32nd Multi-Service Networks workshop (MSN 2020)

Congestion Control via Endpoint-Driven, Wireless Physical-Layer Capacity Measurements

Kyle Jamieson
with Yaxiong Xie (lead) and Fan Yi

PAWS
Princeton Advanced Wireless Systems

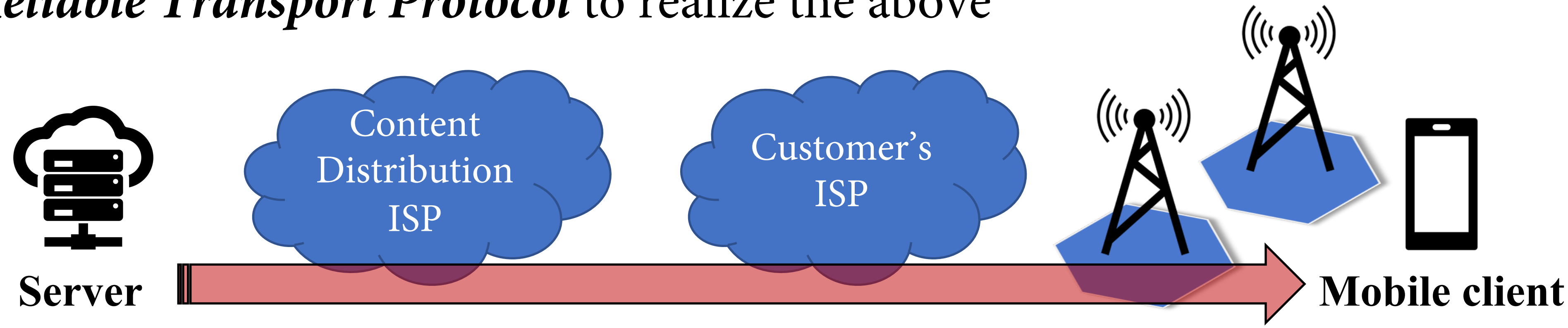
PRINCETON UNIVERSITY

Introduction

- Many downlink data flows terminate at a **wireless** last hop

- Wireless last hop does the most **damage** to a data flow

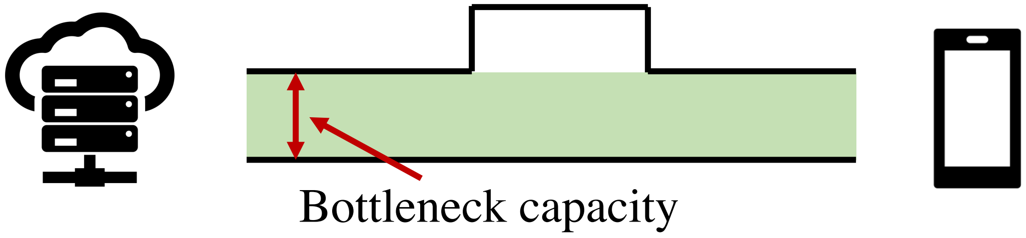
- **Today:** Endpoints are best positioned to measure wireless congestion
  - **Feedback** measurements end-to-end ***via a well-defined API***
  - A ***Reliable Transport Protocol*** to realize the above



Our Design Goal: Exact Congestion Control

- Google showed Kleinrock's optimal operating point is achievable [BBR, 2016]

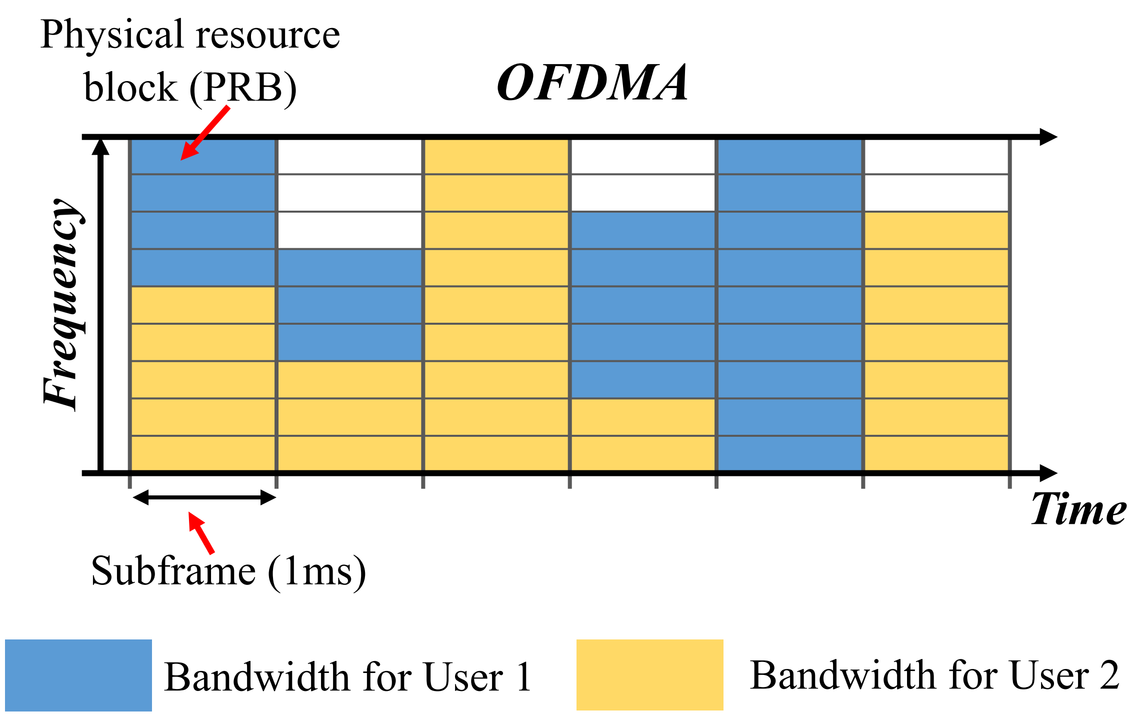
- **Design goal:** **Match** sender's rate to bottleneck's capacity:



