Characterization of Multi-User Augmented Reality over Cellular Networks

Kittipat Apicharttrisorn, Bharath Balasubramanian, Jiasi Chen,
Rajarajan Sivaraj, Yu Zhou, Rittwik Jana,
Srikanth Krishnamurthy, Tuyen Tran, and Yi-Zhen Tsai

UNIVERSITY OF CALIFORNIA UCRIVERSIDE AT&T Labs Research

Motivation

- AR promises new immersive experiences (e.g., AR glasses)
  - Forecast to reach $100 billion market in 2021
- **Yet we don't understand how AR apps communicate**
- AR differs from other apps (e.g., video streaming, web)
  - No playback buffers
    - Unlike video: allows video chunks to arrive late
  - No application adaptation
    - Unlike video: adaptive bit rate
    - Unlike web: first paint above the fold
  - Uplink-heavy TCP traffic
    - Unlike QUIC in YouTube or UDP in gaming

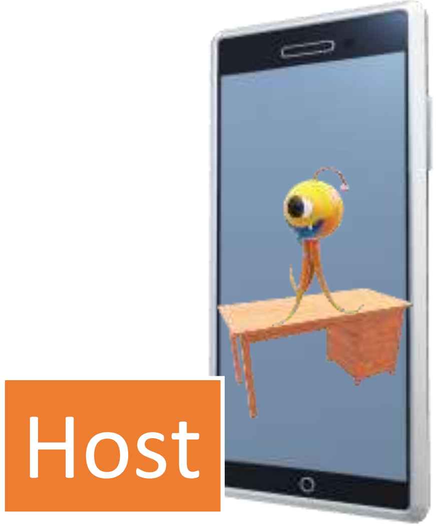
Why AR over Cellular?

- Cellular networks cover 70% of the US*
- Outdoor AR apps (e.g., Pokemon Go) use the cellular network

- **Key question**: How does the cellular network contribute to AR performance?
- **Key finding:** Cellular networks accounts for 30% of end-to-end AR latency
  - We break down the sources of latency and propose client/network solutions

* https://www.whistleout.com/CellPhones/Guides/Coverage

Multi-User Augmented Reality



User A **hosts** an object on the table

User B **resolves** the object in its field-of-view

End-to-end latency = latency from when user A places the virtual object, to when user B sees it on the screen

