# DChannel: Accelerating Mobile Applications With Parallel High-bandwidth and Low-latency Channels


**William Sentosa (UIUC)**

Balakrishnan Chandrasekaran (VU Amsterdam)

Brighten Godfrey (UIUC and VMware)

Haitham Hassanieh (EPFL)

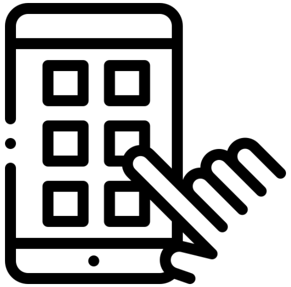
Bruce Maggs (Duke university and Emerald Innovations)





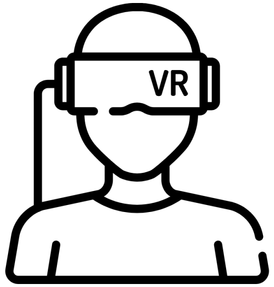

# Low latency is critical


**Web browsing**



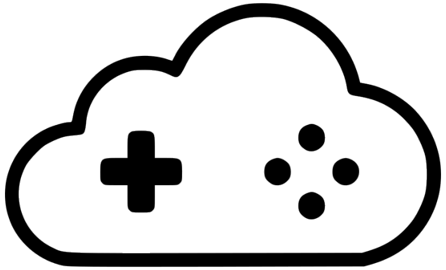
**Interactive mobile apps**

“**Additional 100ms** of latency can result in as much as 1% of revenue loss (Amazon)” [1]




**Virtual reality**

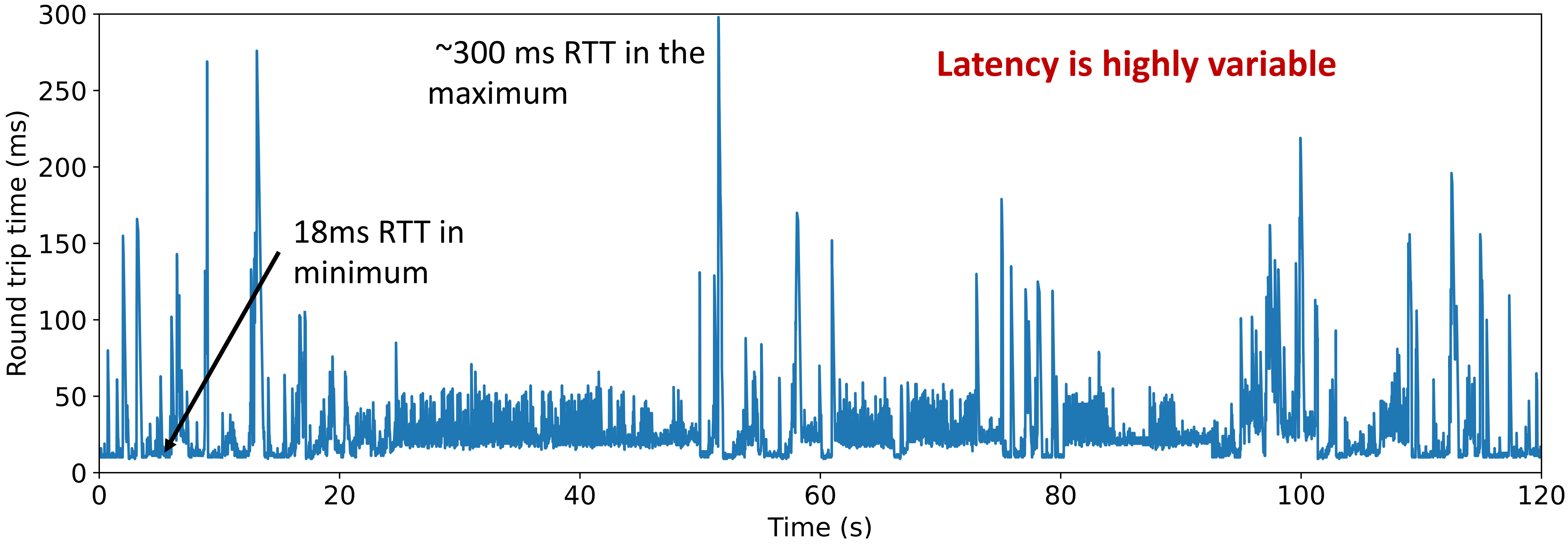
“**Require < 20ms latency** to avoid simulator sickness” [2]



**Cloud gaming**

“For the best gaming experience, **network latency should be < 10ms**” [3]

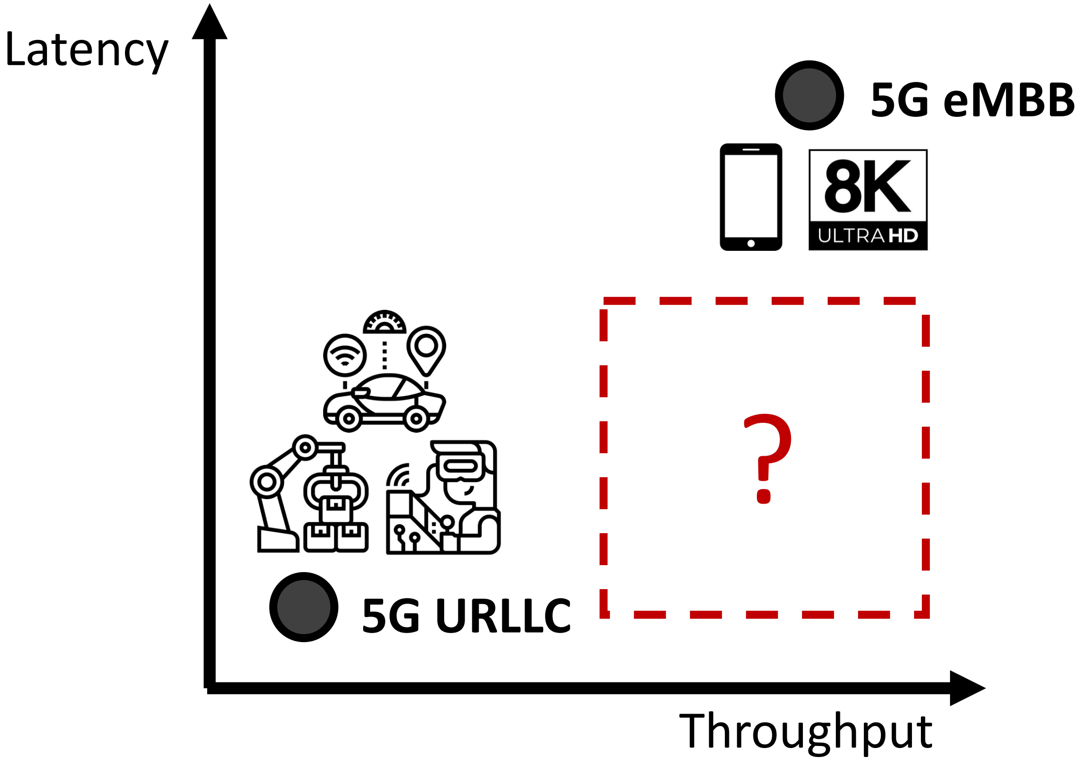
# What does current 5G latency look like?