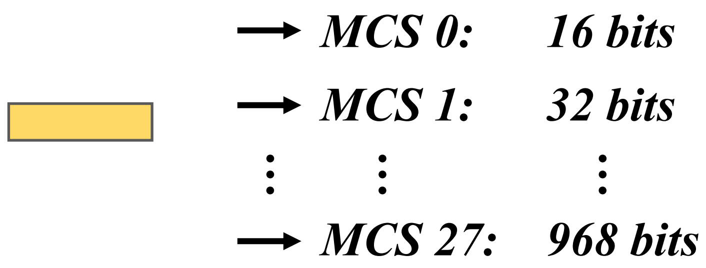
- **Key challenge:** Estimating **bottleneck capacity**
  - BBR and most others use **end-to-end measurements** to estimate capacity
    - Packet transmit and acknowledgement times, packet sizes

Capacity **varies significantly** in cellular networks

- Capacity of cellular wireless link depends on ***allocated bandwidth*** and ***wireless channel quality***



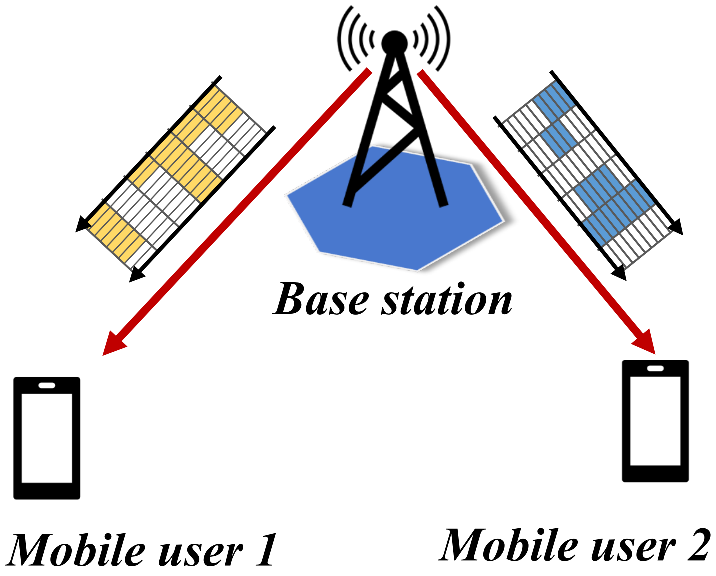
The **number of bits** that one resource block carries is determined by the **channel quality**



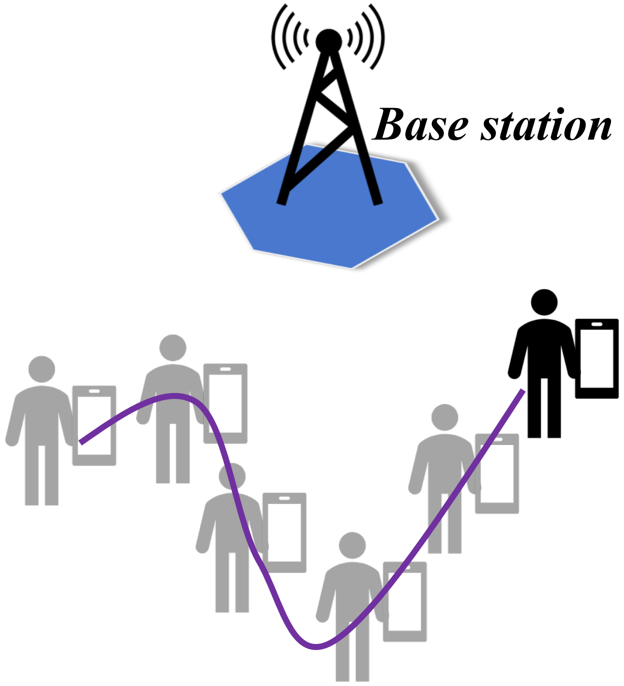
Capacity **varies rapidly** in cellular networks

