UCLouvain

Integrating a Multipath Transport Protocol in my Application

Challenges to Get Benefits

Quentin De Coninck (UCLouvain, Belgium) - November 24, 2022 - WOS'22
https://qdeconinck.github.io

fnrs
FREEDOM TO RESEARCH

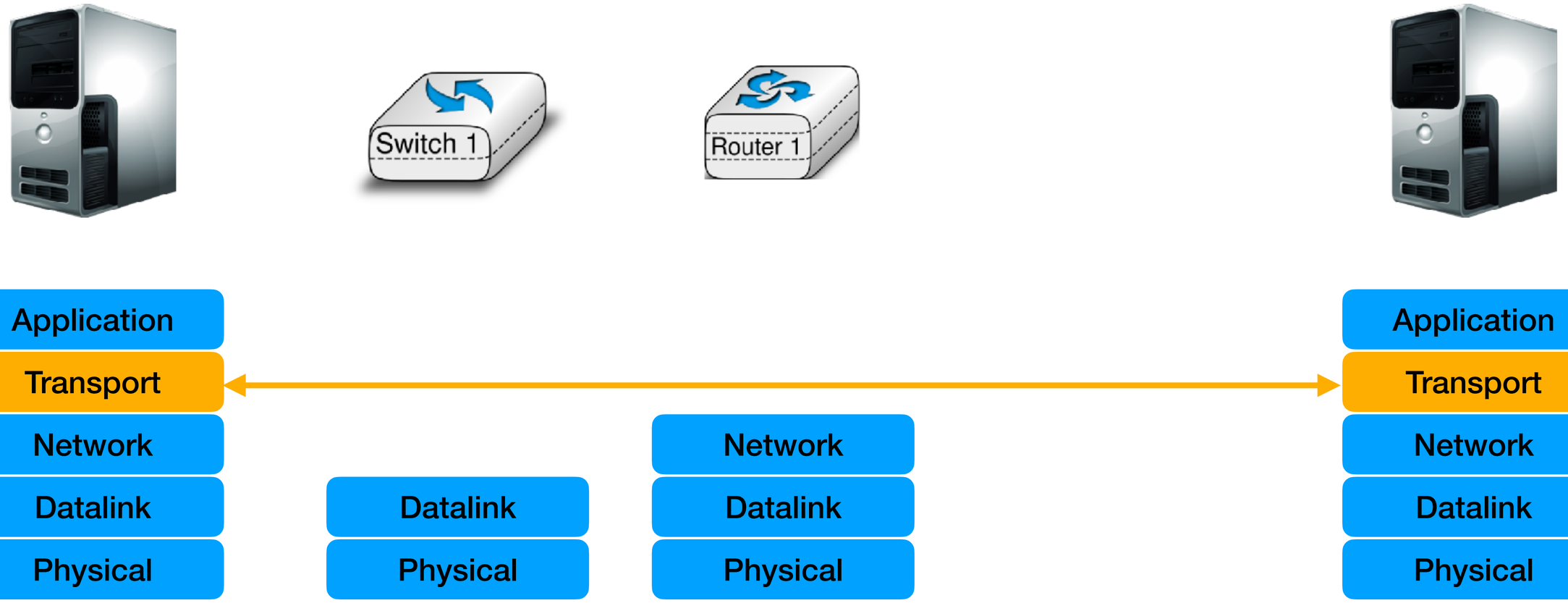
Quick Outline

- State-of-the-art: (Multipath) TCP
- Multipath in two simple applications
- (Multipath) QUIC
- Applications to Video Streaming