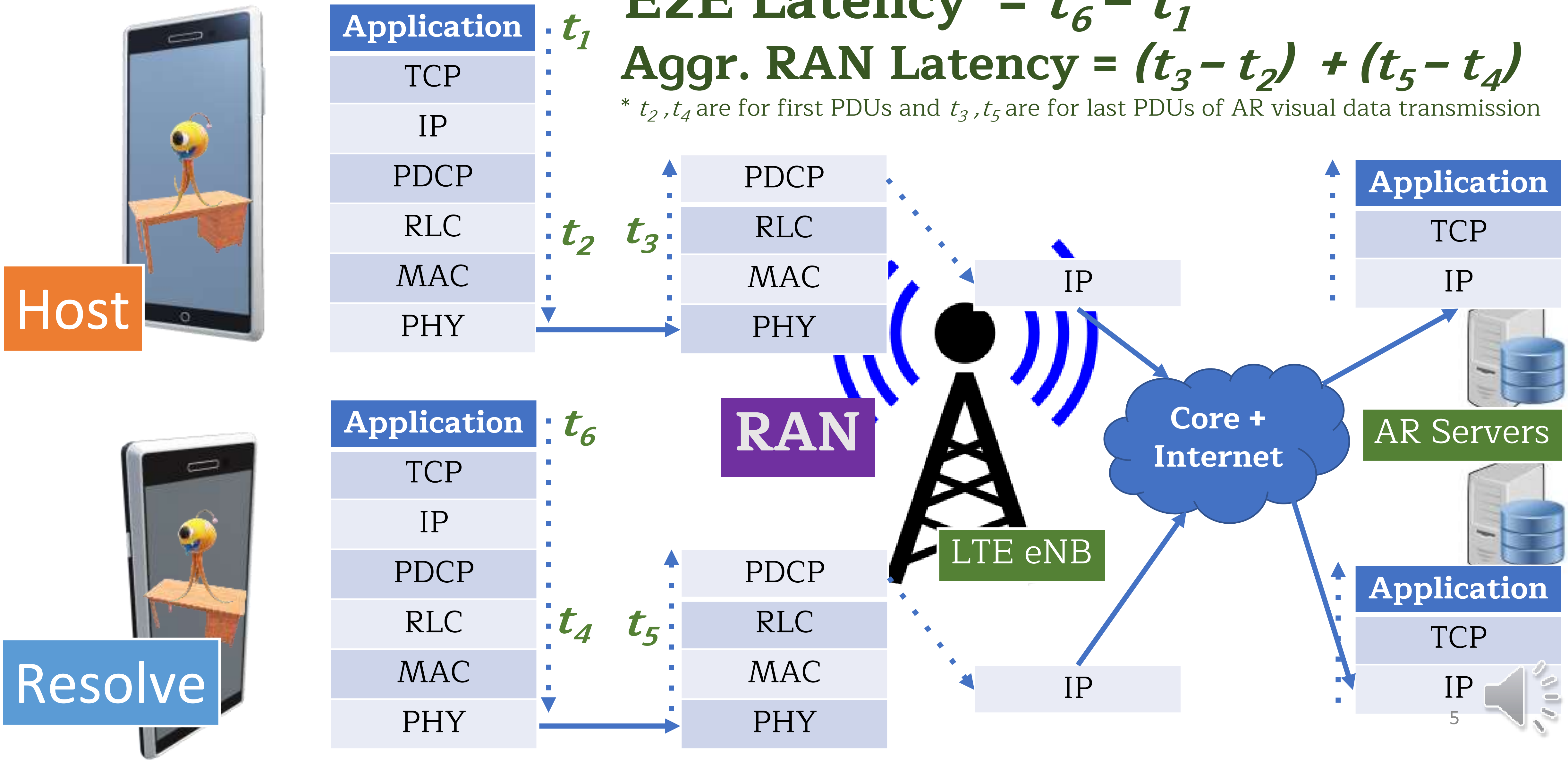
Aggregate RAN latency = the air interface portion of end-to-end latency

AR, Cellular and RAN: A Quick Primer

E2E Latency $= t_6 - t_1$

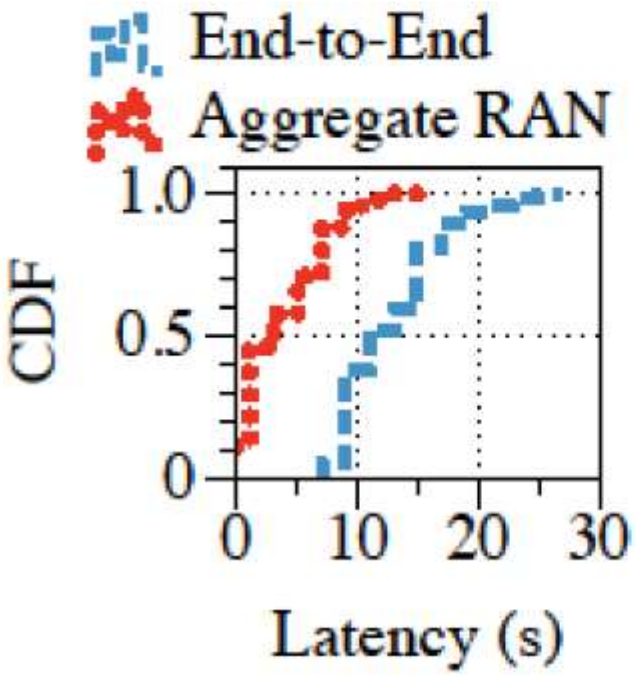
Aggr. RAN Latency $= (t_3 - t_2) + (t_5 - t_4)$