- Capacity of cellular wireless link depends on ***allocated bandwidth*** and ***wireless channel quality***

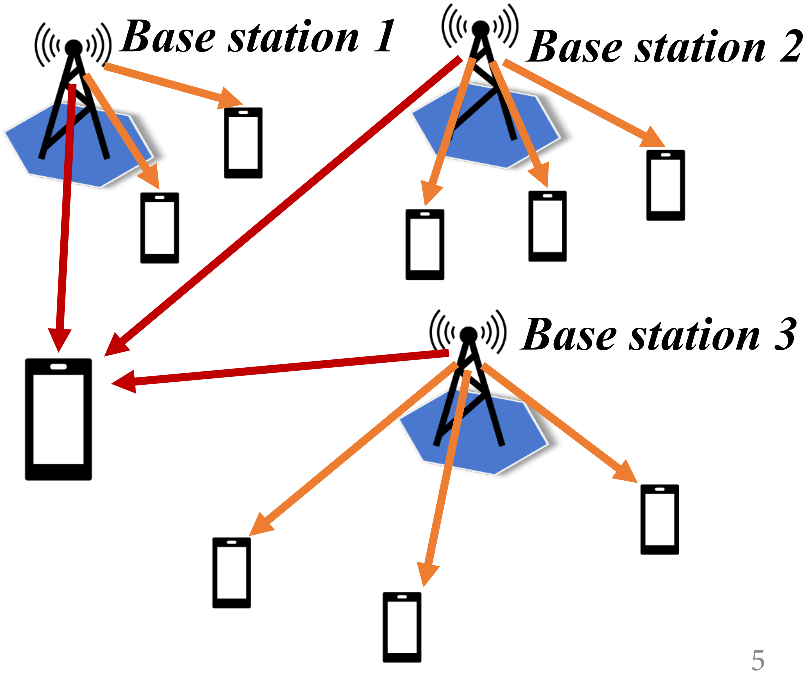
Competition between mobile users



User mobility



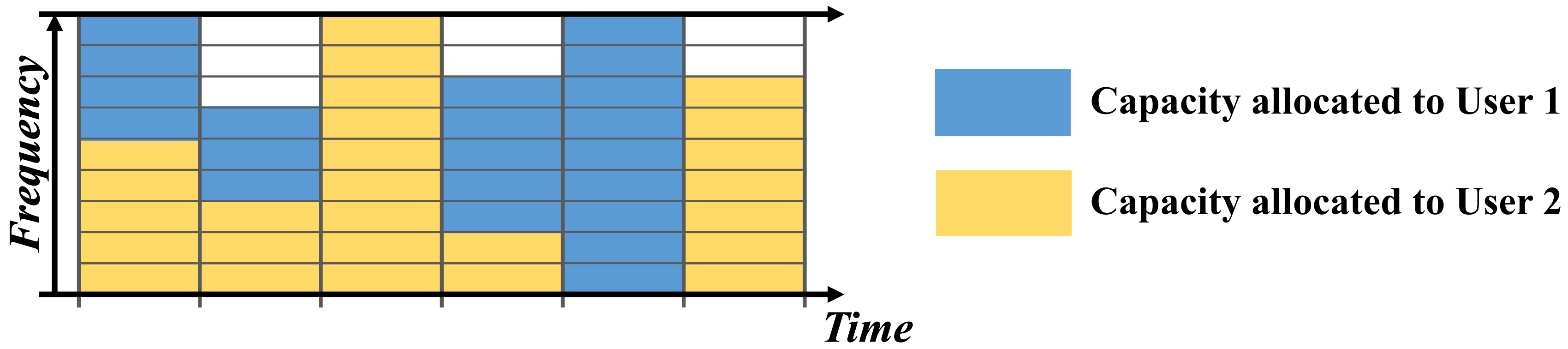
Carrier aggregation



Opportunity: Mobile Monitors

- Resource allocation, carrier aggregation, and PHY configuration (MCS) are **broadcasted** by the cell tower via the PHY control channel

