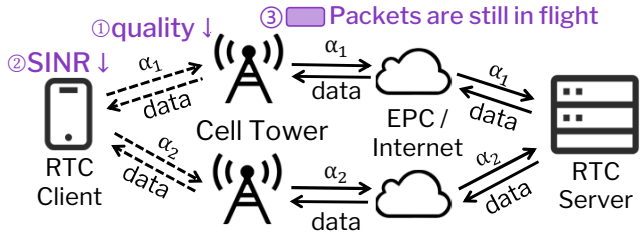
APNet 2021 – 5th Asia-Pacific Workshop on Networking. Jun 24, 2021, Shenzhen, China.

# 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,
  - Congestion control, AQM ... ×
  - Duplicate data on another path (in good condition) √




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

## Motivation

Existing solutions:

| **Redundancy** | Aggressive | Conservative |
|---|---|---|
| **Path condition profiling** | Oblivious | Base on RTT measurement |
| **Example** | ReMP TCP | LowRTT |
| **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

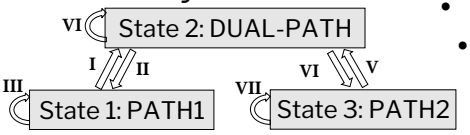
- Use **physical-layer indicators** to decide multipath redundancy rates
  - PHY-layer indicators can more timely reflect path degradation

## Design Challenges

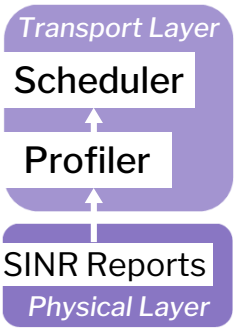
**C#1**: Which PHY-layer indicator shall we use?
**S#1**: SINR (*signal to interference noise ratio*)
- directly related to path condition, accessible

**C#2**: How to use SINR?
**S#2**: A latency *Profiler*


**C#3**: How to seek a balance between low tail latency and high goodput?
**S#3**: A Scheduler to optimize multipath redundancy rates



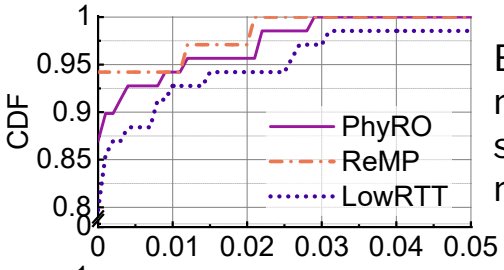

- A Mealy FSM
- Probability modelling



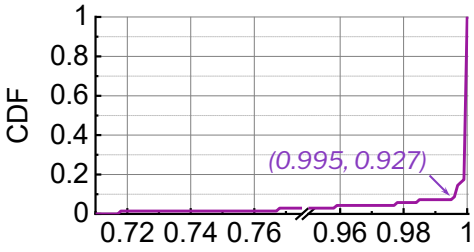

**PhyRO Overview** (intended for the client side)

## 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 *Welcome Advice!*

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