The Common (Transport) Protocol: TCP

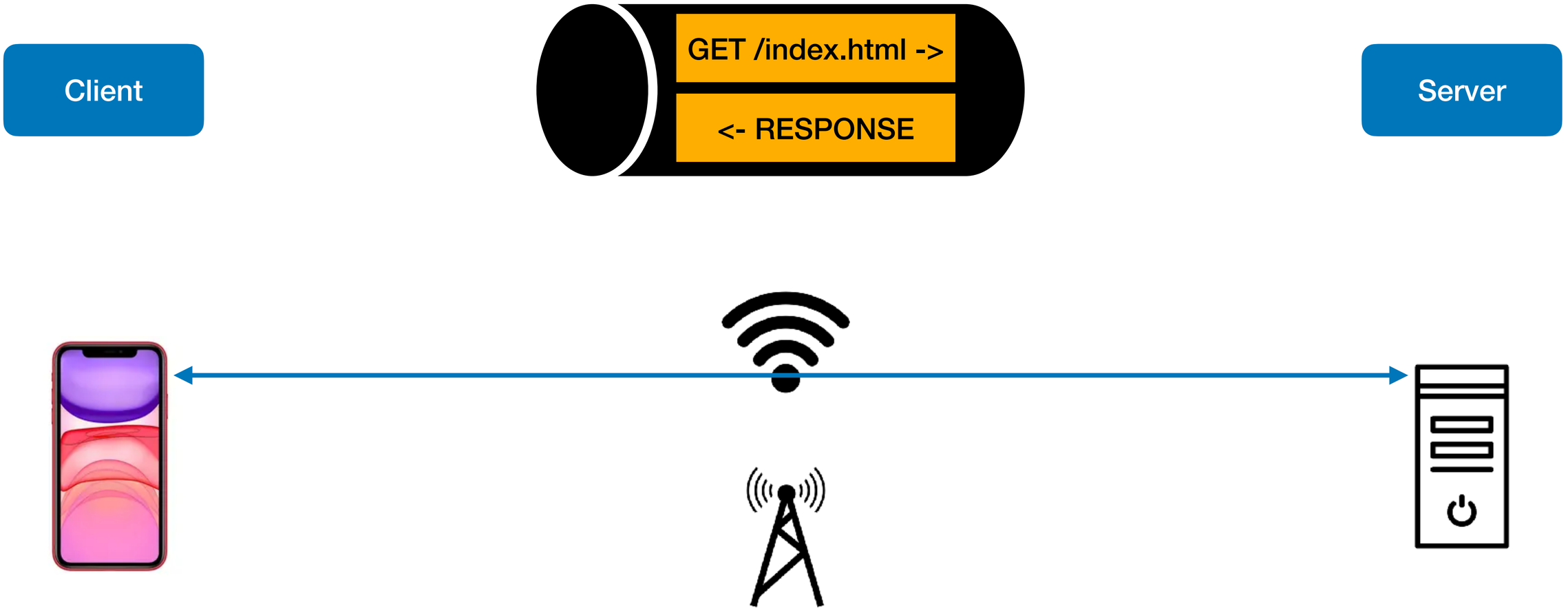
UDP being the other major one

End-to-End Principle



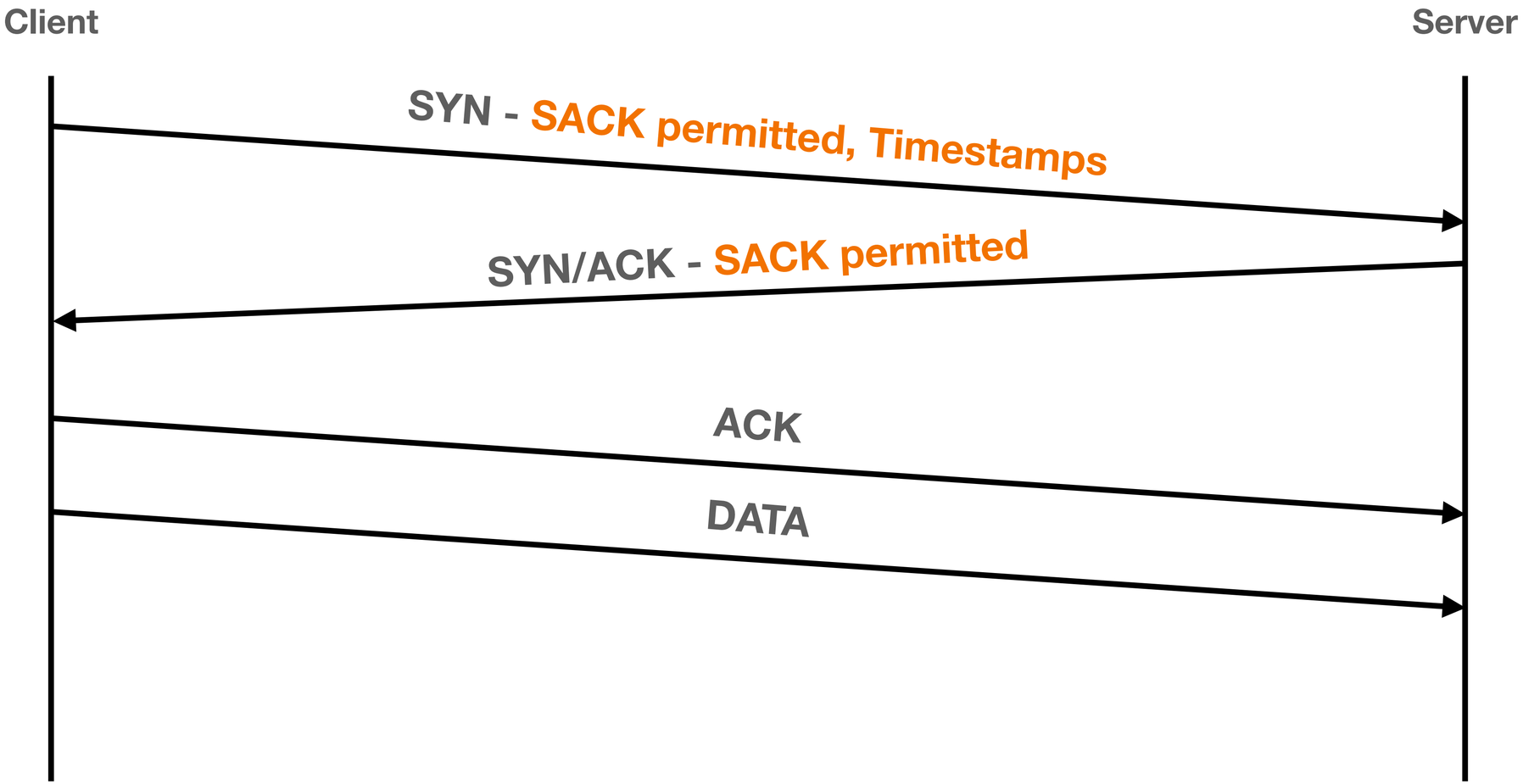
TCP and the Bytestream Service

Reliable, in-order delivery



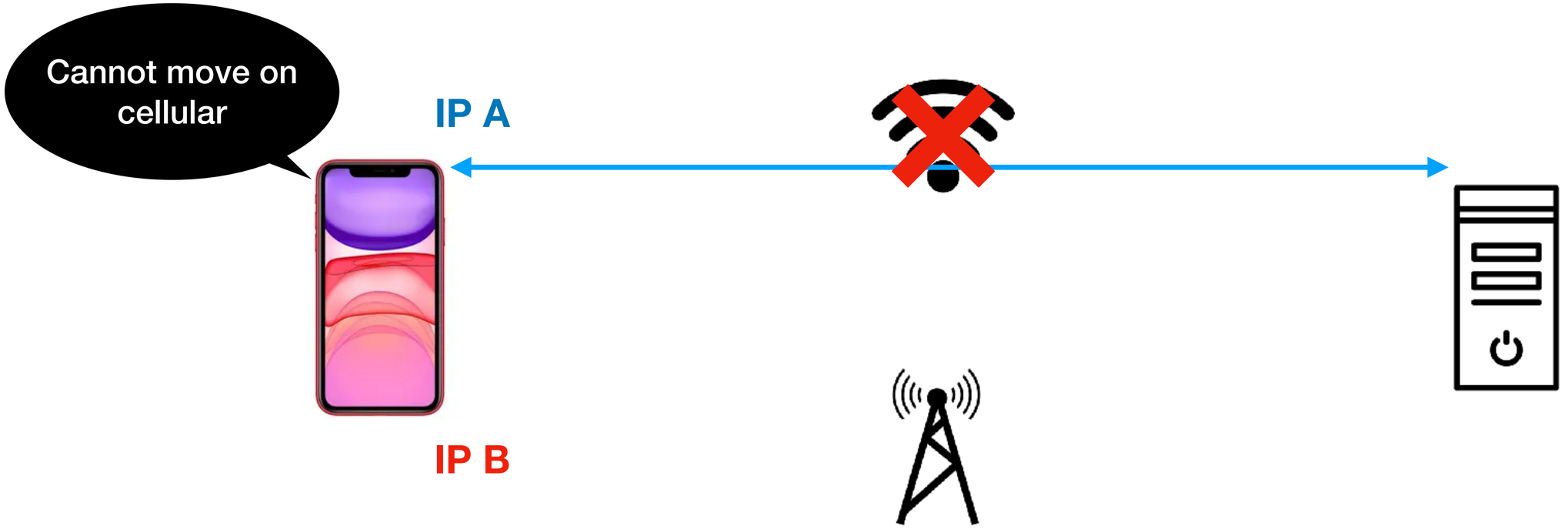
Establishing TCP Connections

a.k.a., the TCP Joke



The Mismatch with Mobile Devices

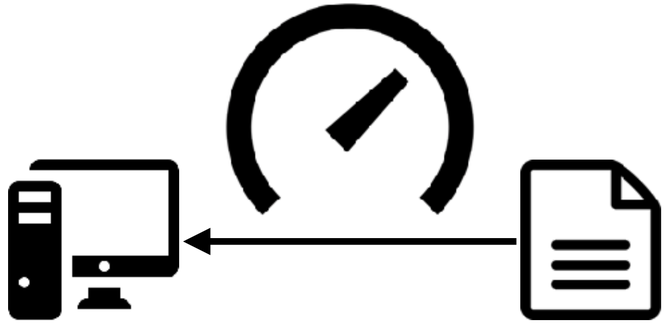
TCP Connections Are Bound to their 4-tuple (IP_{src}, $port_{src}$, IP_{dst}, $port_{dst}$)

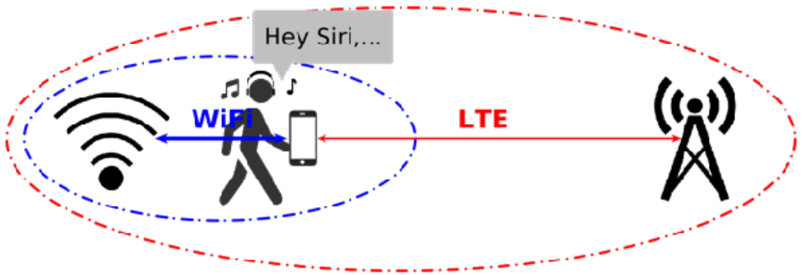


Losing the network path = Broken connectivity

Why Multipath?

- Bandwidth Aggregation
  - E.g., download your files faster
- Keeping Low Latency
  - E.g., non-stable condition, network handover,…
- Network Management
  - E.g., control where data is sent (privacy,…)



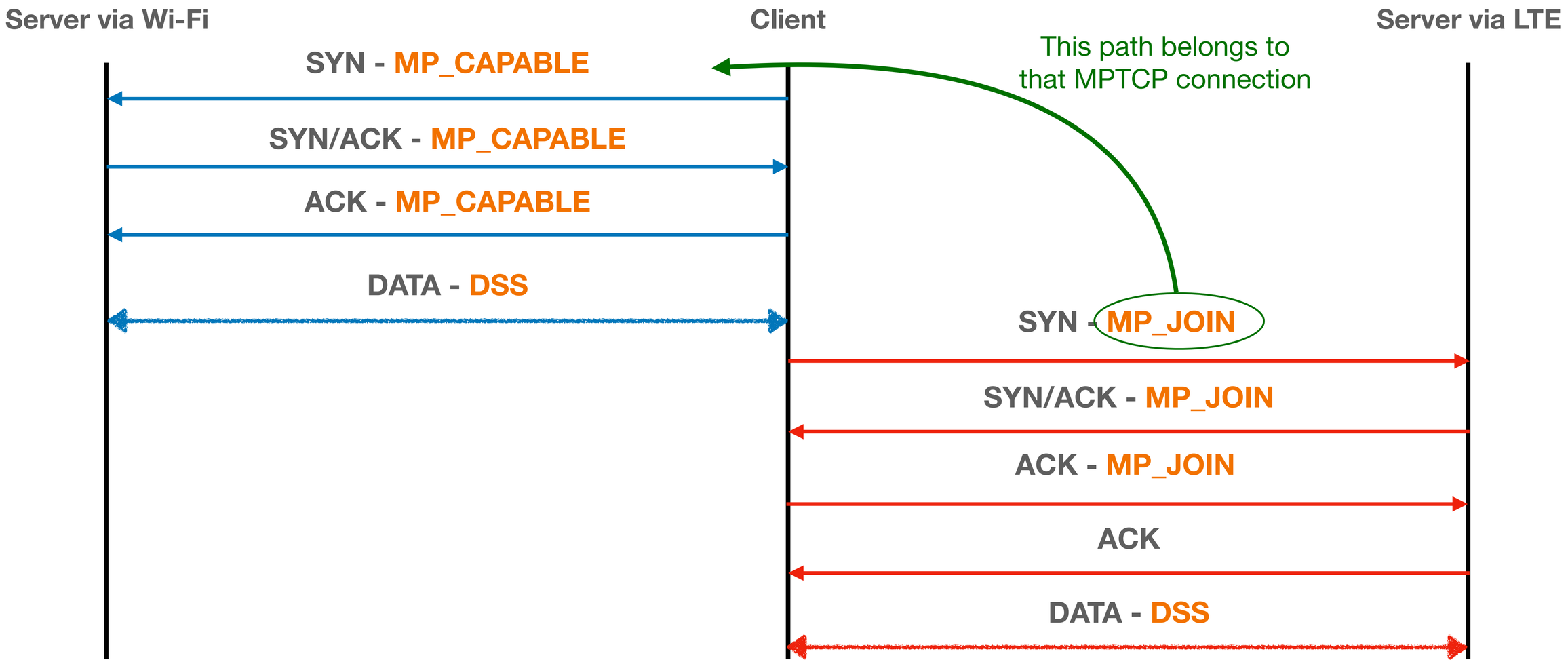


Bringing Multipath into TCP

RFC 6824
RFC 8684

Multipath TCP

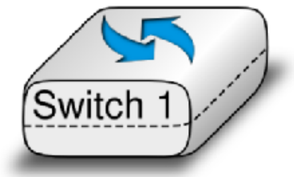
As a TCP extension, using TCP options



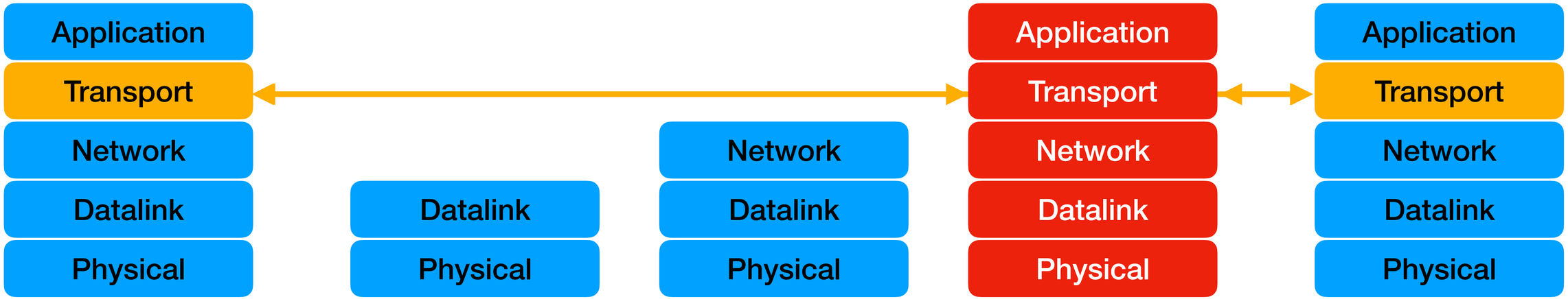
Sherry, Justine, et al. "Making middleboxes someone else's problem: Network processing as a cloud service." *ACM SIGCOMM Computer Communication Review* 42.4 (2012): 13-24.

End-to-End Principle: The Reality

TCP header (including TCP options) in clear-text!