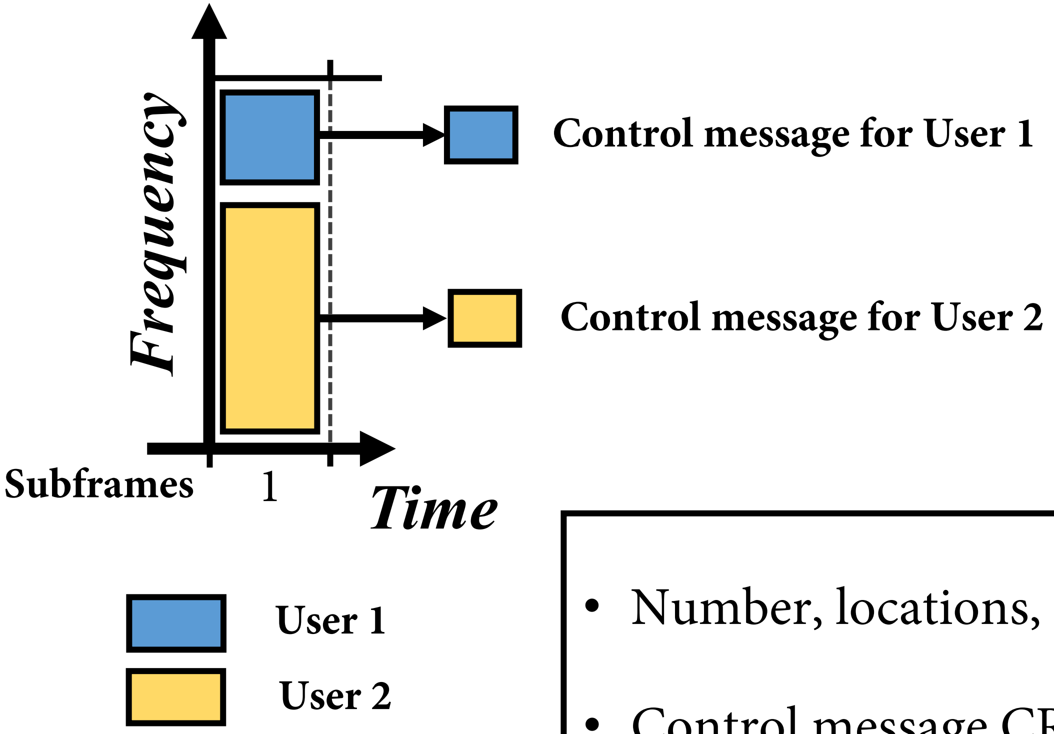
- **The opportunity:** have the **mobile end points snoop on the cell towers**
  - Congestion control endpoints **know detailed state immediately**
  - No infrastructural changes required

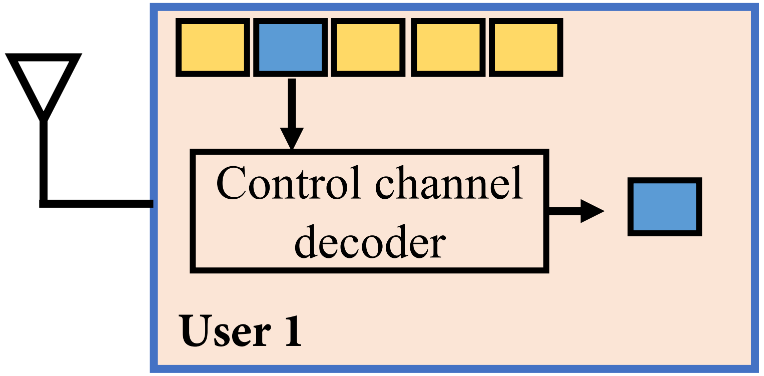


Resource allocation at base station (overheard by User 1)

Control channel decoder

- One mobile user only decodes its own control messages!





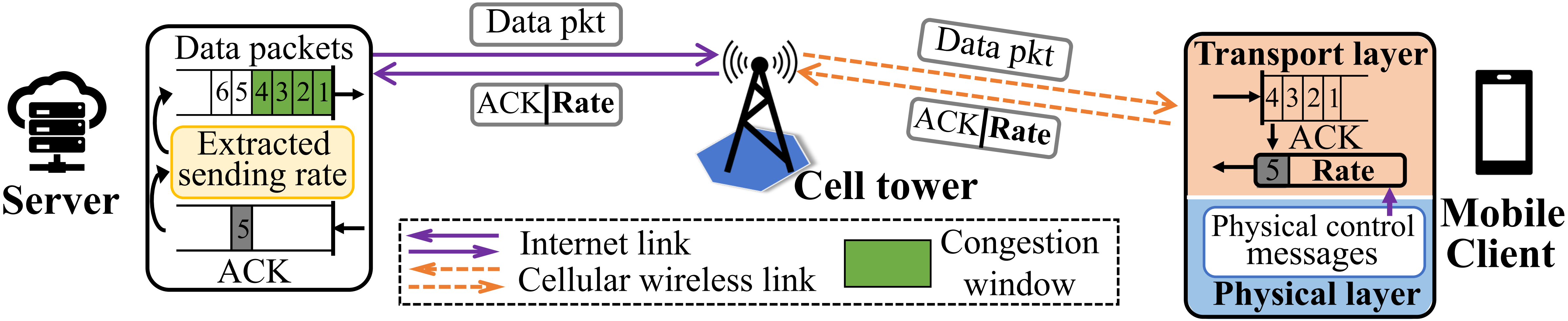
Unknowns

- Number, locations, and formats of control messages in the control channel
- Control message CRCs are XORed with the ID of the user (which is unknown)
- Interference from neighboring cells may result in phantom control messages