* t_2, t_4 are for first PDUs and t_3, t_5 are for last PDUs of AR visual data transmission



Multi-User AR over Cellular Networks

- **How much end-to-end (E2E) latency is experienced by AR users?**
  - Median of **3.9 s** (air interface) and **12.5 s** (E2E)
  - Far from the dream of seamless AR <= E2E *0.5 s*
  - High E2E latency can cause inconsistent user views
    - E.g., one user sees an object already removed by another

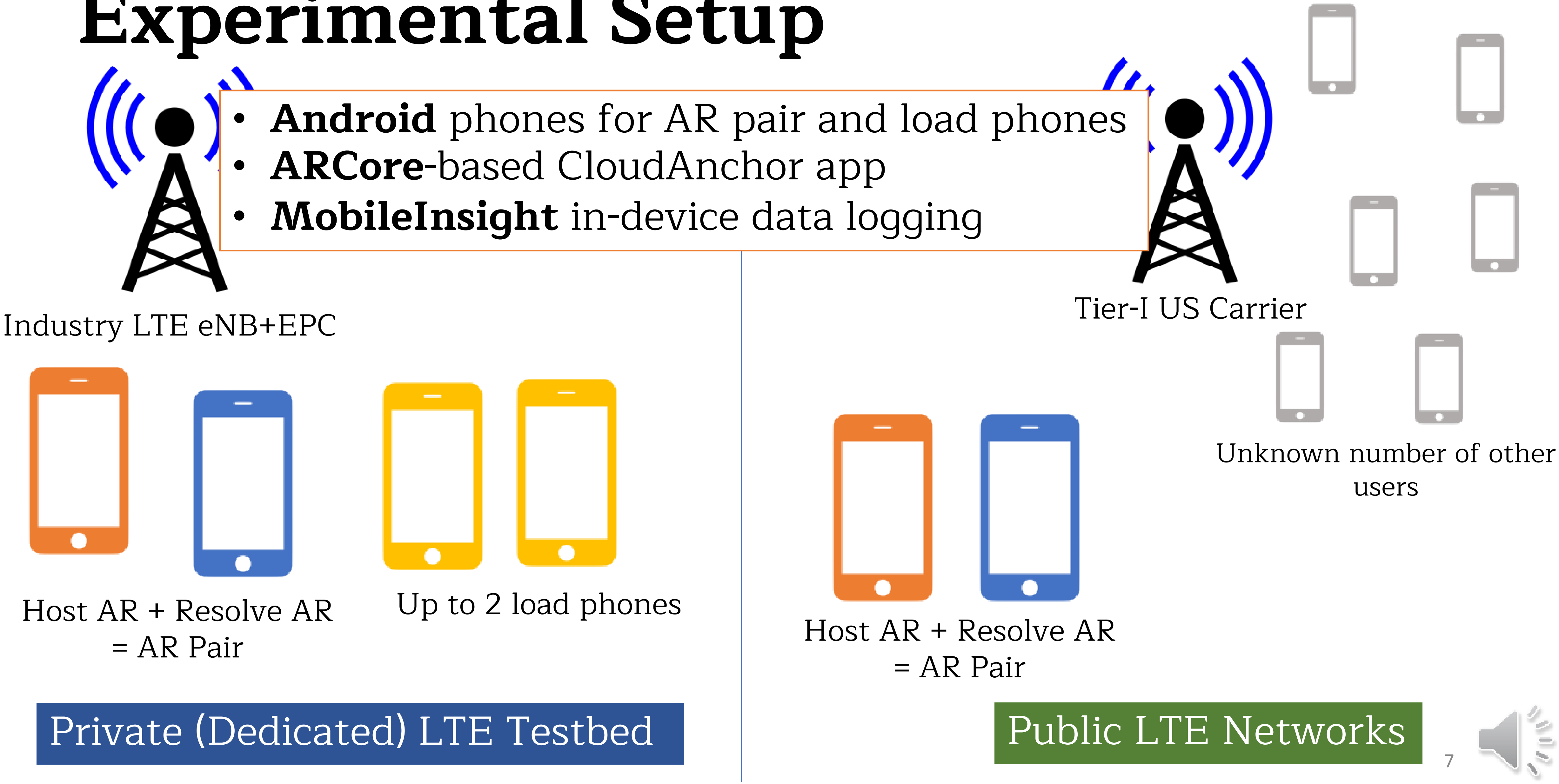


*AR app over a Tier-1 operator in the US, > 50 trials on 5 different locations

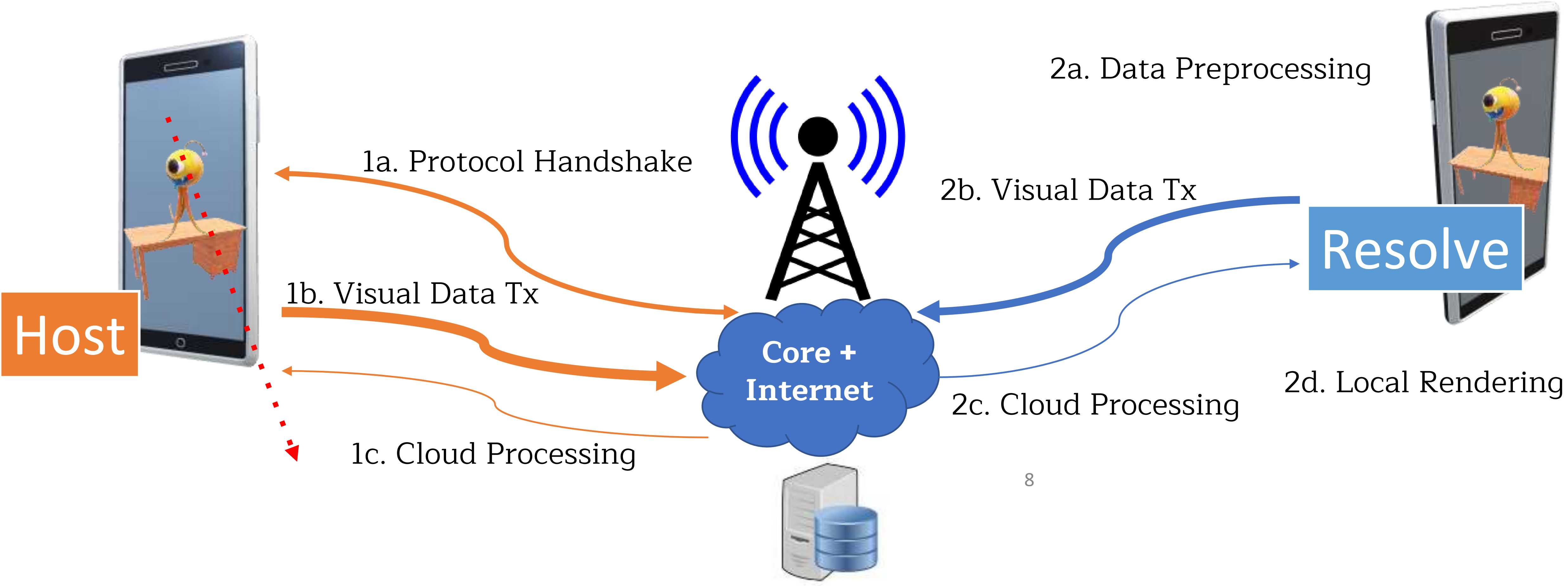
- E2E = latency from host user taps the screen to host a virtual object to resolve user sees it on the screen
- Aggr. RAN = the air interface portion of E2E

Experimental Setup

- **Android** phones for AR pair and load phones
- **ARCore**-based CloudAnchor app
- **MobileInsight** in-device data logging