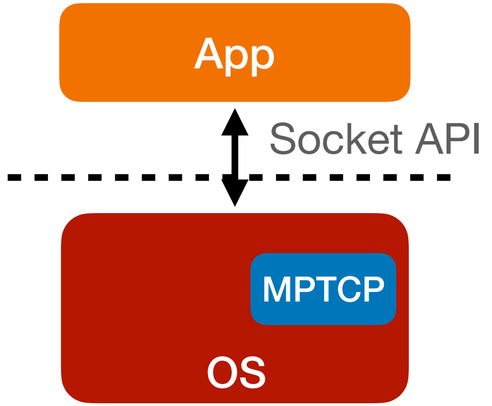


Firewalls
NATs
“Network optimisers”
…



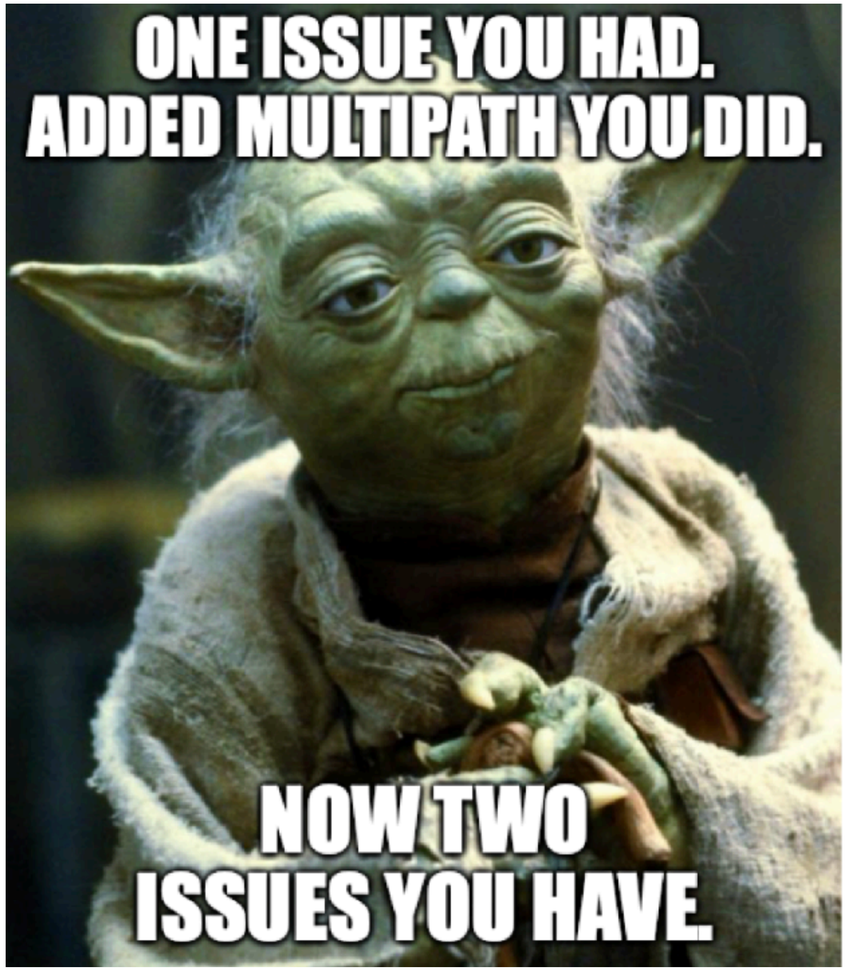
A Deployable Multipath TCP

- Design objective: Multipath TCP must work where TCP works
  - Each MPTCP path behaves likes a regular TCP connection
  - Network anomaly? Fallback to plain TCP
- Several **kernel-space** implementations (Apple, Linux,…)
  - App can do MPTCP without modification!



Multipath Algorithms

- Path management
  - Creating paths
- Packet scheduling
  - Using paths
- (Congestion control algorithm)
  - Per-path state

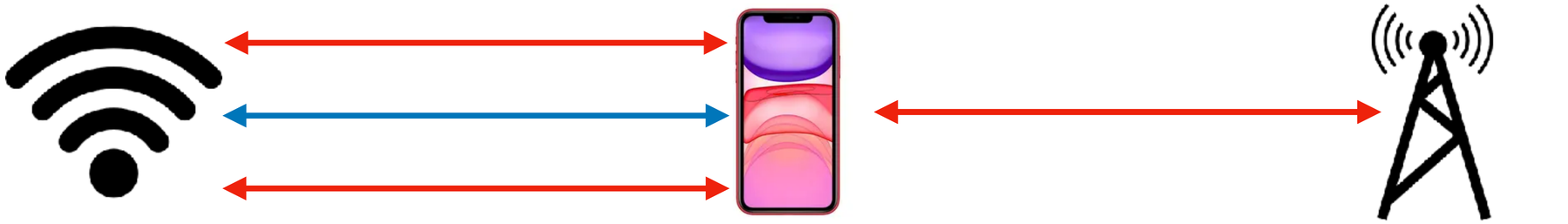


Path Management

Creating Paths

In-kernel implementation

Fullmesh: create one path per network interface as soon as possible



Server via Wi-Fi

Client

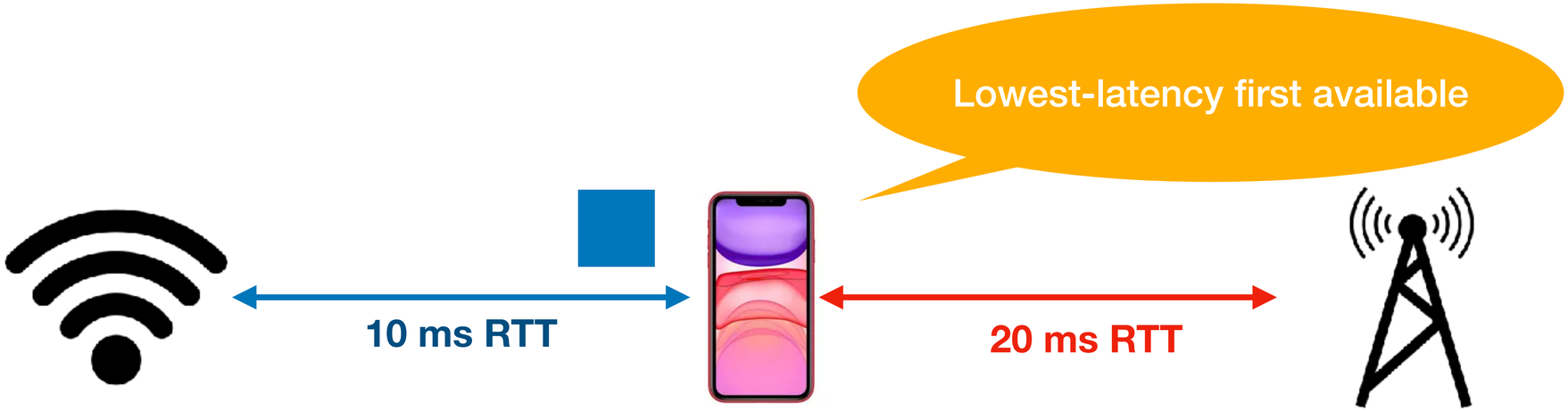
Server via LTE

Client decision

Packet Scheduler

Using Paths

In-kernel implementation

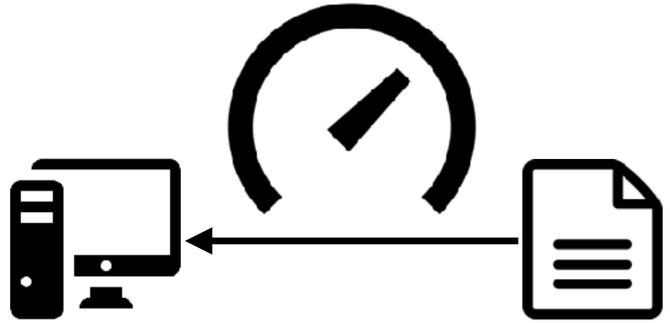


Sender decision

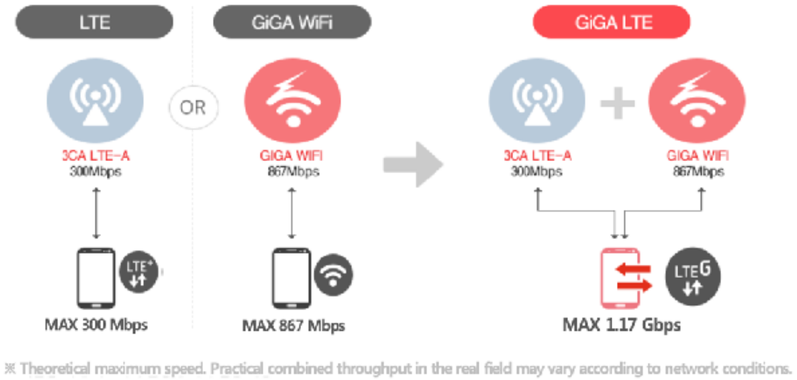
Upload → mostly client decisions

Download → mostly server decisions

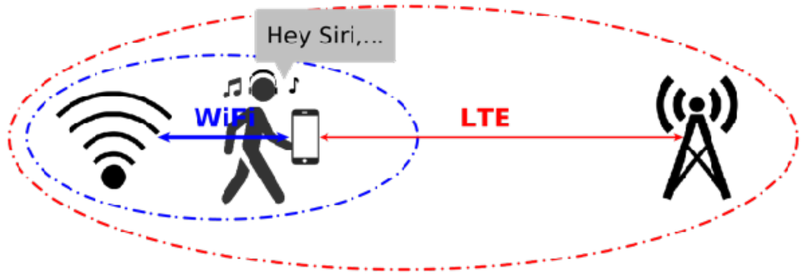
Our Two Basic Applications



- File downloader
  - Bandwidth Aggregation driven
  - E.g., Korean Telecom GiGA LTE

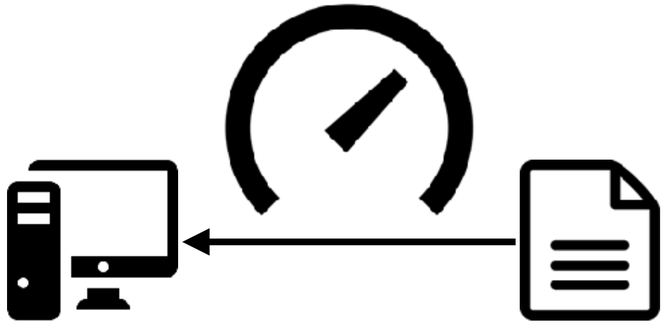


- Interactive Application (under mobile scenario)
  - Request-response based, latency-sensitive
  - E.g., voice-activated applications like Apple Siri



