Physical-Layer Informed Multipath Redundancy Optimization for Mobile Real-Time Communication

Jing Chen, Zili Meng, Mingwei Xu (INSC, Tsinghua University)

**Background**

- Popular RTC applications pose strict requirements on end-to-end latency
  - E.g., cloud video gaming, video conferencing, remote surgery ...
- High variations of mobile “last mile” greatly impact the path condition
- A common solution: Send data redundantly on multiple paths
  - E.g., when the condition of a path worsens, use congestion control, AQM ...
- Duplicate data on another path (in good condition)

**Question:** How to adapt multipath redundancy rates to path condition?

**Design Challenges**

**C#1:** Which PHY-layer indicator shall we use?

**S#1:** SINR (signal to interference noise ratio)
  - directly related to path condition, accessible

**C#2:** How to use SINR?

**S#2:** A latency Profiler

-5dB (SINR) \(\rightarrow\) Profiler \(\rightarrow\) Latency distribution

**C#3:** How to seek a balance between low tail latency and high goodput?

**S#3:** A Scheduler to optimize multipath redundancy rates

- A Mealy FSM
- Probability modelling

**Motivation**

**Existing solutions:**

<table>
<thead>
<tr>
<th>Redundancy</th>
<th>Aggressive</th>
<th>Conservative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path condition profiling</td>
<td>Oblivous</td>
<td>Base on RTT measurement</td>
</tr>
<tr>
<td>Example</td>
<td>ReMP TCP</td>
<td>LowRTT</td>
</tr>
</tbody>
</table>

**Performance**

- Low goodput
- High tail latency

Our design goal: to strike a balance between tail latency & goodput

**Key Factor:** timely and accurately observe path degradation

However: transport-layer observation of path degradation is delayed
(illustrated in the Background figure)

**Our Contribution: PhyRO**

- Use **physical-layer indicators** to decide multipath redundancy rates
- PHY-layer indicators can more timely reflect path degradation

**Preliminary Evaluation**

Trace-based simulation

Wireless link propagation: ITU-R 1411 NLoS model

UE mobility: random walking model

Effectively reduces the stuttering rate

Negligible goodput degradation

**On-going Work**

- Verify Profiler’s accuracy
- Measure the algorithm overhead
- Experiment in NS-3 & testbed
- Implement into protocol stack