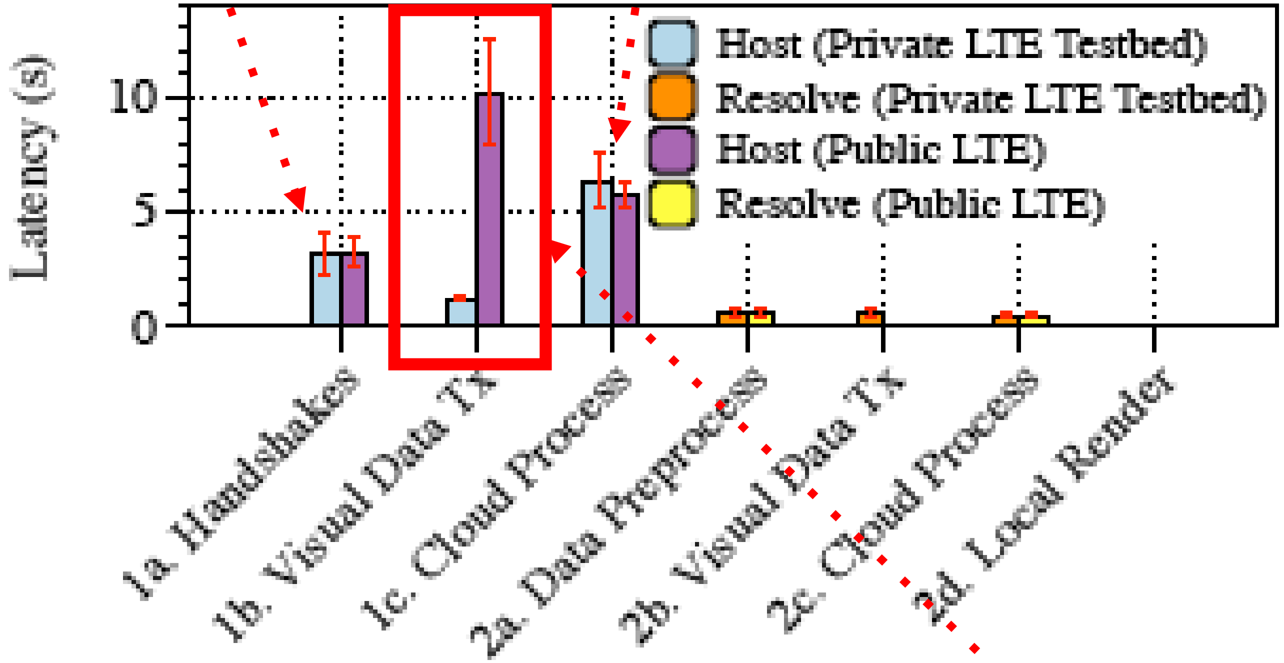


How is the E2D latency broken down?



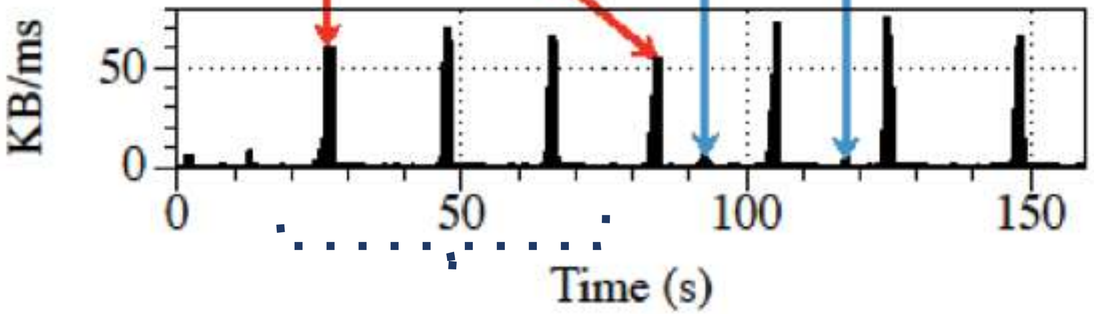
How is the E2D latency broken down?