A Multipath-Enabled File Downloader

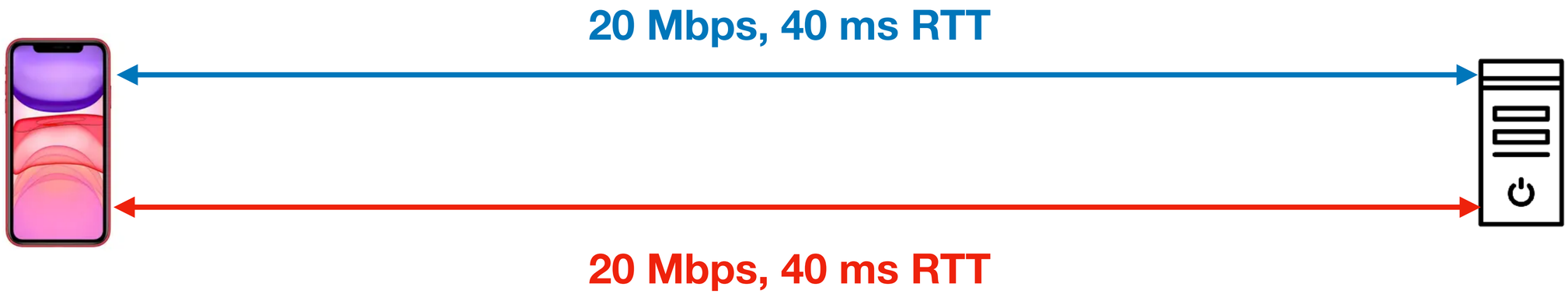
- User expects increased throughput
  - Regardless of file size
  - Regardless of network conditions
- Assume plain HTTP GET request for a file



A Multipath-Enabled File Downloader

Homogeneous case

Fullmesh path manager
Lowest-latency first scheduler



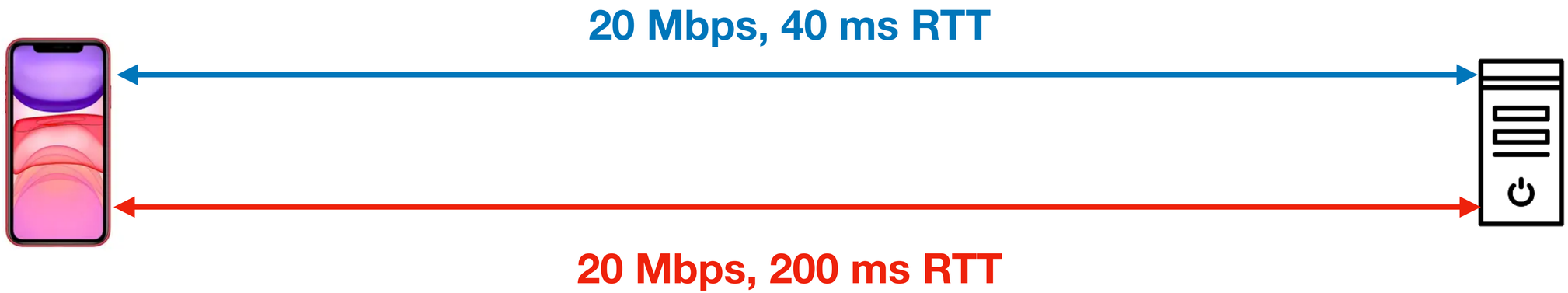
File size	20 MB	512 KB
Download time TCP (only blue path)	8.9 s	390 ms
Download time MPTCP (both paths)	4.65 s	250 ms

~1.9x faster ~1.56x faster 😀

A Multipath-Enabled File Downloader

Heterogeneous case

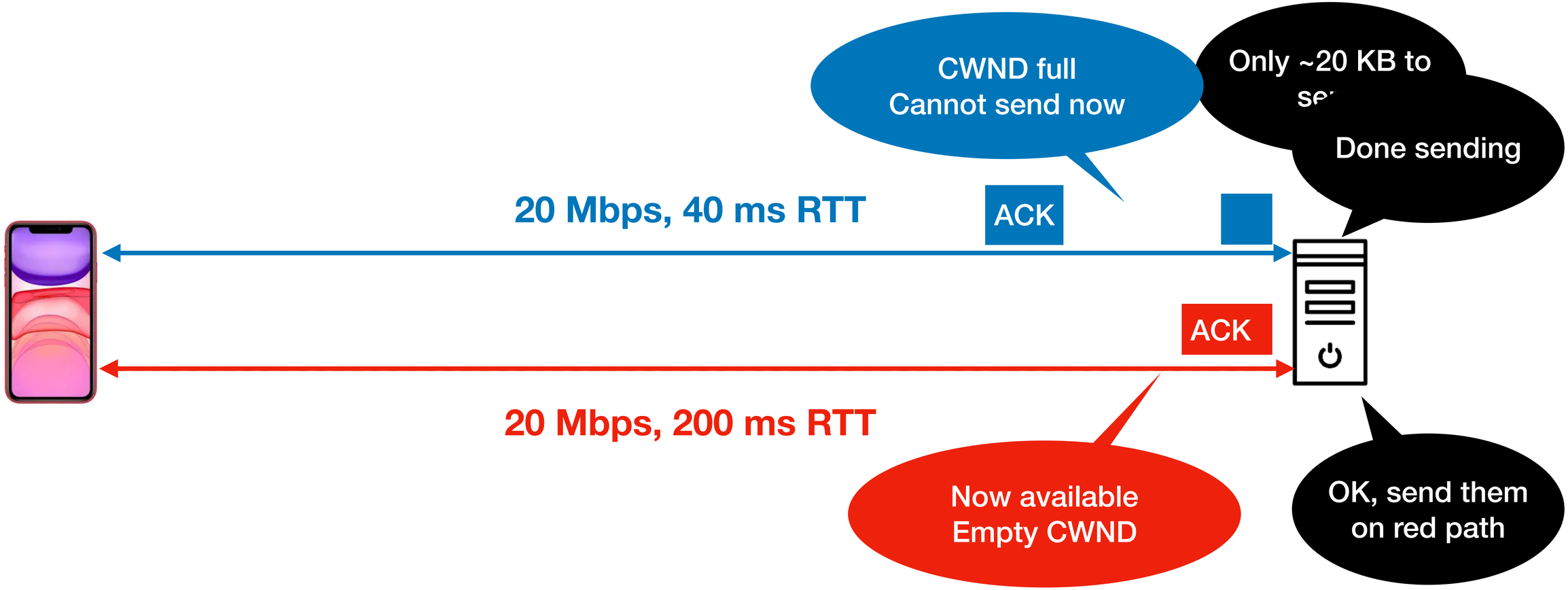
Fullmesh path manager
Lowest-latency first scheduler



File size	20 MB	512 KB
Download time TCP (only blue path)	8.9 s	390 ms
Download time MPTCP (both paths)	5.69 s	480 ms

~~~1.56x faster~~ ~1.25x slower
~~~

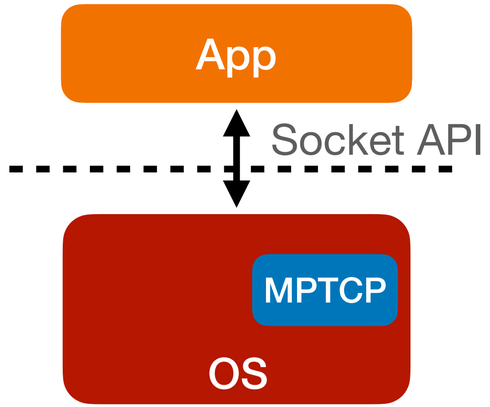
What Happened?



[Networking'16] Ferlin, Simone, et al. "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks." *2016 IFIP networking conference (IFIP networking) and workshops*. IEEE, 2016.
[CoNext'17] Lim, Yeon-sup, et al. "ECF: An MPTCP path scheduler to manage heterogeneous paths." Proceedings of the 13th international conference on emerging networking experiments and technologies. 2017.

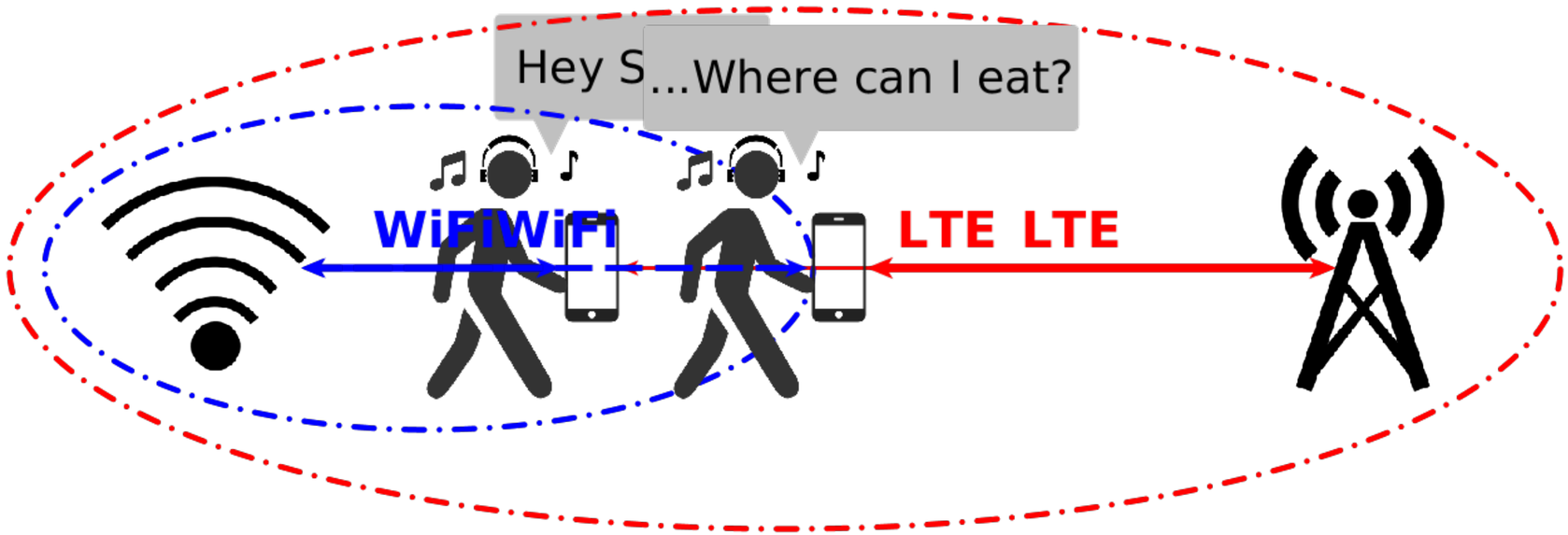
Can We Do Better?

