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

Jing Chen[1,3], Zili Meng[1,3], Mingwei Xu[1,2,3]

[1]Institute for Network Sciences and Cyberspace, [2]Department of Computer Science and Technology, Tsinghua University
[3]Beijing National Research Center for Information Science and Technology (BNRist)
j-chen16@tsinghua.org.cn,zilim@ieee.org,xumw@tsinghua.edu.cn

CCS CONCEPTS

• **Networks** → **Mobile networks**; *Cross-layer protocols.*

ACM Reference Format:
Jing Chen, Zili Meng, Mingwei Xu. 2021. Physical-Layer Informed Multipath Redundancy Optimization for Mobile Real-Time Communication. In *5th Asia-Pacific Workshop on Networking (APNet 2021) (APNet '21), June 24–25, 2021, Shenzhen, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3469393.3469673

1 INTRODUCTION

Mobile real-time communication (RTC) applications (e.g., video conferencing, cloud gaming) are growing popular, but pose strict (i.e., millisecond-level) requirements on the end-to-end user-perceived latency. Various mechanisms at the transport layer [1] are proposed to reduce the end-to-end latency by carefully designing mechanisms against congestion and packet losses. Specially, for mobile scenarios where the *capacity* of the last-mile link frequently experiences high variations [5], merely relying on transport layer adaption cannot achieve satisfactory quality of experience. Therefore, a common way to further reduce the tail latency is to duplicatively send data over several independent cellular connections on user devices [2, 3], which offers a much greater chance to deliver the traffic in time at the application layer.

To accurately mask the tail latencies with minimum additional bandwidth, it is crucial to decide the proper multipath redundancy rate. However, existing multipath redundancy solutions fail to strike a good balance between latency and goodput for mobile RTC. On one hand, aggressively duplicating traffic on multipaths oblivious to the path conditions (e.g., ReMP TCP [2]) could compellingly reduce the tail latency, but would introduce unnecessary redundant transmission and consume more bandwidth. On the other hand, optimizations towards high goodput could save bandwidth and increase the goodput yet have little improvement in terms of latency.

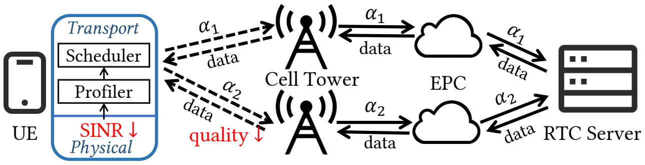


Figure 1: PhyRO Overview. The UE calculates redundancy rates α_1,α_2 based on PHY-layer indicators and requests the server to schedule redundant data accordingly.

The failure to balance between latency and goodput roots in the delayed observation of path degradation at the transport layer. When reactively duplicating traffic after observing the degradation of one path, the packet delivery time to the RTC application would be at least delayed for one RTT of the slow path. Therefore, our key observation is that physical-layer indicators (e.g., signal strength) can more timely reflect the degradation of cellular channels with 0.5 RTT in advance of transport-layer indicators. As illustrated in Fig. 1, when the channel quality of the cellular last mile worsens, the user equipment (UE) almost immediately observes a drop in the signal strength. However, the transport layer can only observe the degradation by receiving a successfully delivered packet from the cell tower and update the RTT information. Considering the last mile is becoming the major bottleneck of RTC services, such a time (the packet delivery time from the cell tower to the UE) could be as high as almost 0.5 RTT, which severely increases the delay at the application layer.

In response, we propose PhyRO, a multipath redundancy optimizer that takes PHY-layer information as the criterion. The design challenges are (1) which PHY-layer indicator shall we use, (2) how to use it, and (3) how to balance between latency and goodput in choosing the path and the corresponding redundancy rates. In response, we choose the signal strength as the indicator based on its close relation to the channel quality. PhyRO uses a `Profiler` to predict the probability distribution of the transport-layer latency based on real-time signal strengths. Then, PhyRO introduces a `Scheduler` to optimize the redundancy rate by mathematically seeking a balance between tail latency and goodput

Supported by National Key R&D Program of China (2018YFB1800302) and NSFC (61625203 and 92038302). Mingwei Xu is the corresponding author.

APNet '21, June 24–25, 2021, Shenzhen, China
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8587-9/21/06…$15.00
https://doi.org/10.1145/3469393.3469673

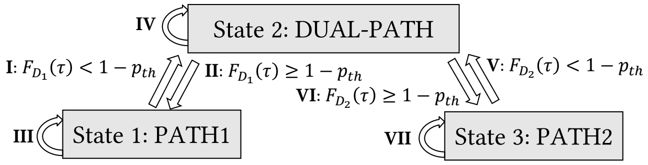

State 2: DUAL-PATH
State 1: PATH1
State 3: PATH2
IV
III
VII
I: $F_{D_1}(\tau) < 1 - p_{th}$
II: $F_{D_1}(\tau) \geq 1 - p_{th}$
VI: $F_{D_2}(\tau) \geq 1 - p_{th}$
V: $F_{D_2}(\tau) < 1 - p_{th}$

Figure 2: The Scheduling FSM

with a finite state machine. Evaluation with NS-3 shows that PhyRO improves the 95th percentile stuttering rate by 54.8% against LowRTT scheduler, with negligible goodput loss.

2 DESIGN

Multiple PHY-layer indicators are used in cross-layer networking designs, including the downlink physical resource block (PRB) allocation and signal strength related indicators (e.g., RSSI, RSRP, SINR) [5]. Since the available bandwidth reflects the combined influence of the path condition and competing traffic, we focus on the signal strength, specifically, the *signal to interference noise ratio* (SINR), which directly reflects the path condition and is accessible to the upper layers via multiple tools (e.g., MobileInsight [4]).