End-to-end network RTT* when driving in downtown Chicago with Verizon 5G mmWave

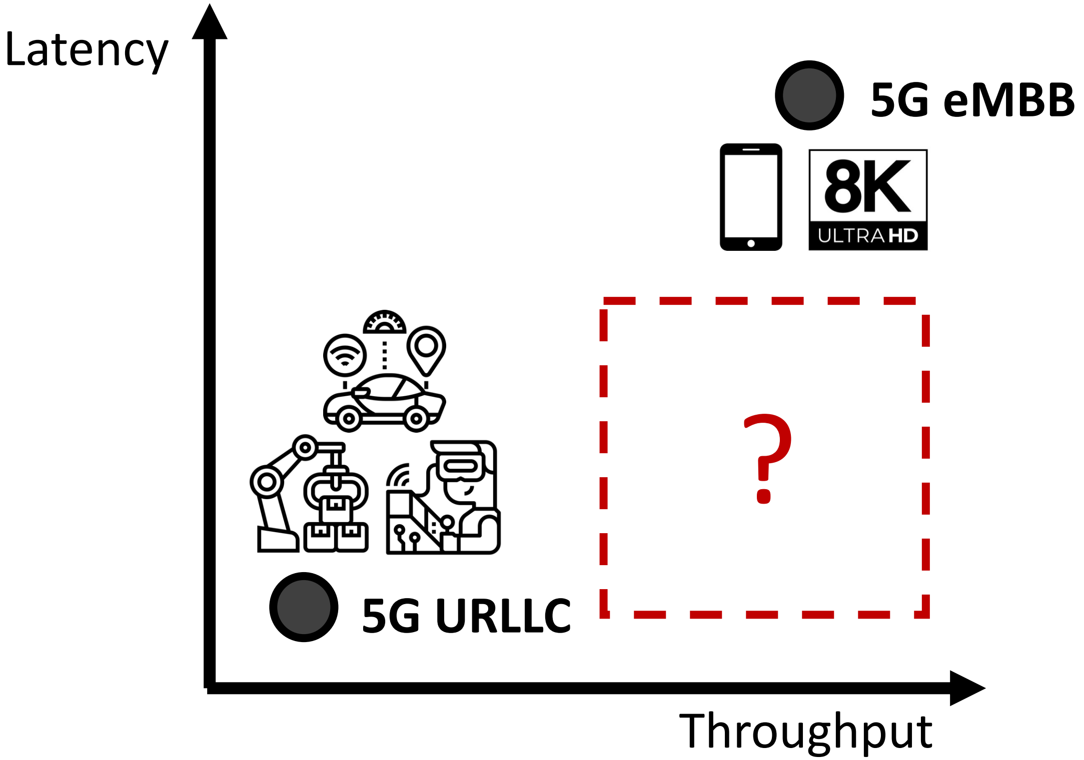
*measured by sending a single 1400 bytes packet to echo server in every 15ms.

# Channels (services) tradeoff: Latency vs. Throughput



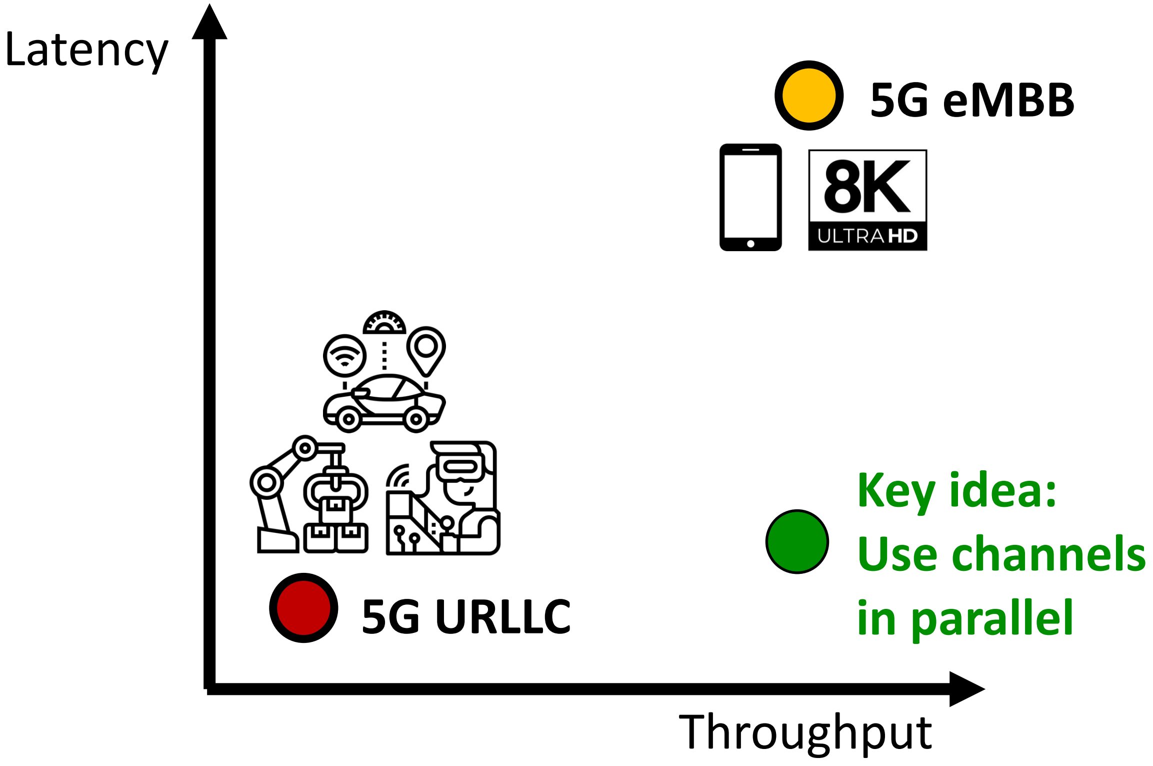

- Enhanced Mobile Broadband (**eMBB**)
  - Up to 2Gbps throughput
  - High and inconsistent latency
- Ultra Reliable and Low Latency Communication (**URLLC**)
  - 0.4Mbps – 16Mbps per user [6]
  - 2-10ms end-to-end RTT [6]

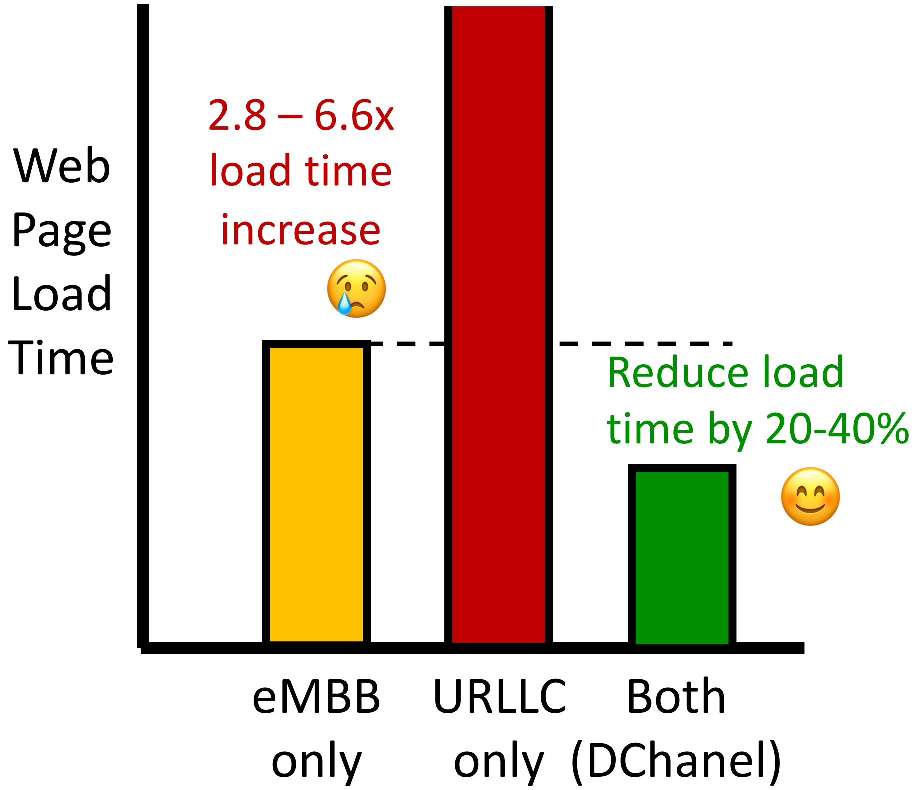
# Channels (services) tradeoff: Latency vs. Throughput