- If known transfer size, avoid using slow path at the end of the transfer
  - But application should give data to the kernel…



- Other heuristics to avoid using slow path if it may block the fast one
  - BLEST [Networking'16], ECF [CoNext'17],…

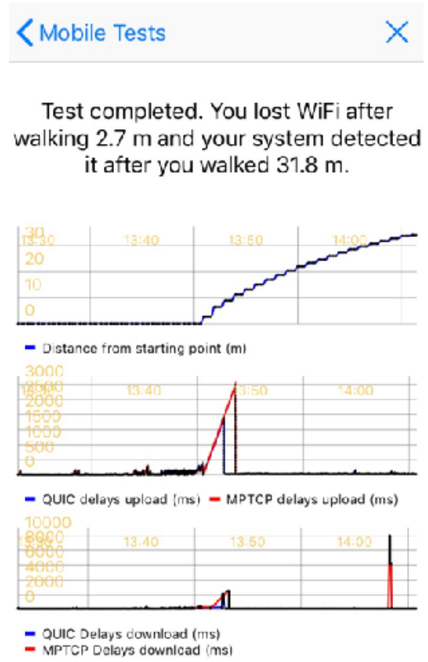
Interactive Applications under Mobility



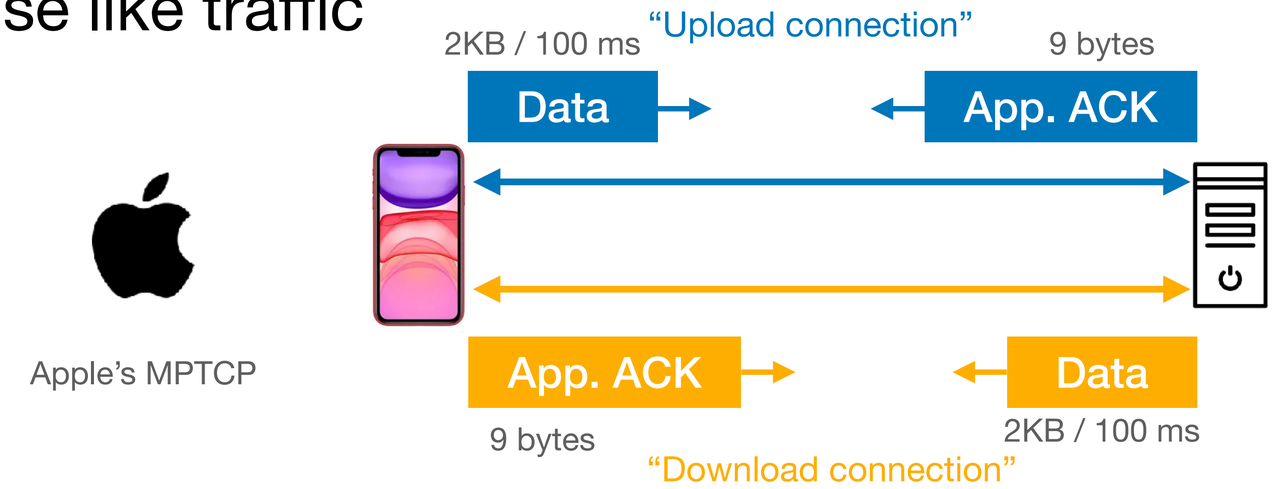
De Coninck, Quentin, and Olivier Bonaventure. "Multipathtester: Comparing mptcp and mpquic in mobile environments." *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2019.

Interactive Application under Mobility

Evaluating Application Perceived Latency

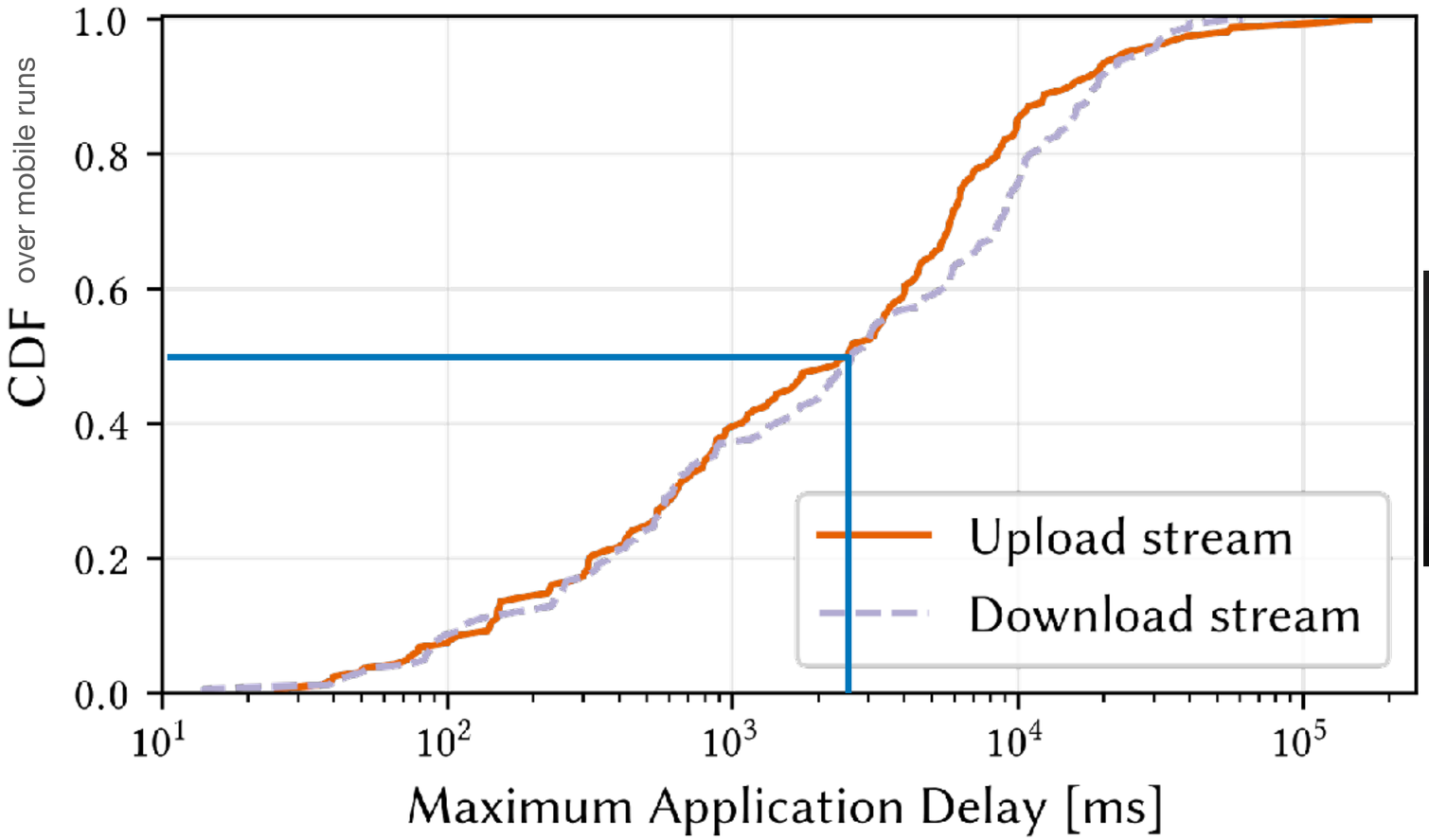


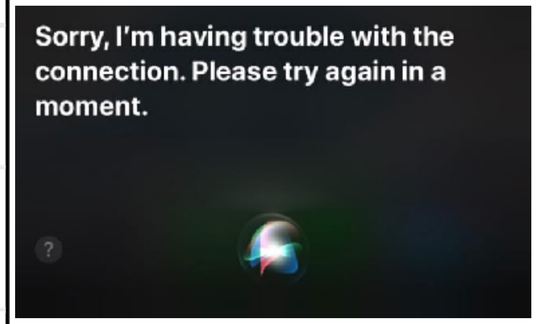
- Active measurements through MultipathTester iOS application
- Run traffic until client device declares Wi-Fi as lost
  - 1100 test runs, +250 unique devices
  - Request-response like traffic



De Coninck, Quentin, and Olivier Bonaventure. "Multipathtester: Comparing mptcp and mpquic in mobile environments." *2019 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 2019.