- **Protocol handshakes**
  - -> reduced by protocol streamlining of AR platforms
- **Cloud processing**
  - -> reduced by more cloud resources or efficient processing

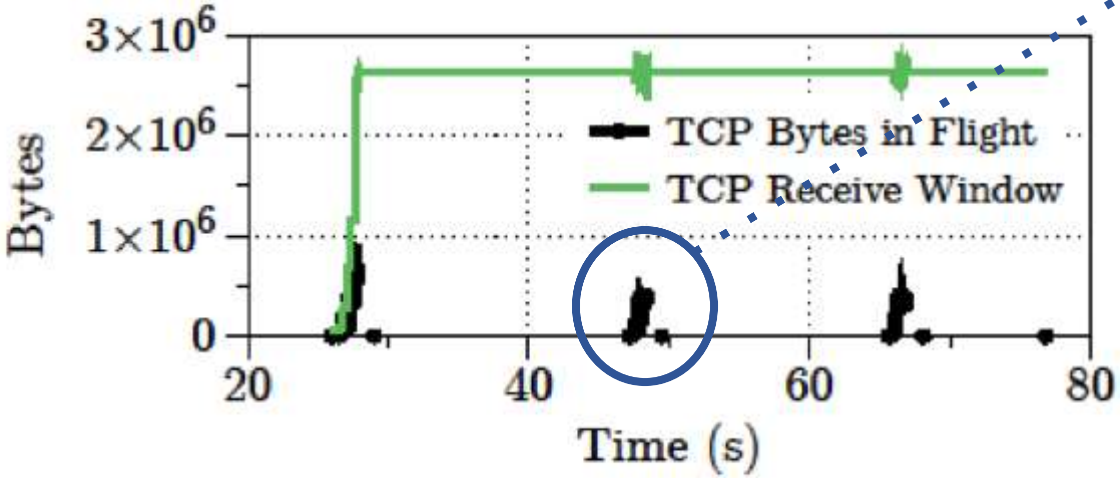


Visual data tx latency is significant (~ 30 % of E2E). -> focus of this paper

What are AR traffic characteristics?

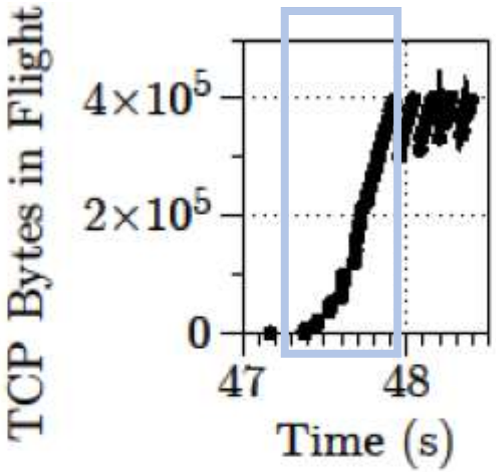


TCP information of the first three spikes



TCP BIF ≅ TCP cwnd

AR traffic enters TCP slow-start every time a user places a new virtual object.

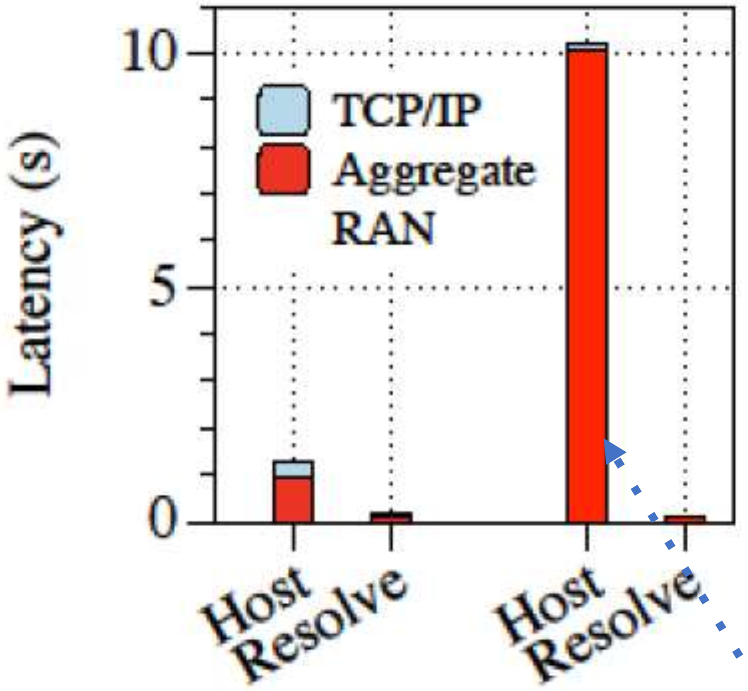


Slow-Start Restart (SSR)