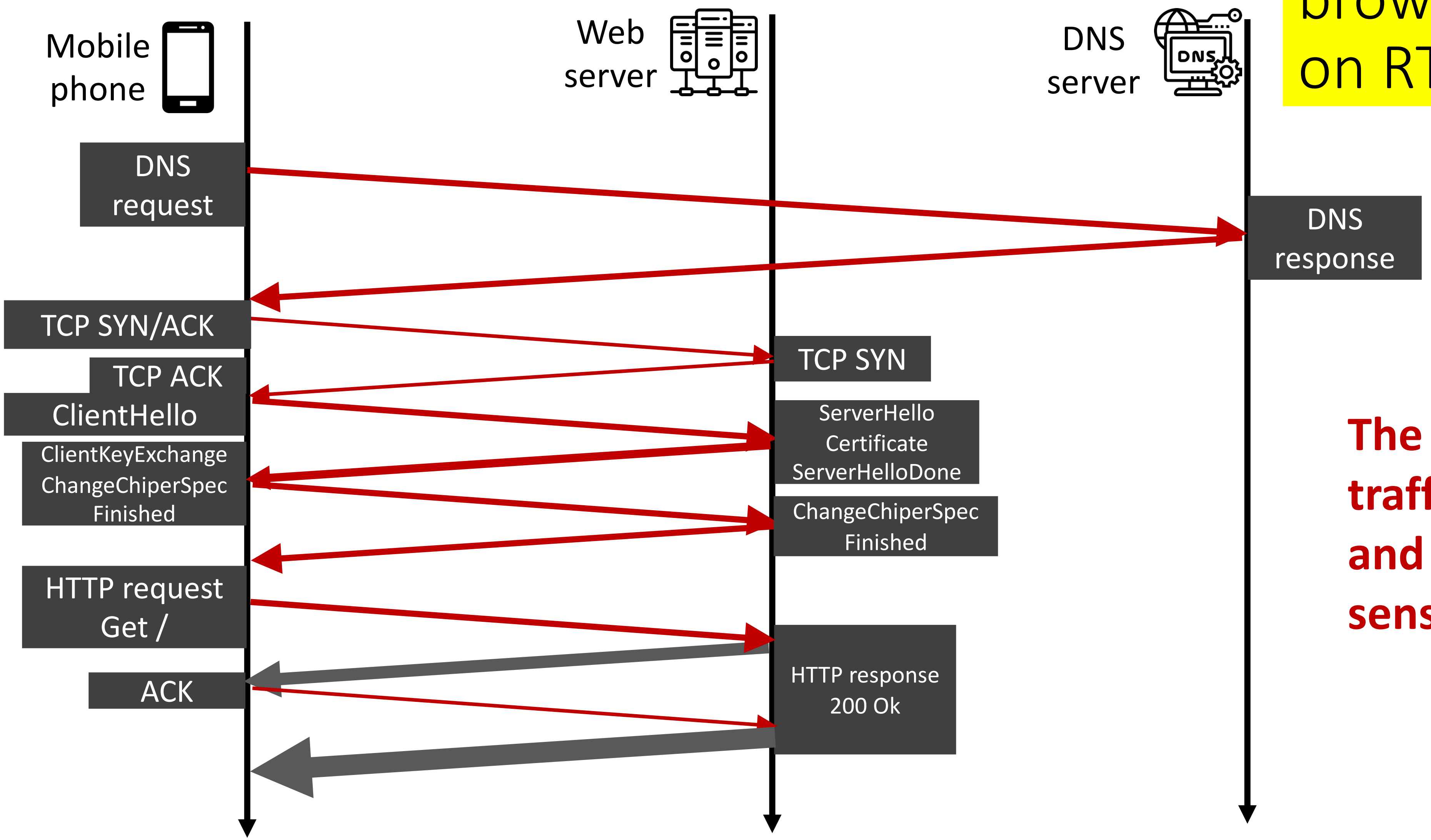

Can we break the latency throughput tradeoff?

# Breaking through the tradeoff barrier



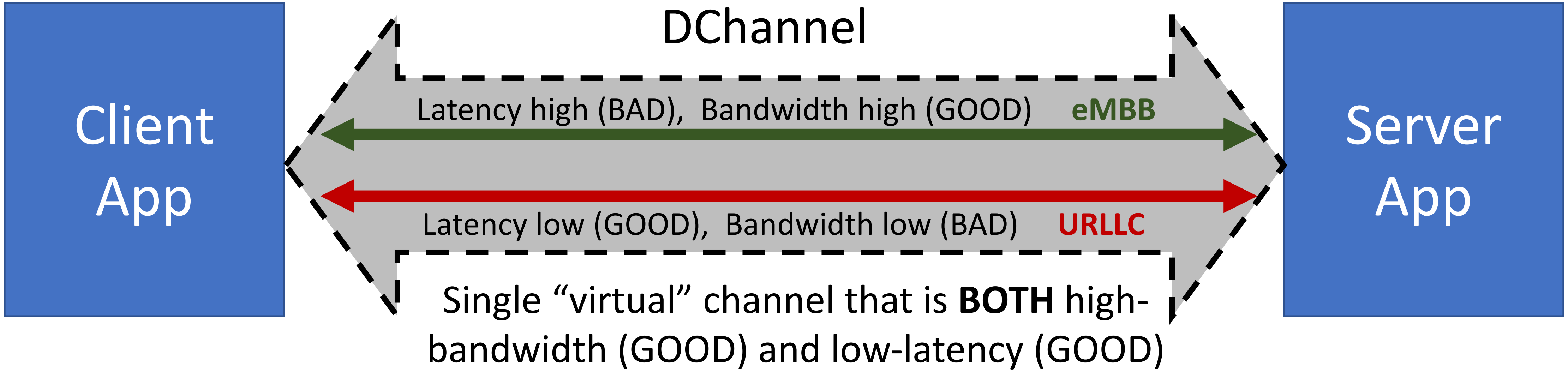



# Intuition: web browsing depends on RTT



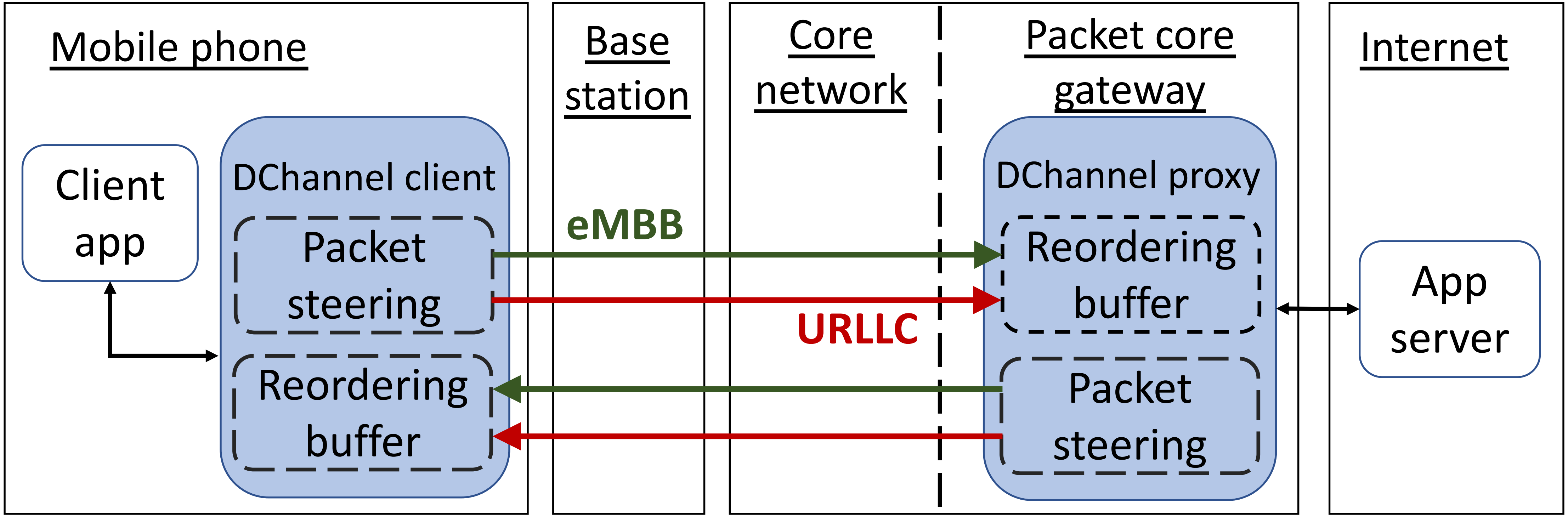

**The highlighted traffic are tiny and latency sensitive**

# DChannel: Leveraging multiple channels in 5G




- Steering traffic in the network layer (IP packets)
- Steer traffic transparently without app input