Our congestion control design

- **Mobile:** measures available wireless capacity based on decoded control information
  - Sends capacity as feedback in acknowledgements, back to the server

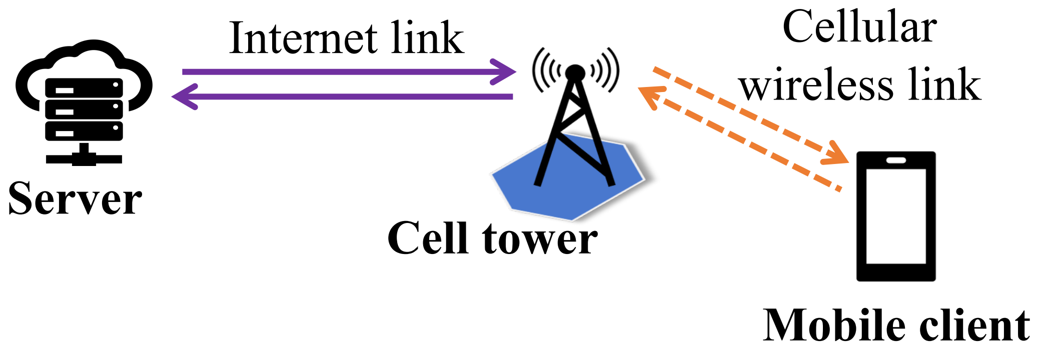
- **Sender: Explicit rate control** (similar in spirit to XCP) based on the mobile's reports



Bottleneck of an end-to-end connection

- **Goal of congestion control:** **match sending rate** to capacity of the bottleneck

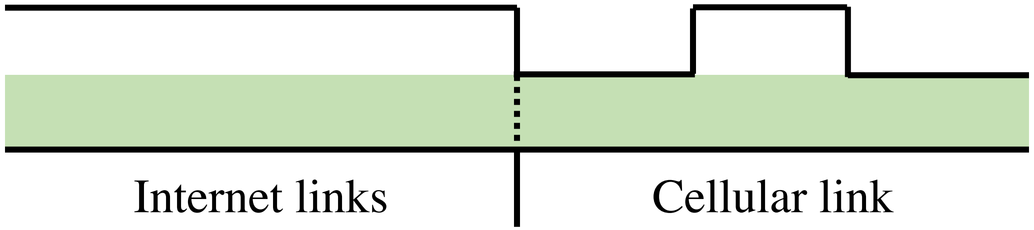
- **Challenge**: Bottleneck location may **alternate** between Internet and wireless link
  - **Can only estimate wireless capacity** with decoded physical control information



Strawman: Just match wireless capacity?

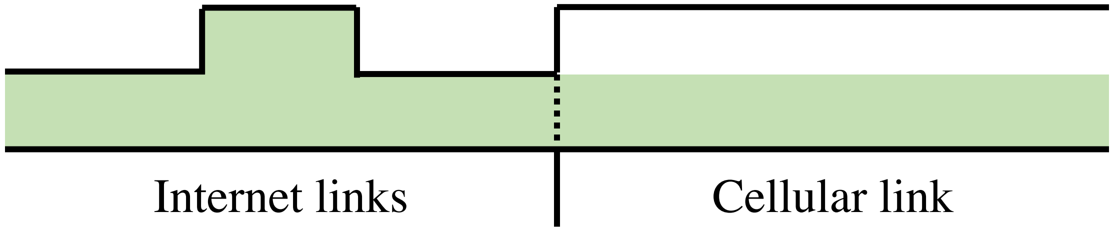
Blindly matching the sending rate to the capacity of the wireless cellular link

Bottleneck at the cellular link



Kleinrock's operating point

Bottleneck at the Internet link



Over-utilization/saturation

- Send rate exceeds Internet link capacity
- Packets queue at Internet bottleneck
  - **Increased delay → packet drops**