Applications Perceive Network Handover

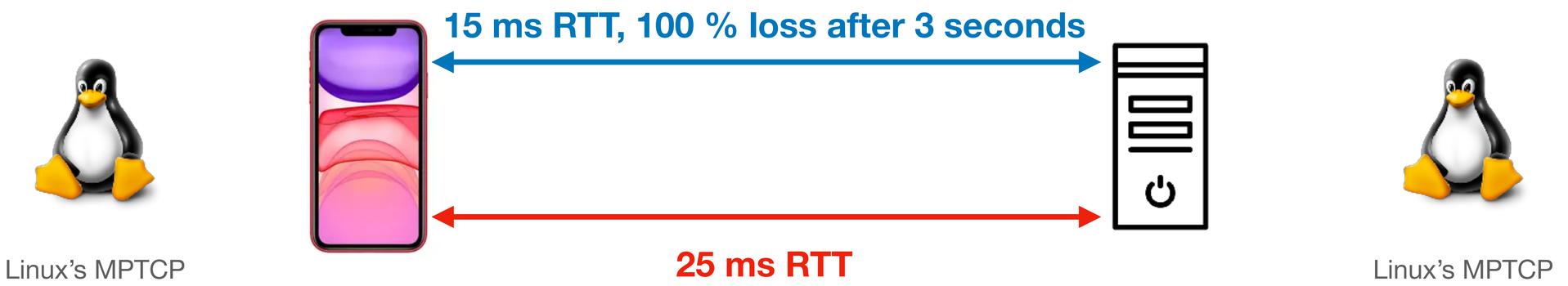




Possible unresponsiveness for several seconds

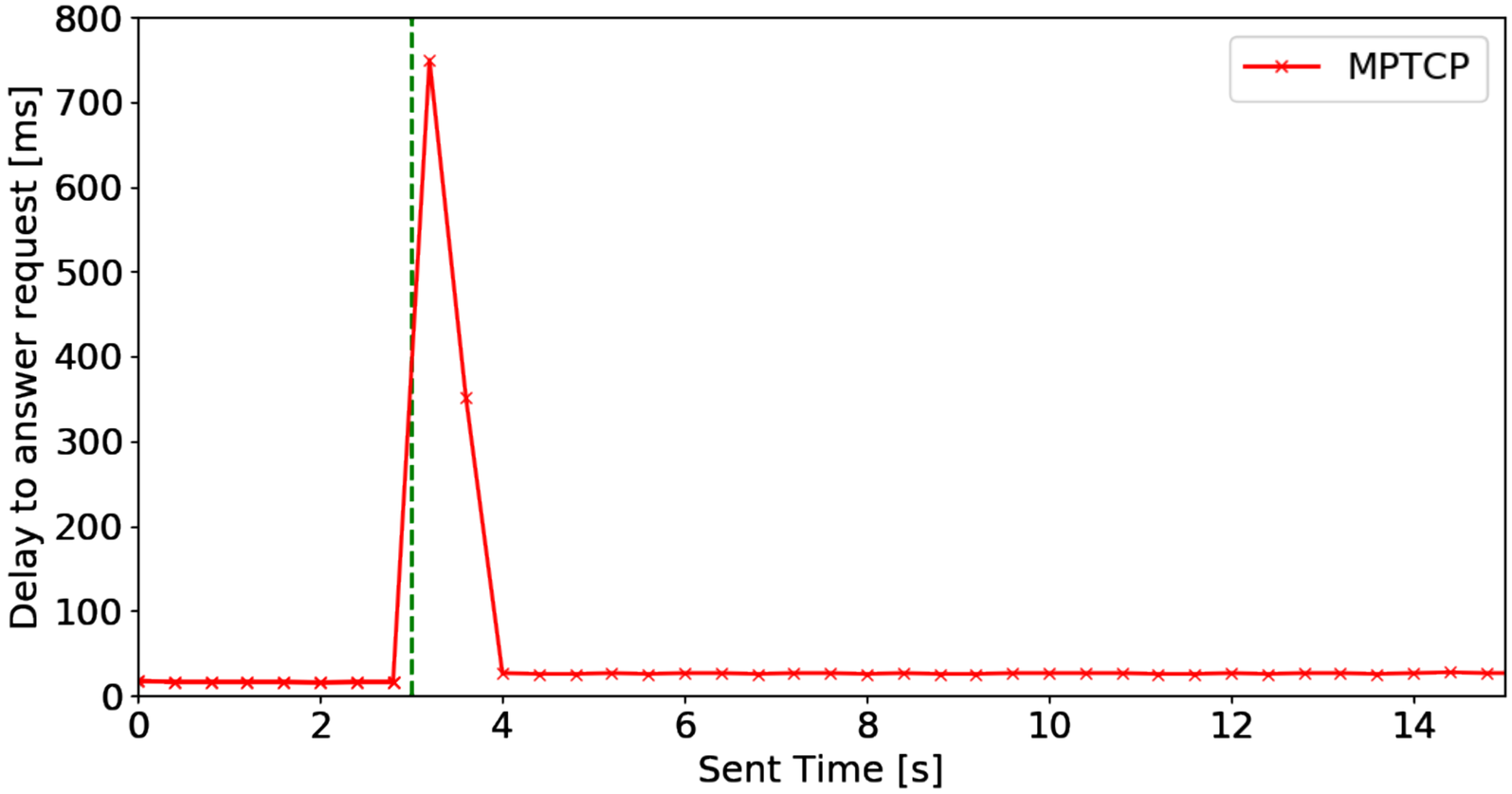
Understanding The Root Cause

- Consider a simpler request/response traffic
  - 750 bytes request/response every 400 ms
  - Measure the delay seen by the client



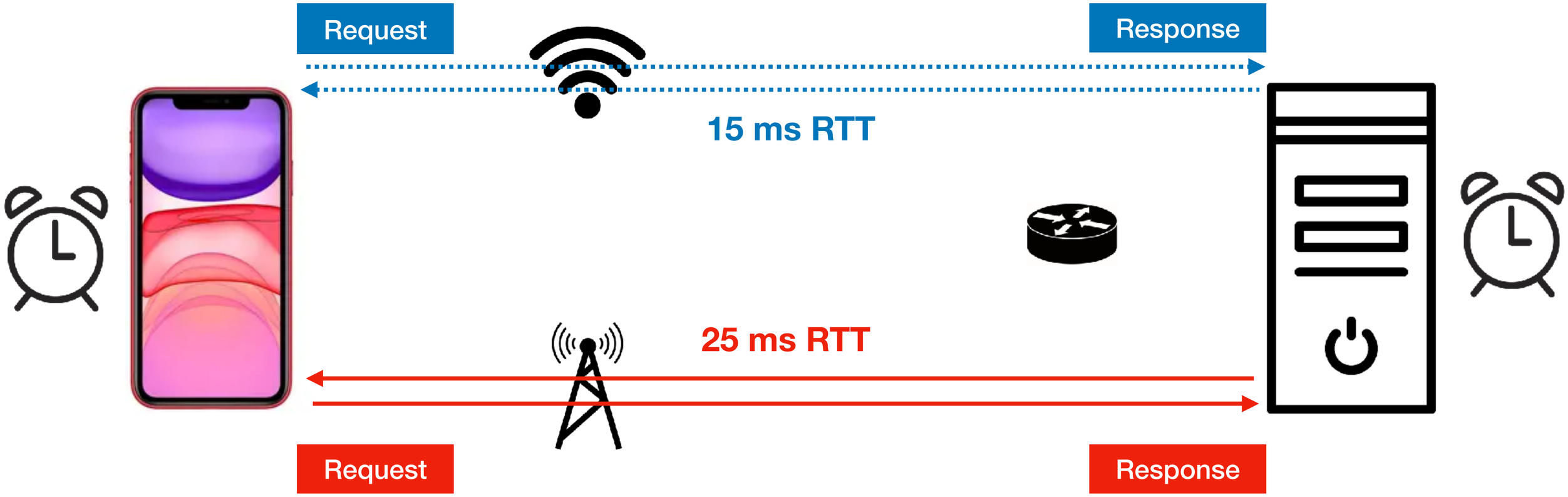
Multipath TCP Handover

Fullmesh path manager
Lowest-latency first scheduler



What Happened Here?

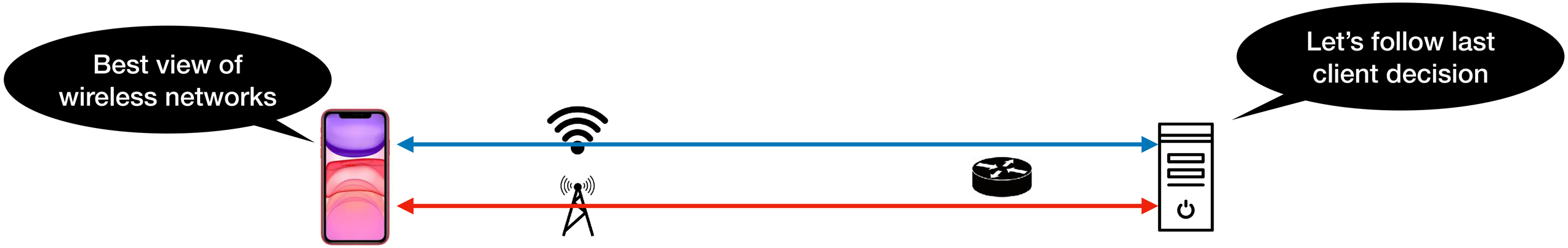
Fullmesh path manager
Lowest-latency first scheduler



Retransmission timeout at both sides

[Networking'18] De Coninck, Quentin, and Olivier Bonaventure. "Tuning multipath TCP for interactive applications on smartphones." *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE, 2018

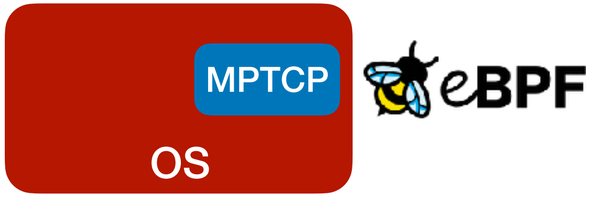
Could We Adapt Multipath Algorithms?



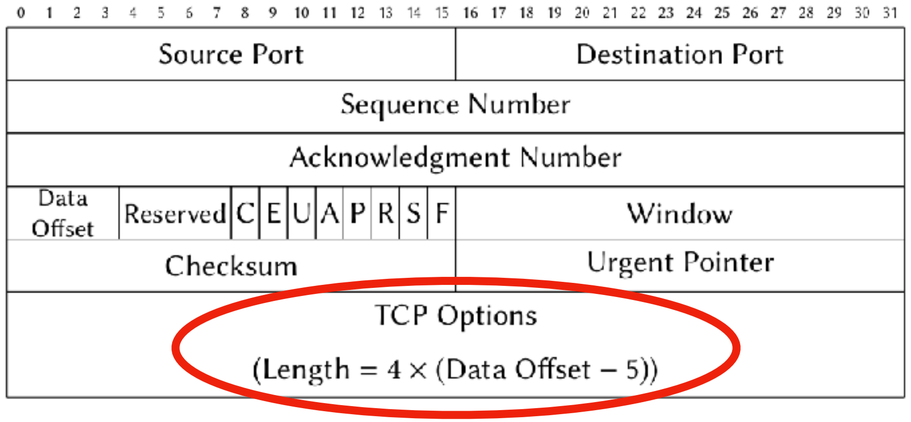
- Server-side scheduler [Networking'18]
  - Remove server-side RTO in mobile scenarios (~350 ms delay vs. 750 ms)
  - Relevancy for the file downloader case?

An Application-Tailored Multipath TCP?