# DChannel architecture for 5G



# DChannel design decision

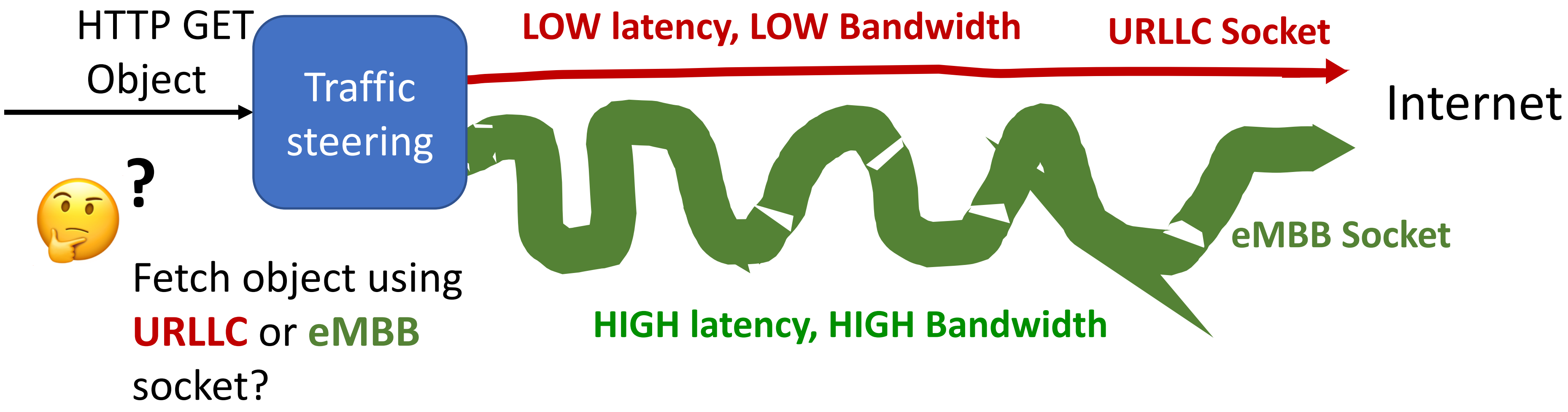
- ❖ Granularity of the traffic steering

- ❖ Steering heuristic

# Steering granularity

- In which granularity should we steer the traffic?

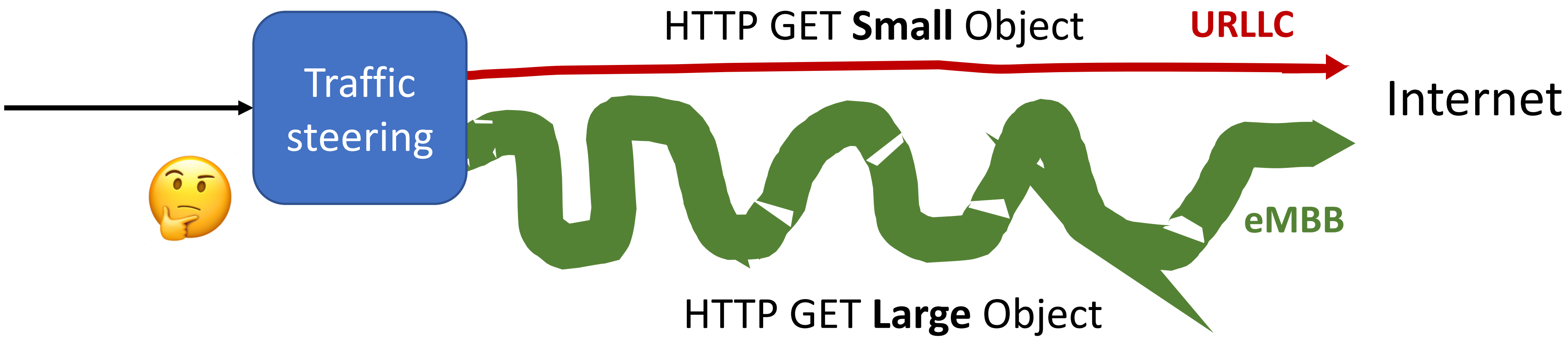
Option 1: **Application-level** (web objects, e.g., HTML, images, CSS)



# Steering granularity

Problems:

1. Misses opportunity to accelerate DNS, TCP/SSL/TLS handshake, HTTP request, acks, etc.
2. Requires app-level information

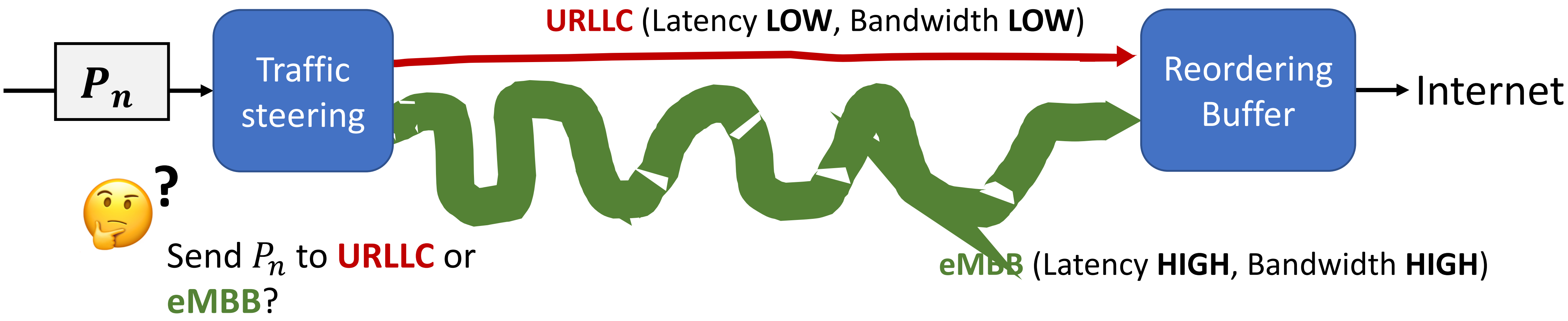


# Steering granularity (Network-level)

## Option 2: Network-level (IP Packets)

Key idea: only offloads **small portion** of traffic to LLC.
Also, it should **give most benefit**

Solution: **Cost-rewards analysis**

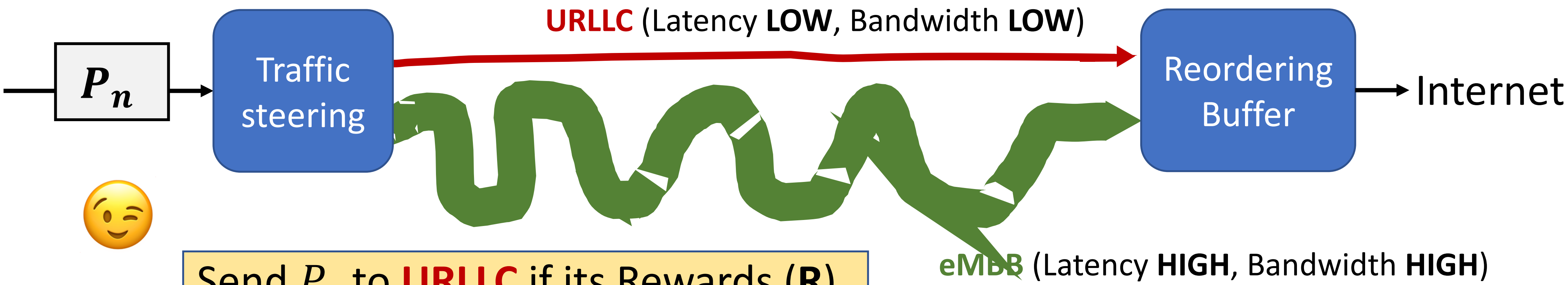


# Steering granularity (Network-level)

## Option 2: Network-level (IP Packets)

Key idea: only offloads **small portion** of traffic to LLC.
Also, it should **give most benefit**.