To tradeoff between the bandwidth consumption and latency and choose proper redundancy rates, we design 2 components (shown in Fig. 1): the `Profiler` first uses the measured signal strength to profile the delay distribution, and then the `Scheduler` optimizes the redundancy decisions with a finite state machine (FSM) and probability modelling.

We first introduce some denotations. Let D represent the one-way delay (OWD) and s represent the SINR value. We denote the maximum tolerable user-perceived one-way delay as τ and the maximum tolerable stuttering rate as p_{th}. The decision is expressed as α_1 and α_2, which are the redundancy rates for data to be transferred on path1 and path2. We denote the Cumulative Distribution Functions (CDF) of the OWD on the two paths as $F_{D_1}(\cdot)$ and $F_{D_2}(\cdot)$.

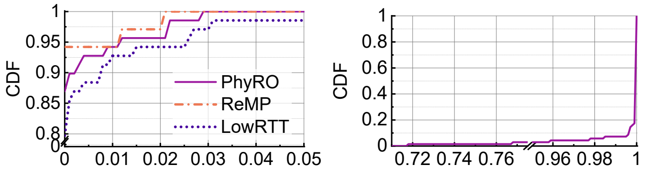
The Profiler aims at estimating the probability distribution of OWD given a SINR measurement, i.e, $F_D(\cdot|s)$. As a preliminary attempt, we collect SINR and OWD traces and fit the OWD values into a Gaussion distribution given a SINR.

The Scheduler maintains a FSM with 3 states: concurrently using the 2 paths (`DUAL-PATH`), only using the first path (`PATH1`) and only using the second path (`PATH2`), as shown in Fig. 2. When the `Profiler` updates the OWD distributions based on a new SINR measurement, the `Scheduler` checks the triggering conditions (marked in Fig. 2), goes to the next state and calculates (α_1, α_2) as follows:

For transition II, III, VI and VII where one path is enough to keep the stuttering rate lower than p_{th}, data are scheduled only to that path (e.g., $\alpha_1 = 1, \alpha_2 = 0$ for II).

For the state transition I (and similarly, V) where the previously used path is not enough, we should immediately schedule redundant data on the other path, ensuring that the overdue probability is smaller than p_{th}:

$$[1 - F_{D_1}(\alpha_1\tau)][1 - F_{D_2}(\alpha_2\tau)] \leq p_{th} \qquad (1)$$



(a) Stuttering Rate (b) Goodput Ratio

Figure 3: Simulation results on 69 7-minute traces.

So the redundant scheduling decision should be:

$$\alpha_1 = 1, \alpha_2 = F_{D_2}^{-1}[1 - p_{th}/(1 - F_{D_1}(\tau))]/\tau \qquad (2)$$

For the state transition IV, we adaptively renew the redundancy rates on dual paths by minimizing the bandwidth consumption $(\alpha_1 + \alpha_2)$ under the constraint of Eqn. 1. We optimize with the Lagrange Multiplier method, getting:

$$F_{D_1}(\alpha_1\tau) = 1 - p_{th}^{\alpha_1/(\alpha_1+\alpha_2)},\ F_{D_2}(\alpha_2\tau) = 1 - p_{th}^{\alpha_2/(\alpha_1+\alpha_2)} \qquad (3)$$

However, solving α_1, α_2 from Eqn. 3 is nontrivial, since it is a set of transcendental equations with α_1 and α_2 coupling together. In response, we adopt the Coordinate Descent method to iterate α_1 and α_2 in turns until their convergence.

3 EVALUATION & FUTURE WORK

Our preliminary simulations are based on NS-3 traces. We simulate the wireless link propagation loss with an ITU-R 1411 NLoS model (for Non-Line-of-Sight short-range outdoor communication in the frequency range 300 MHz to 100 GHz). We simulate UE mobility with a random walking model. For comparison, we use the a `ReMP` scheduler (which duplicates all traffic on all available paths) and a `LowRTT` scheduler (which selects the path with the lowest RTT) as baselines. In the experiment, we set τ = 100ms and p_{th} = 0.0001.

We present the CDF of the stuttering rate (defined as the proportion of packets that miss the latency requirement) on 69 traces in Fig. 3(a). PhyRO reduces the 95th-percentile stuttering rate by 54.8% against the `LowRTT` scheduler, and achieves relatively close performance to the optimal stuttering rate of `ReMP` scheduler. We also measure the goodput ratio (goodput / total throughput) , and the goodput ratios of `ReMP` and `LowRTT` are constantly 0.5 and 1 by their design. As shown in Fig. 3(b), PhyRO has a very high goodput ratio (>0.995 by 92.7%), i.e., PhyRO reduces the stuttering rate of `LowRTT` with negligible bandwidth overhead.

In the future, we will take more practical factors (e.g., energy overhead, the impact of various cellular standards) into consideration and conduct testbed experiments.

REFERENCES

[1] Gaetano Carlucci, Luca De Cicco, et al. 2017. Congestion control for web real-time communication. *IEEE/ACM Trans. Netw.* (2017).

[2] Alexander Frommgen, Tobias Erbshäußer, Alejandro Buchmann, et al. 2016. ReMP TCP: Low latency multipath TCP. In *Proc. IEEE ICC*.

[3] HyunJong Lee, Jason Flinn, et al. 2018. RAVEN: Improving interactive latency for the connected car. In *Proc. ACM MobiCom*.

[4] Yuanjie Li et al. 2016. Mobileinsight: Extracting and analyzing cellular network information on smartphones. In *Proc. ACM MobiCom*.

[5] Yaxiong Xie et al. 2020. PBE-CC: Congestion Control via Endpoint-Centric, Physical-Layer Bandwidth Measurements. In *ACM SIGCOMM*.