This causes the communication latency to be longer than what the network can offer.

AR traffic is bursty, which negatively impacts TCP performance

How much does the RAN contribute to network latency?



Dedicated LTE Testbed

Public LTE Network

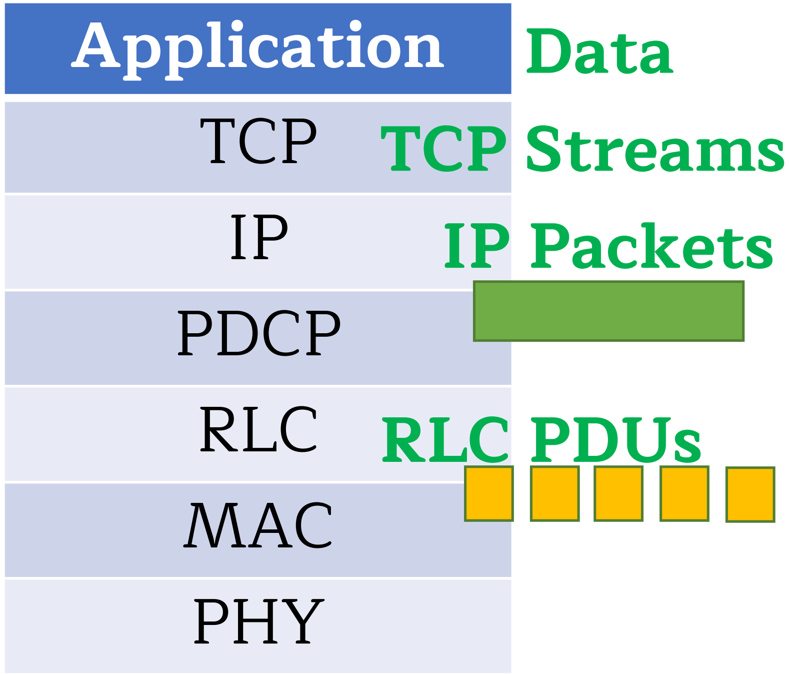
= first visual data packet sent until last ack received