Cross-layer bit rate translation

- **Assumption**: the bit error rate (BER) of each data bit inside one TB is p and that bit errors are *i.i.d.*
  - We estimate the BER using signal to interference noise ratio (SINR)

$$C_p = C_t + C_t\left(1 - (1 - p)^{L}\right) + 0.068 \cdot C_p$$

Transport block error rate: $1 - (1 - p)^{L}$

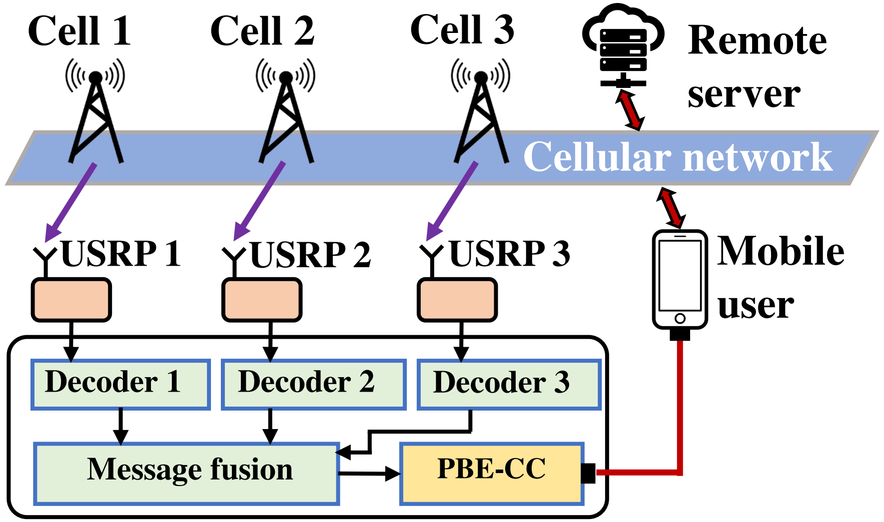
Transport block size $L = C_t \cdot 10^{-3}$

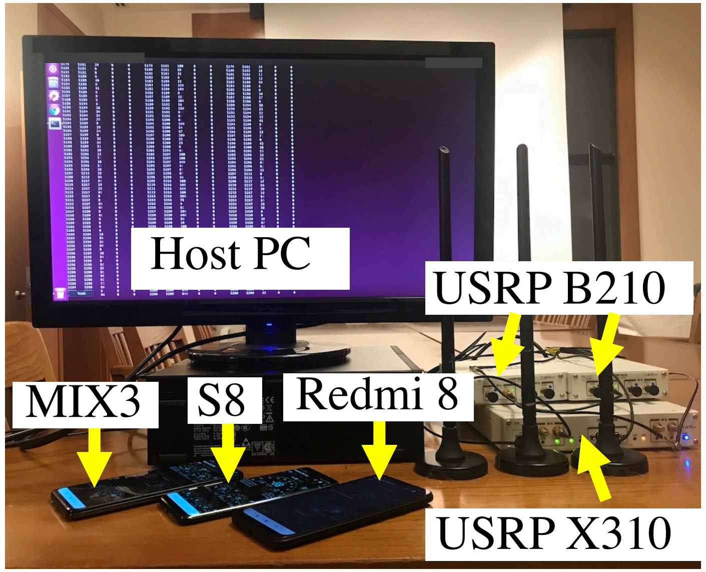
Transport layer capacity (or data rate): C_t

Physical layer capacity (or data rate): C_p

Implementation

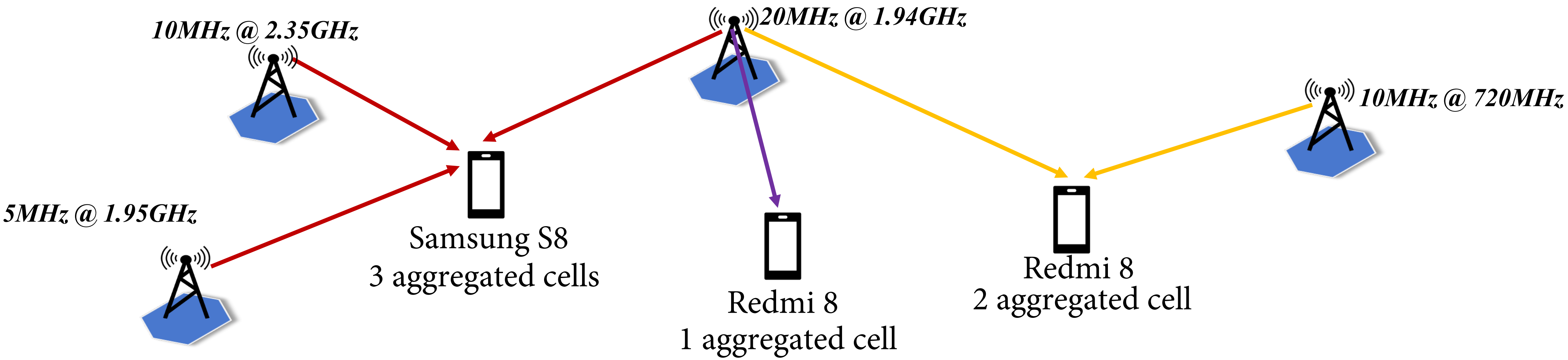
- Programming a mobile phone to decode every control message transmitted over the physical control channel requires customization of the **cellular firmware** inside the phone
- We build an open-source congestion control prototyping platform that supports control message decoding, bypassing the need to customize firmware