- Deployment issues



- TCP Limitations

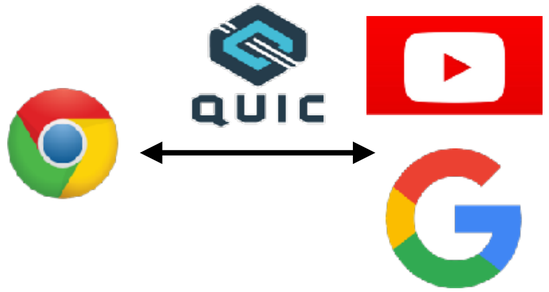


Max 40 bytes

The QUIC Transport Protocol

The QUIC Protocol

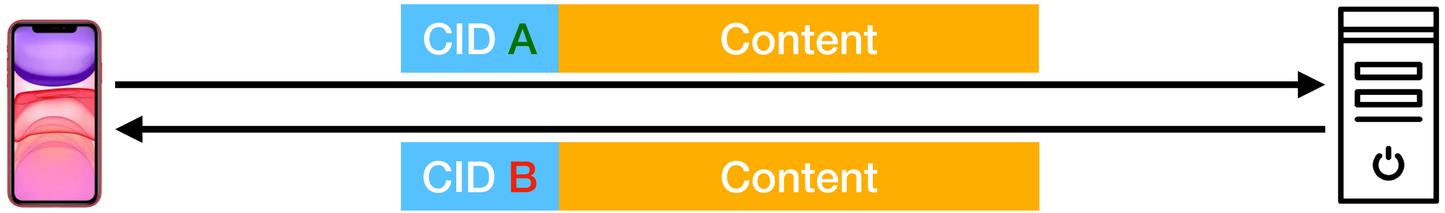
Since 2014



- TCP / TLS 1.3 replacement atop UDP

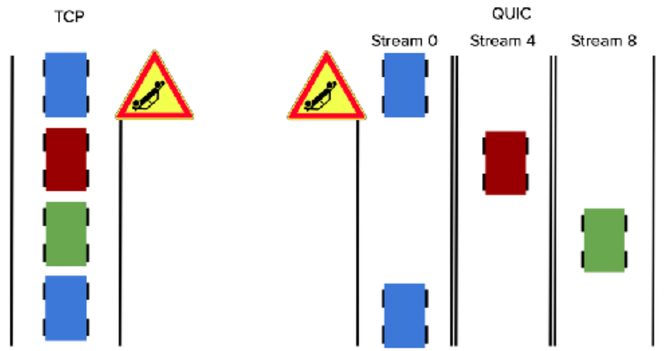
Reliable Encrypted Unreliable

- Unidirectional Connection IDs (CIDs)



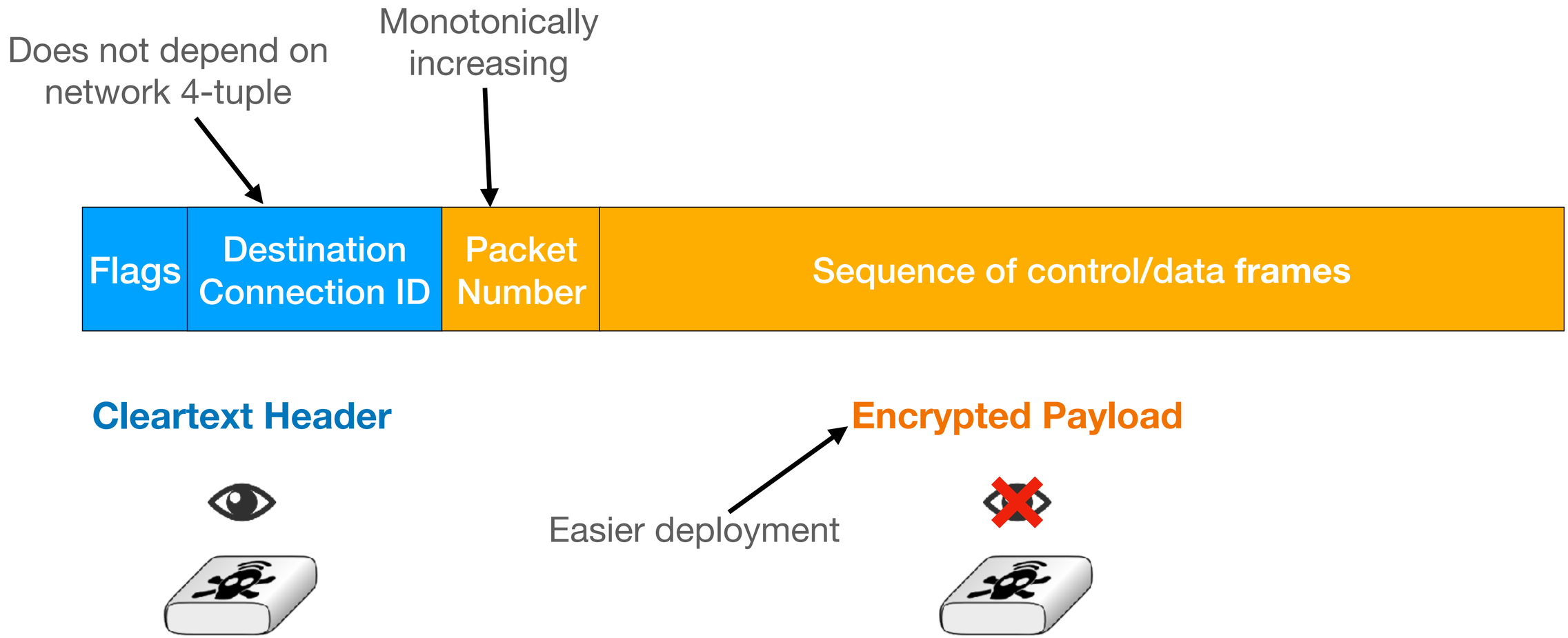
Not bound to a 4-tuple anymore

- Multistreaming support



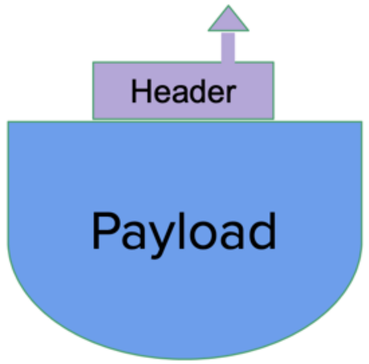
- 0-RTT connection establishment (most of the time)
- Extension negotiation during handshake

