth capacity of AP
  – QLen: queue length of AP

• Flow Control

\[
\text{SndRate} = \text{Capacity} \times 0.9 - \frac{Q\text{Len}}{T_e} \\
\text{BitRate} = \text{SndRate} - \frac{\text{Queue}(\text{sender})}{T_e}
\]

\(T_e\): empty time
Results

Latency-aware Flow Control (COPA)

<table>
<thead>
<tr>
<th>Frame Latency</th>
<th>mean/ms</th>
<th>P90/ms</th>
<th>P99/ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPA</td>
<td>131.6</td>
<td>402.0</td>
<td>815.0</td>
</tr>
<tr>
<td>RTC with sconeipro</td>
<td>72.4</td>
<td>120.5</td>
<td>248.0</td>
</tr>
</tbody>
</table>
Team members

- Jiaxing Zhang email: zhangjiaxing20g@ict.ac.cn
- Qinghua Wu email: wuqinghua@ict.ac.cn
- Zhenyu Li email: zyli@ict.ac.cn
Thank You