Solution: **Cost-rewards analysis**

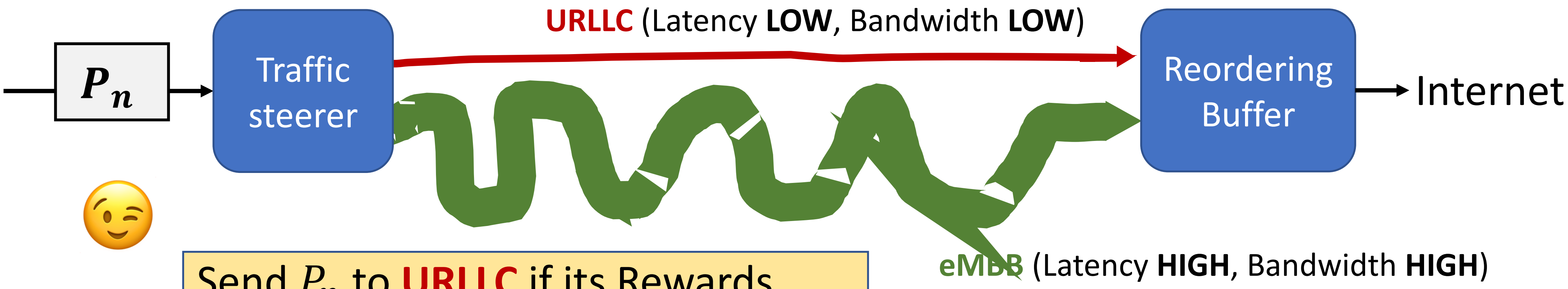



Send $P_n$ to **URLLC** if its Rewards (**R**) outweighs its Cost (**C**)

# Cost and rewards analysis

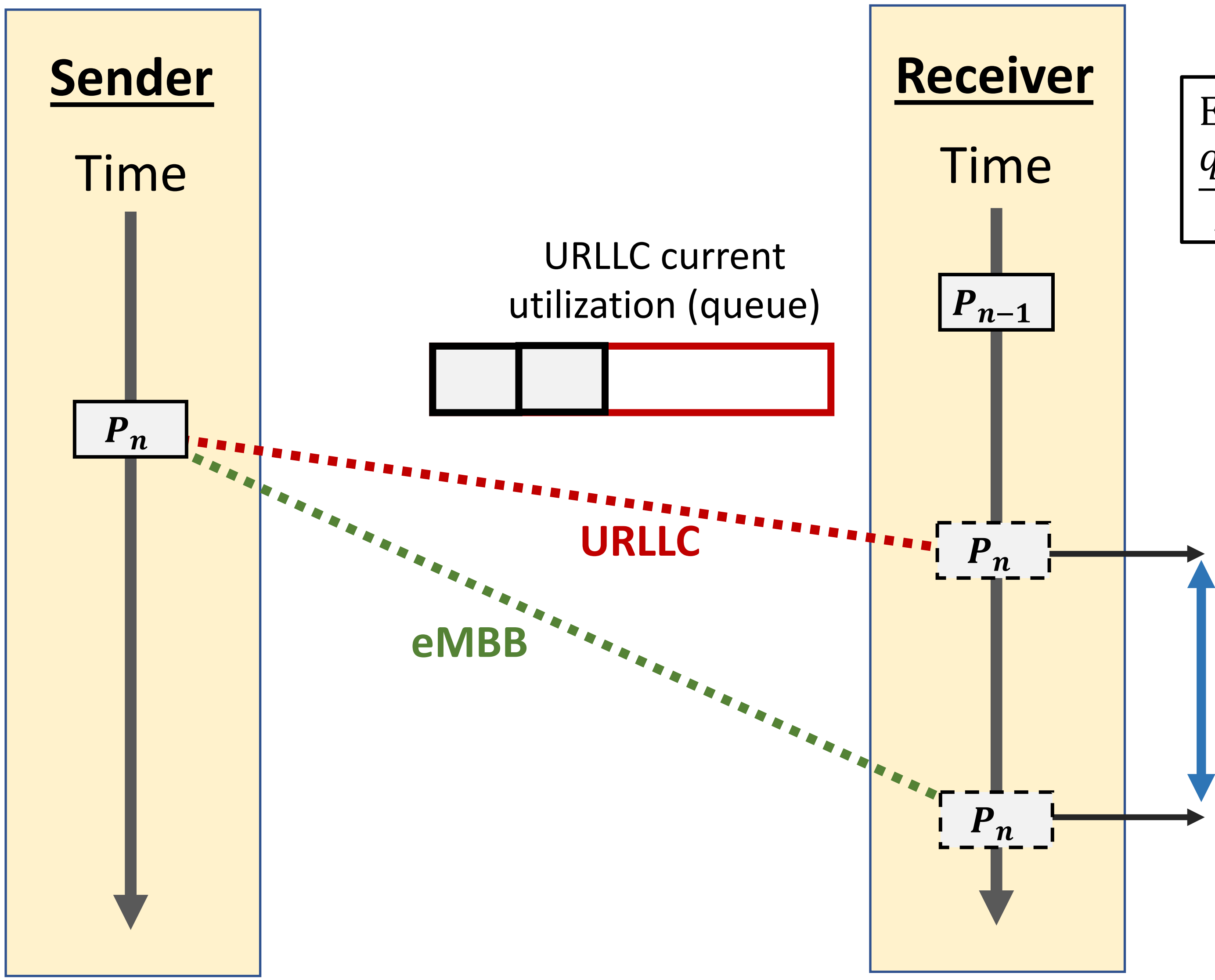
Rewards: If we use **URLLC**, how much faster will a packet be delivered?

Cost: How much will the **URLLC** utilization increase?




Send $P_n$ to **URLLC** if its Rewards outweighs its Cost

# Estimating Rewards



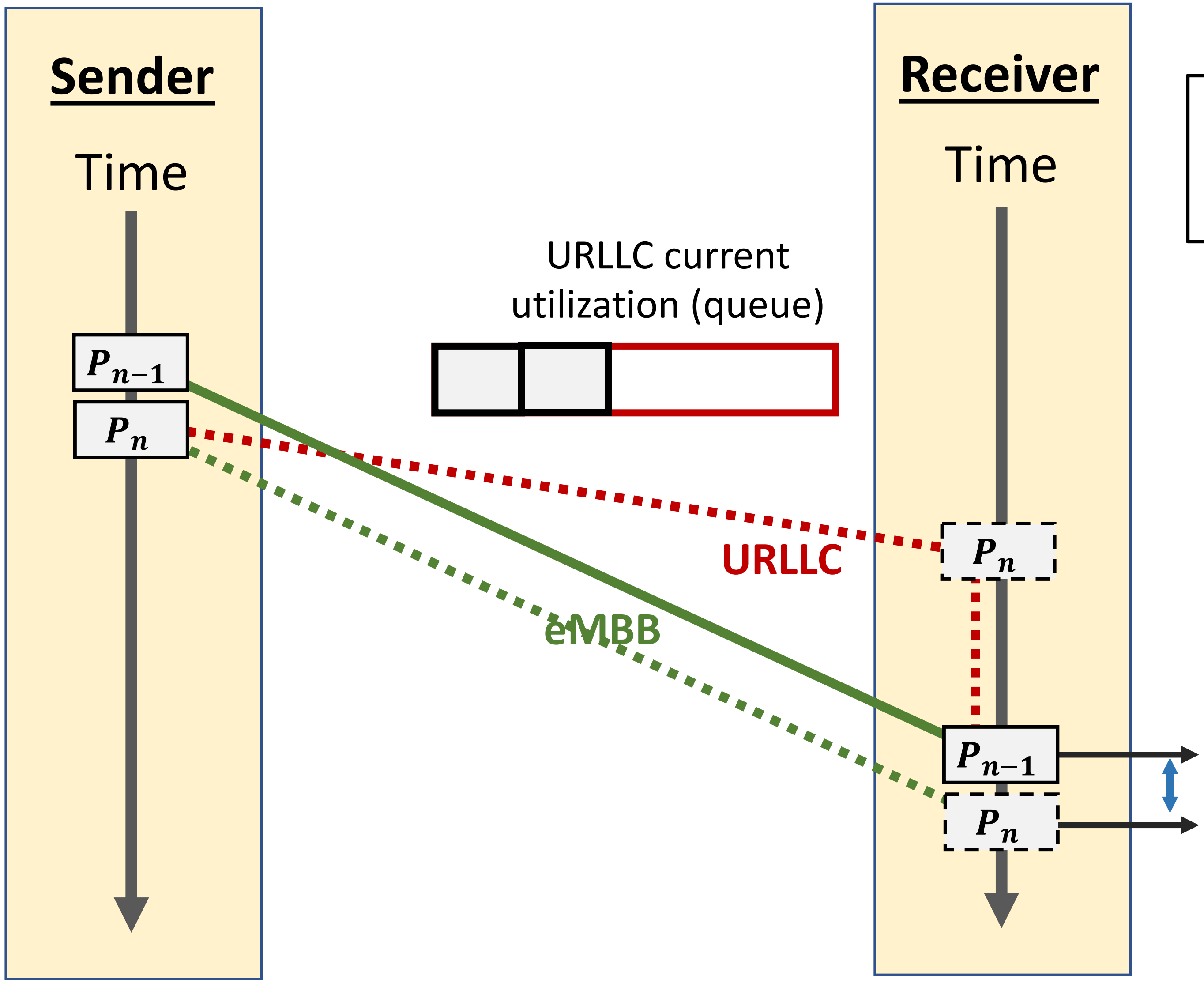

$$\text{Est. } delivery\ time = owd + \frac{queue\ depth}{bandwidth} + \frac{packet\ size}{bandwidth}$$