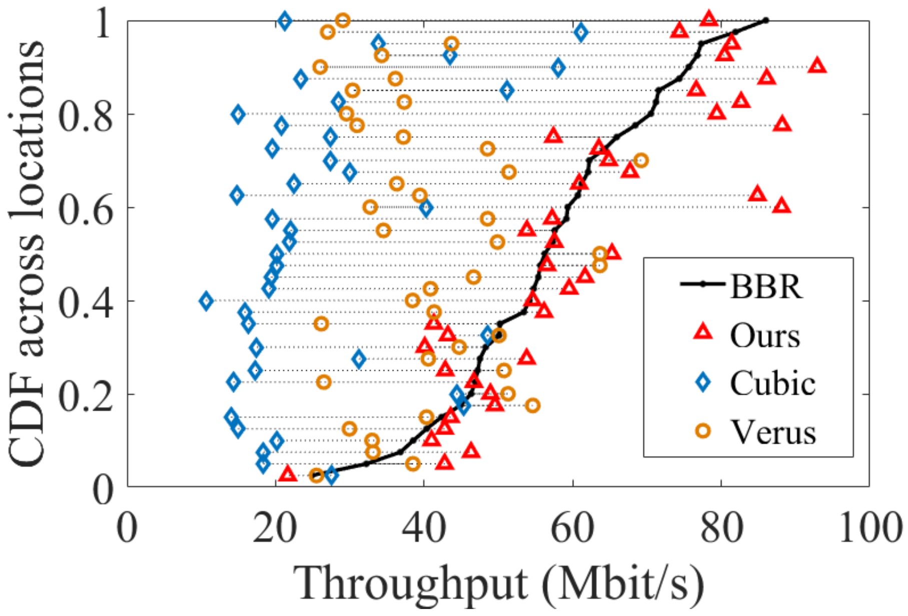
Evaluation: Methodology

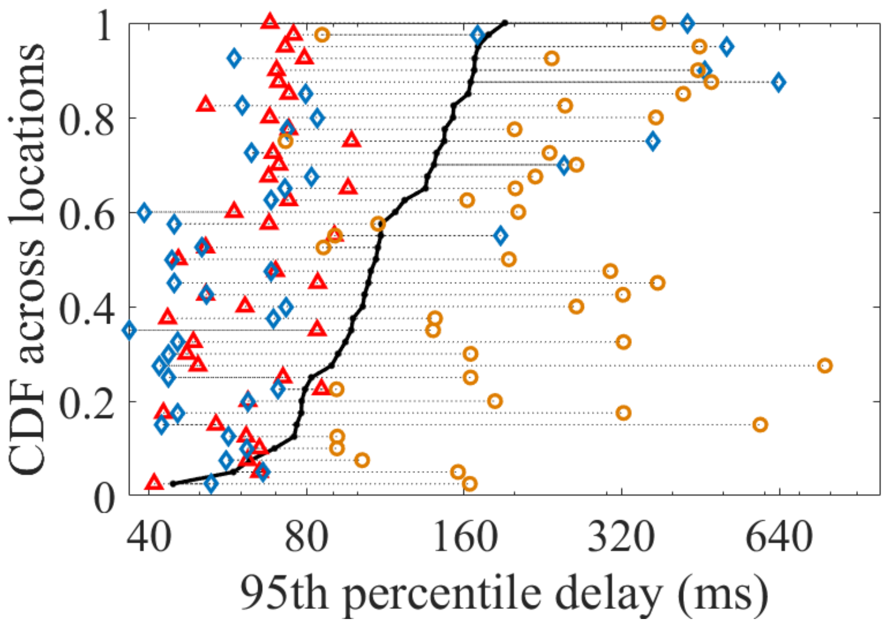
- **Sender:** We configure Amazon AWS servers as the PBE-CC senders.
- **Mobile Clients:** We use three mobiles: Xiaomi MIX3, Redmi 8, Samsung S8
- **Algorithms to compare:**
  - Algorithms designed for cellular networks: Sprout, Verus
  - Algorithms included inside the Linux Kernel: BBR, CUBIC
  - Recently proposed algorithms in top conferences: Copa, PCC, PCC-Vivace