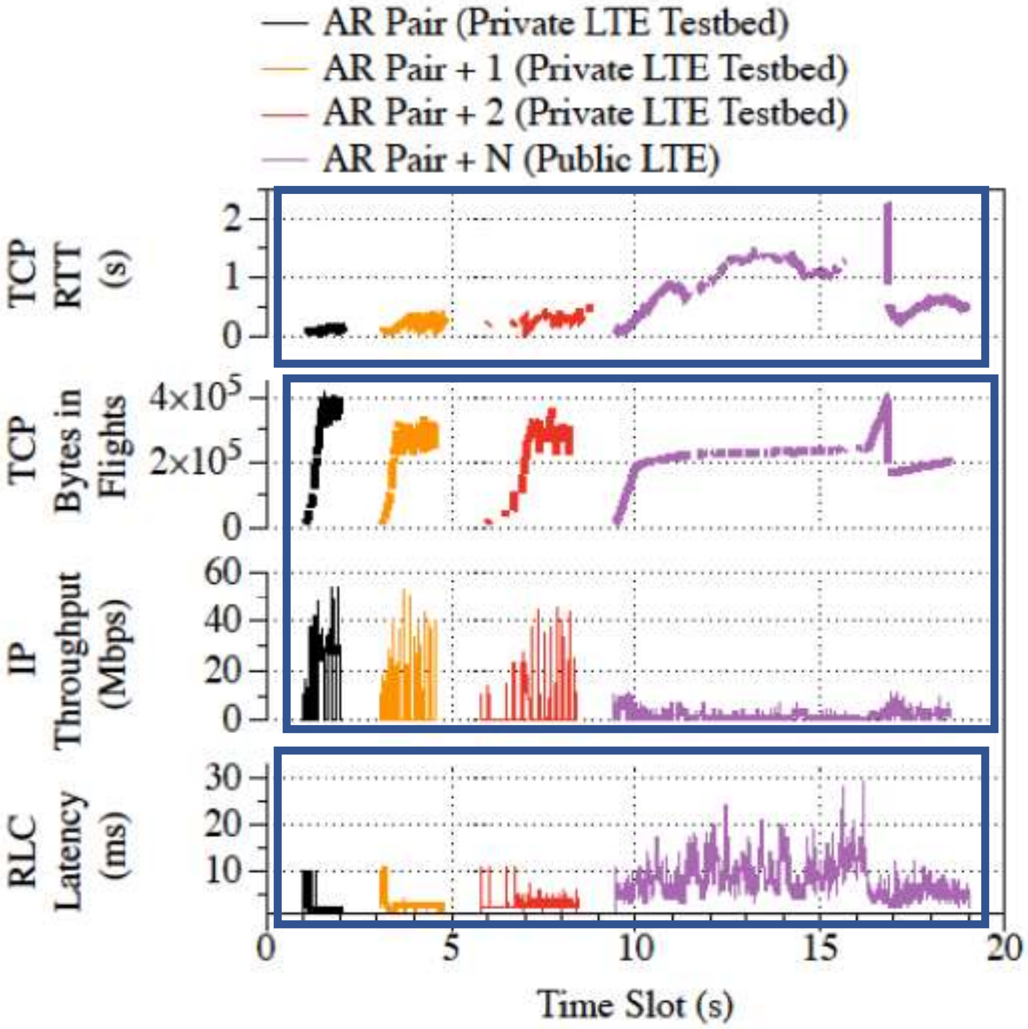
- Below-IP analysis is challenging and new
  - adding hardware-level timestamps on the base station was difficult on our testbed and in production networks
  - logging possible, but no analyzer to extract RAN latency
  - we created a custom analyzer for MobileInsight UE logs
    - https://github.com/patrick-ucr/ran_latency_analyer_mi

Air interface (RAN) latency is a significant portion (71-98%) of the network latency

Even with faster core networks or edge computing, RAN latency is still significant and needs to be reduced.

What causes high RAN latency and how to reduce it?





Higher RLC latency
->Higher TCP RTT

Higher TCP RTT -> longer TCP slow-start / lower IP throughput

More other users in the network
-> Higher RLC latency

Reducing AR IP packet sizes in a highly-congested network may help reduce RAN (E2E) latency.

Proposed Optimizations for AR

- Network aware optimization
  - Smaller IP packet size (1430 -> 650 bytes) reduces 37% RAN latency in high-congestion networks
  - Because it improves IP throughput and application goodput
- Network agnostic optimization
  - When AR device not sending data, base station forces device to return to *an idle state*
  - High overhead of returning from idle → active state
  - AR device sends *periodic small background traffic* to reduce 50% RAN latency
  - Negligible increase in outgoing data

Conclusions

- First in-depth measurement study of multi-user AR apps over cellular networks

- We characterized AR traffic
  - RAN latency is a significant portion (30%) of AR end-to-end latency
  - AR traffic is uplink-heavy and bursty
  - AR has poor interactions with TCP and the cellular network

- We design network-aware and network-agnostic optimizations that can reduce latency ~40-70%

- Future work: Other AR apps, AR over 5G networks

CHARACTERIZATION OF MULTI-USER AUGMENTED REALITY OVER CELLULAR NETWORKS

Thank you! Questions?

UC RIVERSIDE — UNIVERSITY OF CALIFORNIA

AT&T Labs Research