Arrival time if $P_n$ to **URLLC**

$$\boldsymbol{Rewards = Delivery_{eMBB} - Delivery_{URLLC}}$$

Arrival time if $P_n$ to **eMBB**

# Estimating Rewards




$$Est.\, delivery\ time = latency + \frac{queue\ depth}{bandwidth} + \frac{packet\ size}{bandwidth}$$

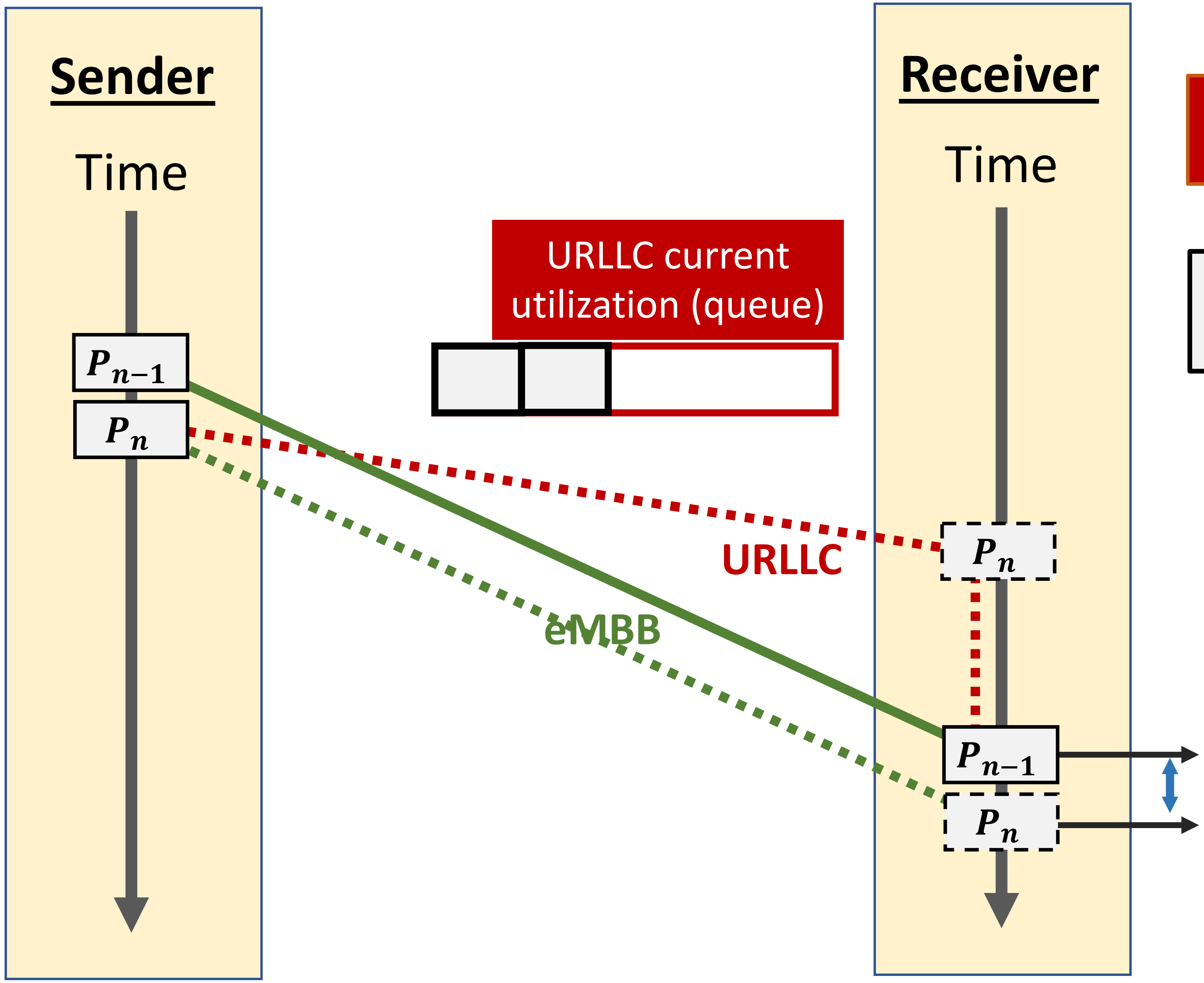
Should we still send $P_n$ to **URLLC**???

**Rewards is close to 0**

Arrival time if $P_n$ to **URLLC**

Arrival time if $P_n$ to **eMBB**

# Estimating Cost




$$Cost = \frac{\text{Queue depth}_{urllc} + packet\ size}{\text{bandwidth}_{\text{urllc}}}$$

Send to URLLC if:
$Rewards > \alpha * Cost$

Should we still send $P_n$ to **URLLC**??? Maybe not!!

**Rewards is close to 0**

# More in the paper

- Picking a good $\alpha$
- Estimating the network latency and queue depth
- Performance under wrong latency estimates
- Details on the reordering buffer

# Implications of cost-rewards heuristics

$$Cost = \frac{\text{Queue depth}_{urllc} + packet\ size}{\text{bandwidth}_{\text{urllc}}}$$

$$\boldsymbol{Rewards = Delivery_{eMBB} - Delivery_{URLLC}}$$

Send to URLLC if:
$R > \alpha * F$

1. Small packets and short packet sequences tend to be steered to **URLLC** (**Rewards=high**, **Cost=low**)

2. Long back-to-back packet sequences tend to be steered to **eMBB** (**Rewards=low**, **Cost=high**)

The implications suit well with the idea of **accelerating control packets** (such as TCP SYN and ACK) and **small messages** (such as HTTP request)

# Evaluation

- How does 5G eMBB+URLLC (DChannel and existing schemes) improve application performance compared to eMBB-only?

- Existing schemes:
  - MPTCP [8]
  - ASAP [9]

- Tested apps:
  - Web browsing
  - Web-based mobile apps (e.g., Reddit, eBay)
  - Bulk download

[8] Multipath TCP in the Linux Kernel v0.94. http://www. multipath-tcp.org, March 2018.
[9] Se Gi Hong and Chi-Jiun Su. ASAP: fast, controllable, and deployable multiple networking system for satellite networks. In IEEE Global Communications Conference (GLOBECOM), 2015.