Throughput and delay improvement for static users

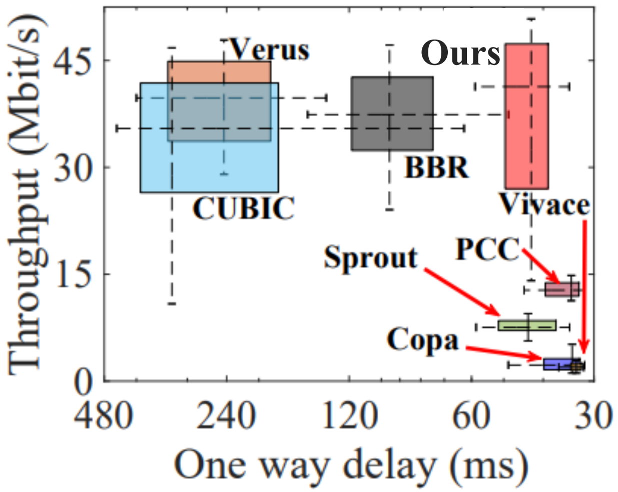
- **<u>Methodology:</u>**
  - We test 40 locations, covering all combinations of indoor/outdoor, one/two/three aggregated cells, busy/idle network conditions
  - We repeat the experiment at each location with different congestion control algorithms
- Comparison among high-throughput algorithms:



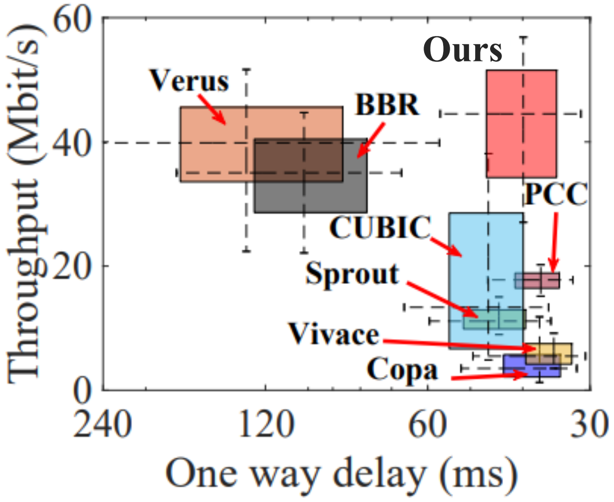


Throughput and delay improvement for static users

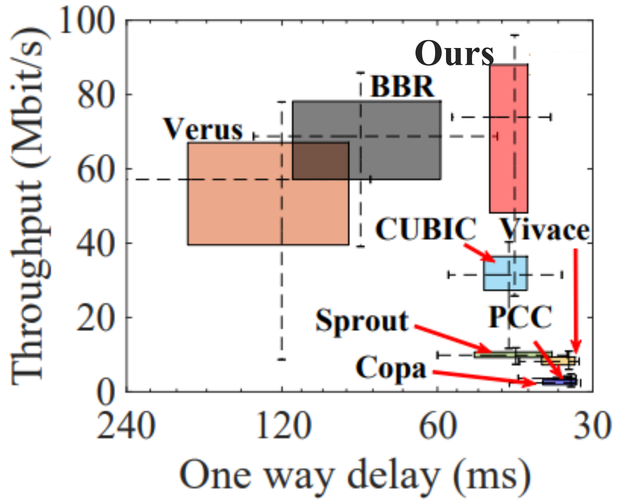
- **Methodology:**
  - We test 40 locations, covering all combinations of indoor/outdoor, one/two/three aggregated cells and busy/idle network conditions
  - We repeat the experiment at each location with different congestion control algorithms
- Detailed comparison among eight algorithms



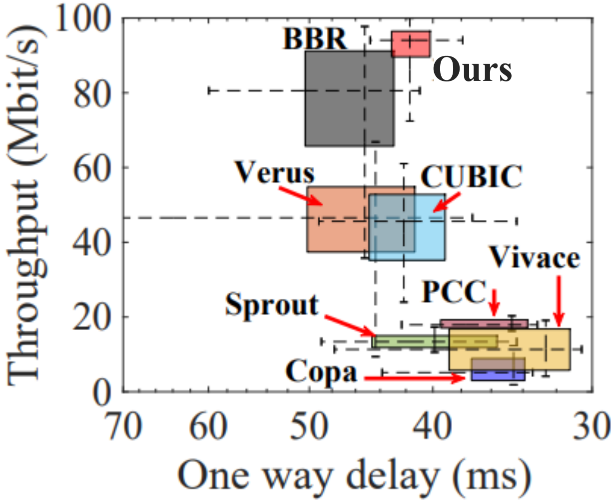
Indoor, busy hours, 1 cell



Indoor, busy hours, 2 cells



Indoor, busy hours, 3 cells



Indoor, idle hours, 3 cells

Conclusion

- First e2e congestion control to **seamlessly integrate mobile client-side wireless physical layer capacity measurement** into its design
  - Crucial for the multi-cell design of **4G and 5G wireless** networks
- **Outperforms** BBR, CUBIC, Copa, and many other leading congestion control algorithms in both latency and throughput
  - 6.3% **higher average throughput** than BBR, while **simultaneously reducing 95th percentile delay by 1.8×**

arxiv.org/pdf/2002.03475.pdf