# Experimental setup: network emulation

## 5G eMBB

### Record-and-replay emulation

5G mmWave

5G low band

Conditions: Stationary, walking, and driving

### Live network

5G mmWave

5G low band

Conditions: Stationary

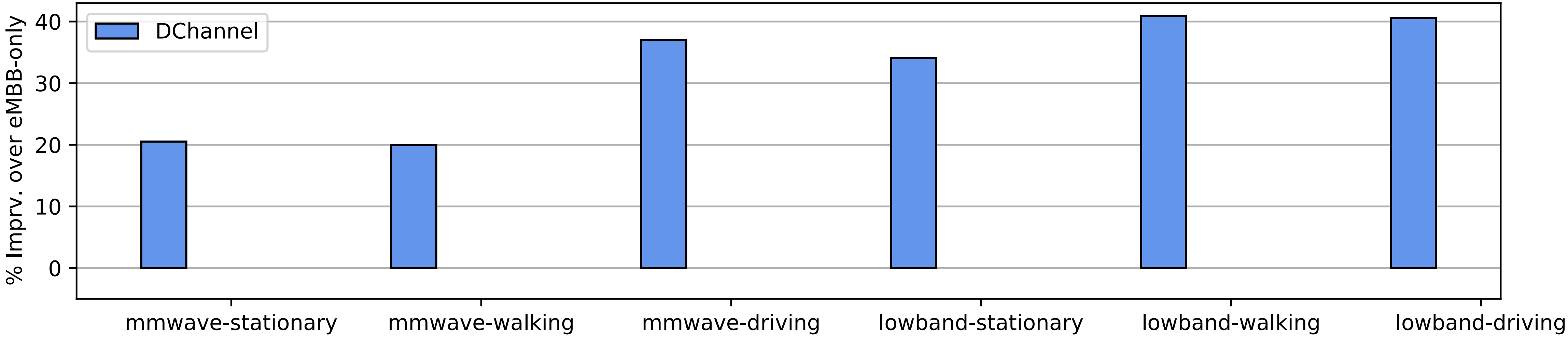
## 5G URLLC

Emulated:
5ms RTT
2 Mbps

# Evaluation results (DChannel)

- DChannel improves web browsing by **~20 – ~40% compared to eMBB-only**

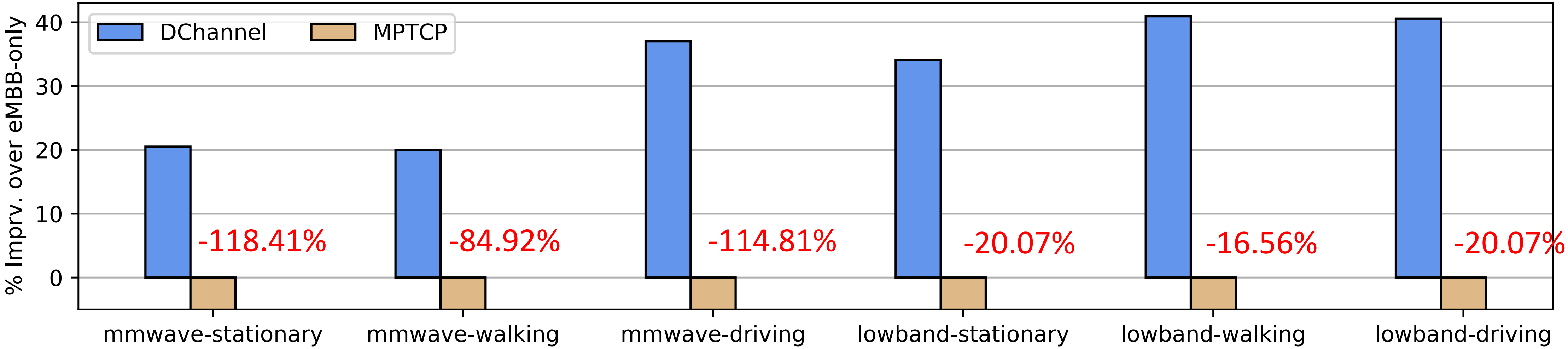
URLLC is at **2Mbps + 5ms RTT**



# Evaluation results (MPTCP)

- MPTCP performs worse than eMBB-only because the paths are asymmetrical

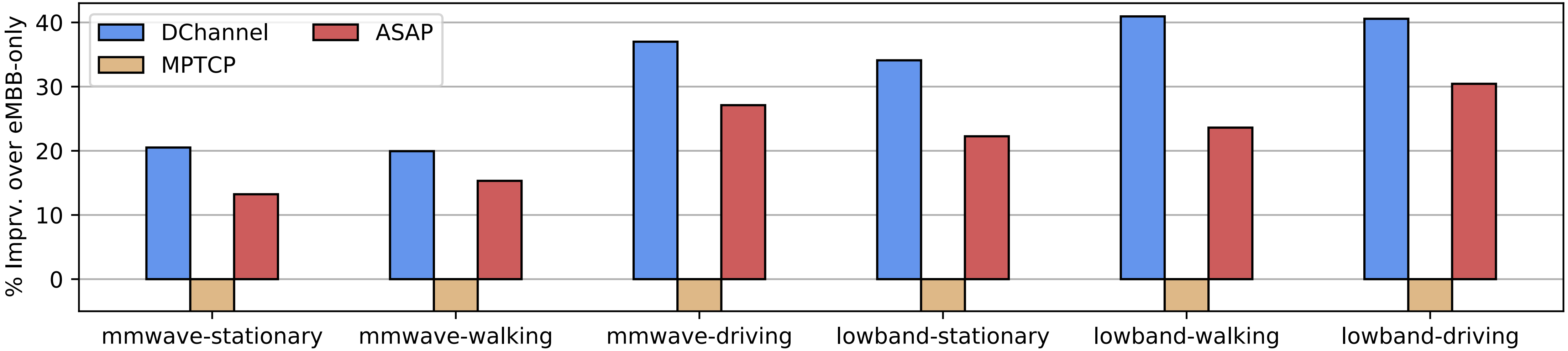
URLLC is at **2Mbps + 5ms RTT**



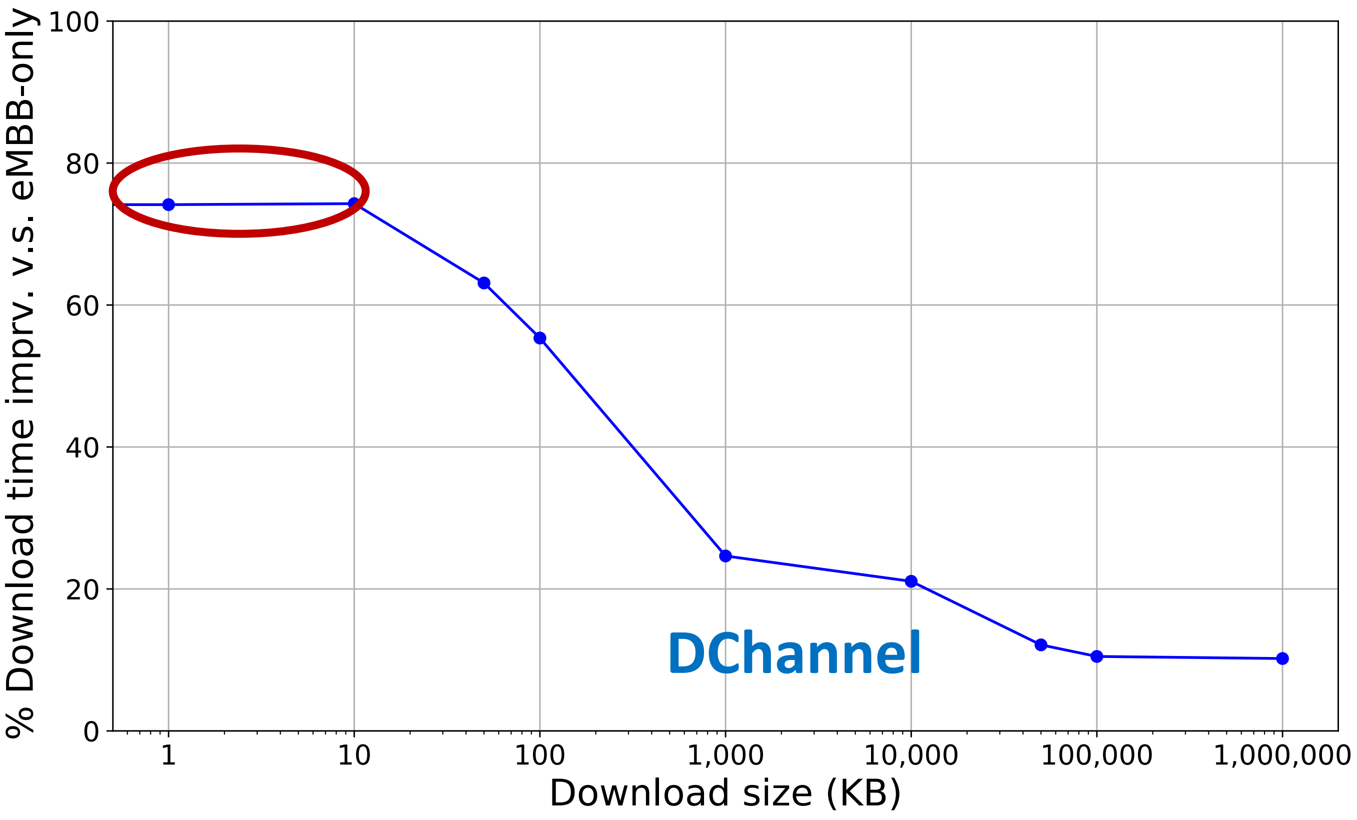
# Evaluation results (ASAP)

- ASAP [9] accelerates connection handshake and HTTP request traffic to low latency path but leaves HTTP responses to high bandwidth path.

URLLC is at **2Mbps + 5ms RTT**

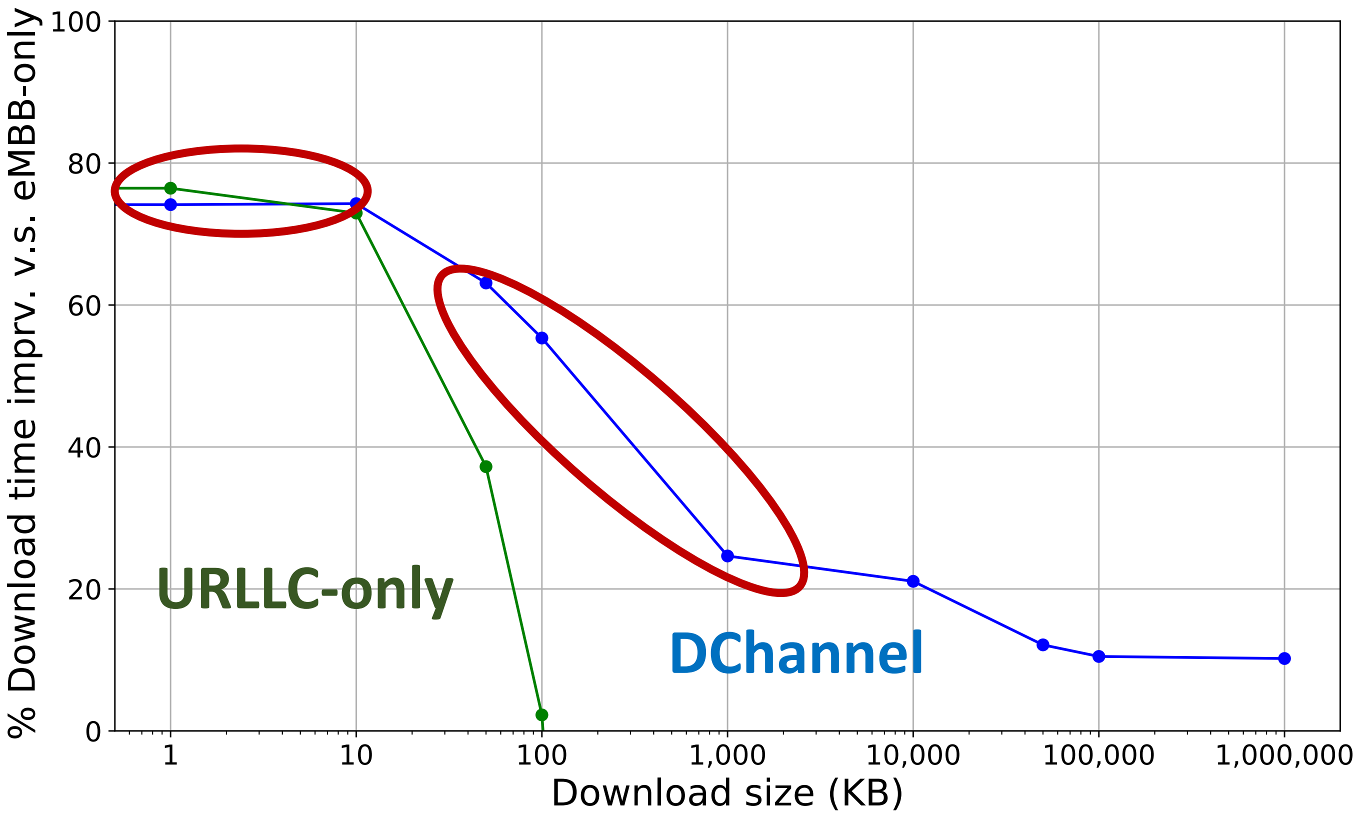


# Microbenchmark: Effect of transfer size



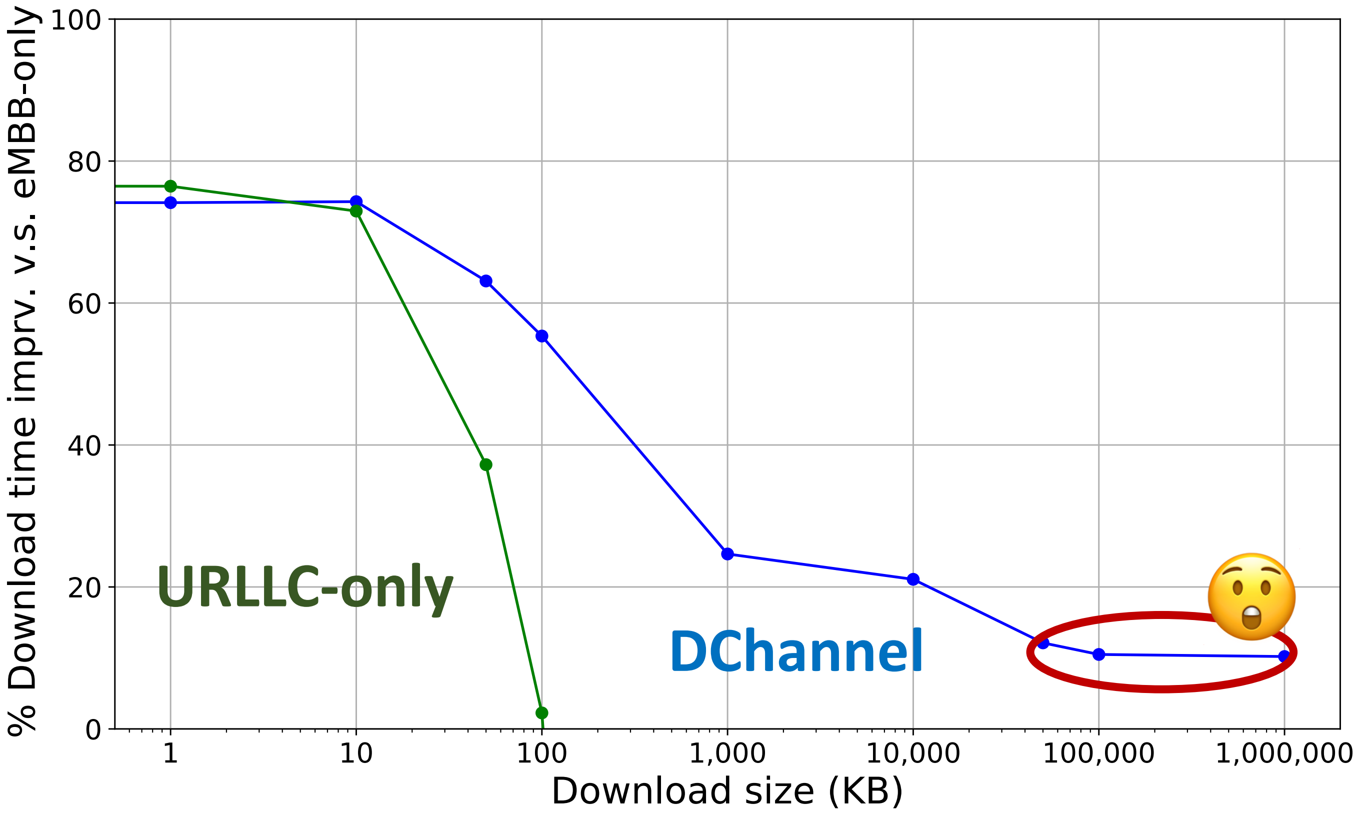

This experiment used the mmwave-driving trace

# Microbenchmark: Effect of transfer size




This experiment used the mmwave-driving trace

# Microbenchmark: Effect of transfer size




This experiment used the mmwave-driving trace

# More evaluation results in the paper

- ❖Live-5G eMBB experiment confirms DChannel performance gains
- ❖DChannel lowers mobile apps (e.g., reddit) response time by 21%
- ❖Reordering buffer (ROB) helps Dchannel
- ❖DChannel still give a good improvement even when URLLC latency fluctuates
- ❖DChannel still works with an incorrect latency estimates

# Conclusions and future directions

- Key takeaway:
  - Using URLLC+eMBB can give applications “illusion” of having a single channel that is both high bandwidth and low latency.

- Future directions:
  - Supporting specialized apps that requires **both high-bandwidth and low-latency** in mobile environment
    - Extended reality (VR/AR)
    - Cloud gaming
    - Remote driving

Thank you!
Contact: sentosa2@Illinois.edu