A QUIC Packet

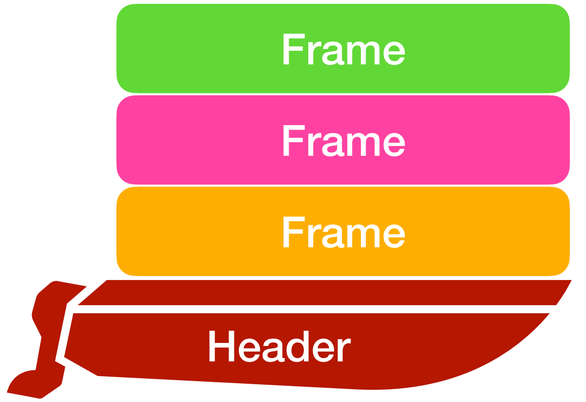


A QUIC Packet

Seen by the Sea



TCP



QUIC

A QUIC Packet

Does not depend on network 4-tuple

Monotonically increasing for efficient recovery

Control frame acknowledging packets

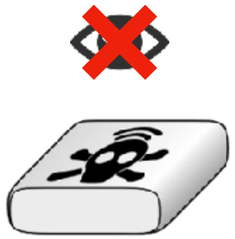
Control frame providing alternate CIDs

Data multiplexing over a single connection

Cleartext Header

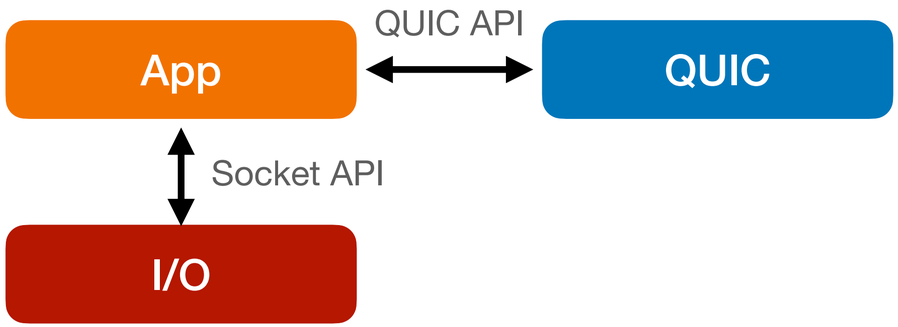
Encrypted Payload

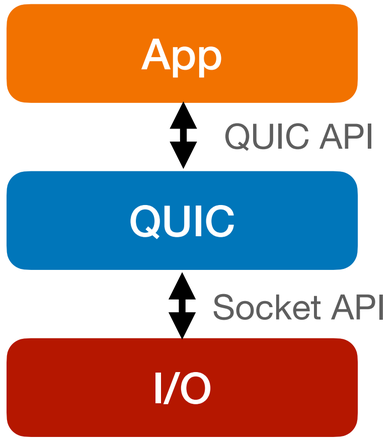
Easier deployment



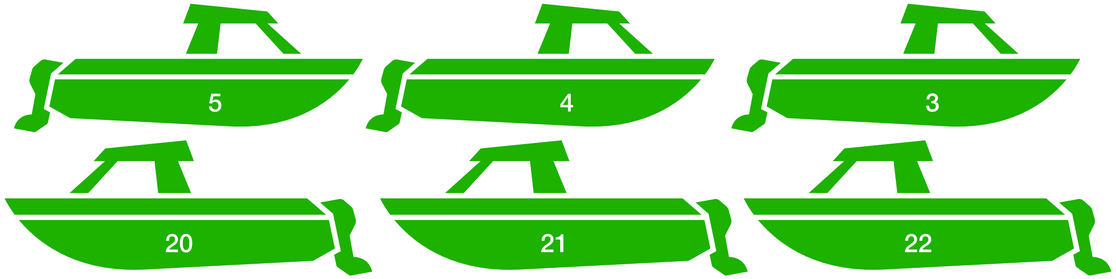
QUIC Implementations

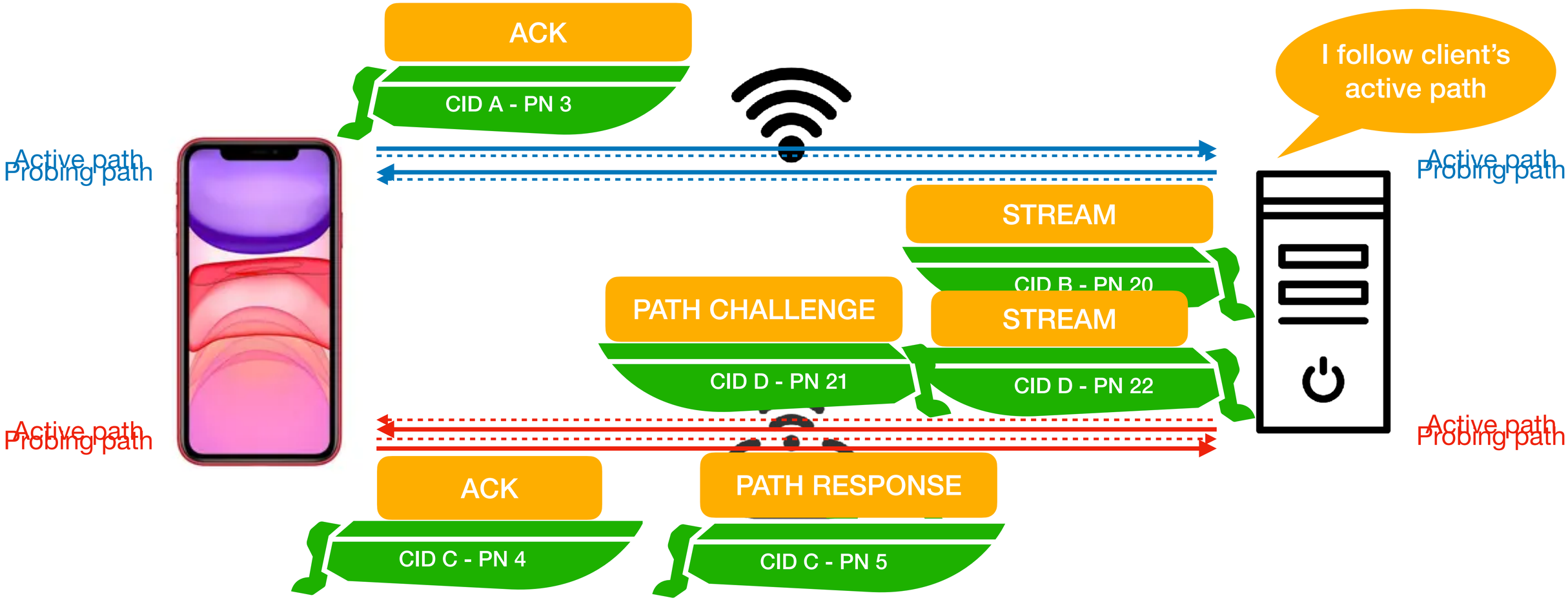
- +20 publicly known implementations
  - Meta’s mvfst, Microsoft’s MSQUIC, AppleQUIC, Cloudflare’s quiche,…
- User-space implementations
  - Can be easily integrated with applications
    - But apps should be adapted





Connection Migration





Limitation: only one "data path" at a time

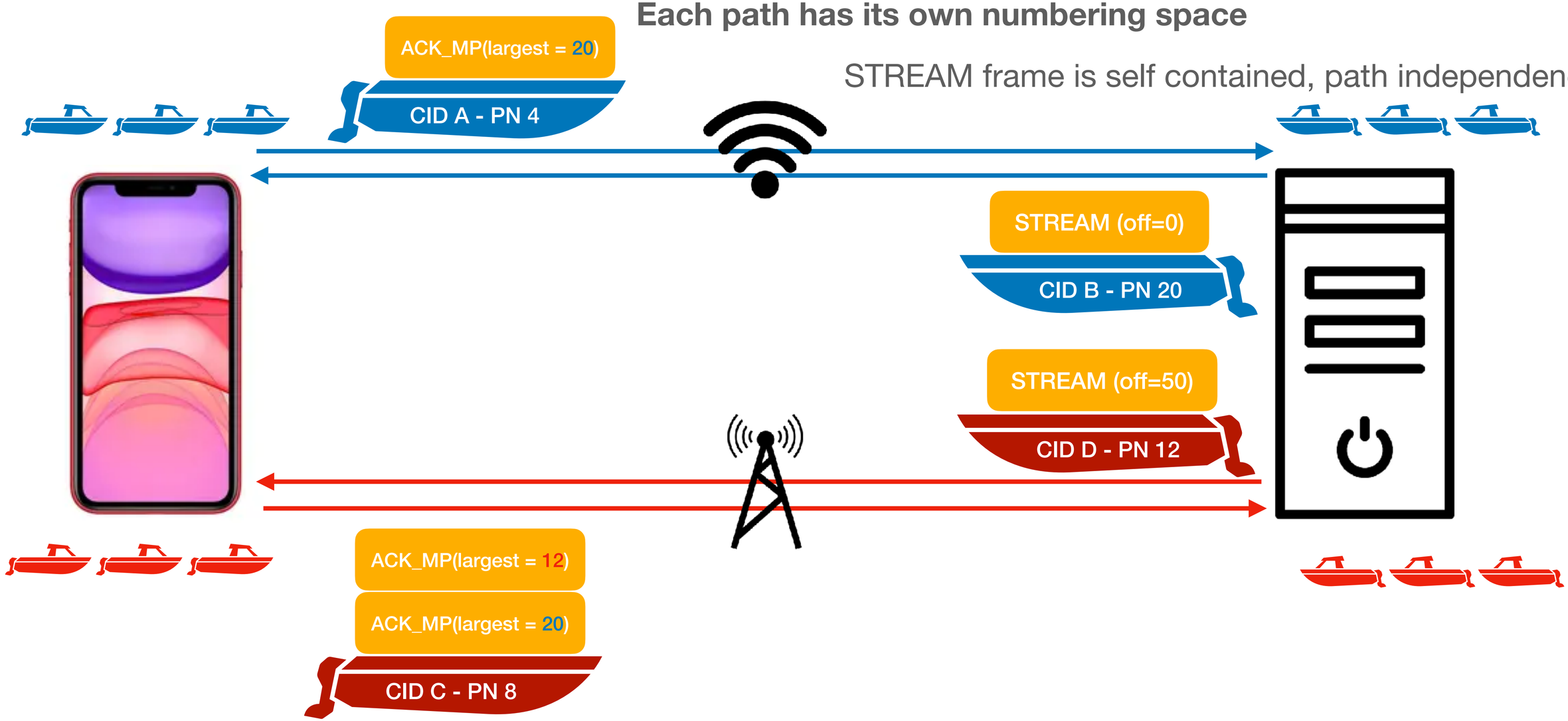
Bringing Multipath to QUIC

Multipath QUIC

Connection IDs provided through NEW_CONNECTION_ID frames

Path identification using Connection IDs
Each path has its own numbering space

STREAM frame is self contained, path independence

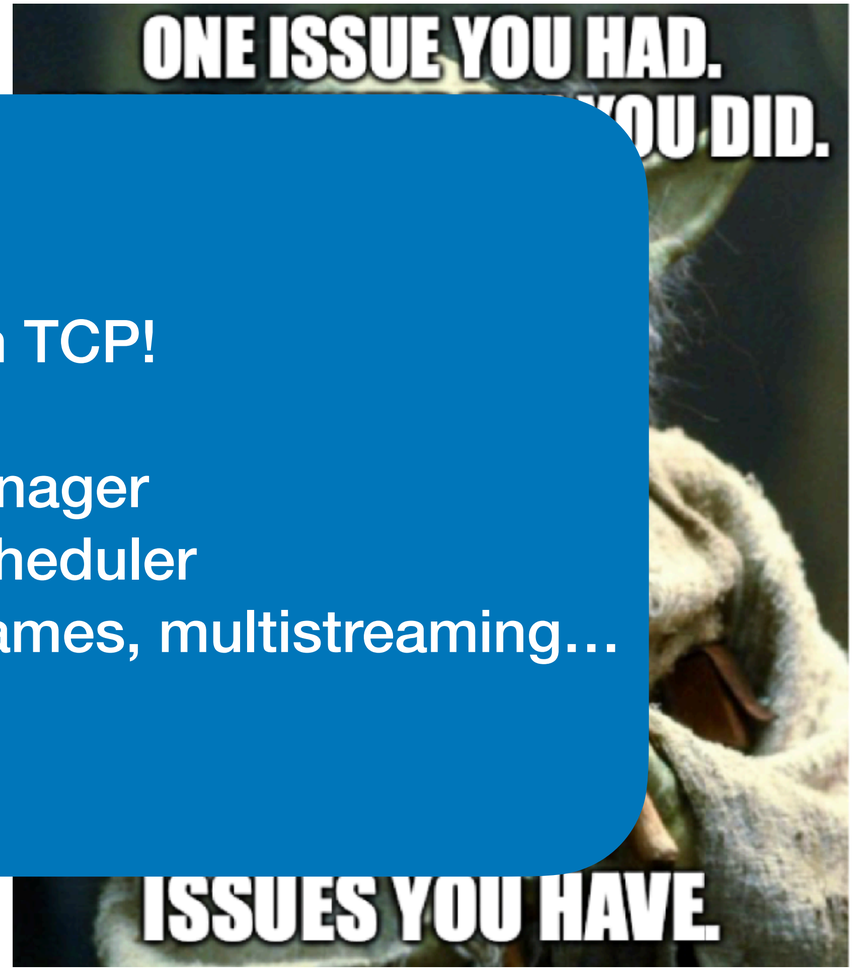


Multipath Algorithms

- Path man
  - Creatin
- Packet sc
  - Using p
- (Congesti
  - Per-pat

Same as Multipath TCP!

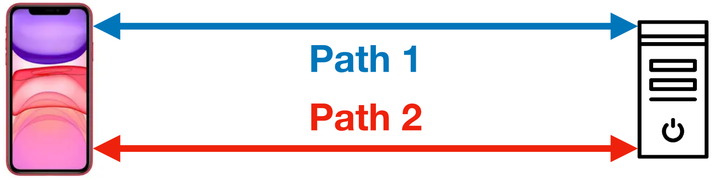
Fullmesh path manager
Lowest-RTT first scheduler
/!\ Also need to schedule control frames, multistreaming…

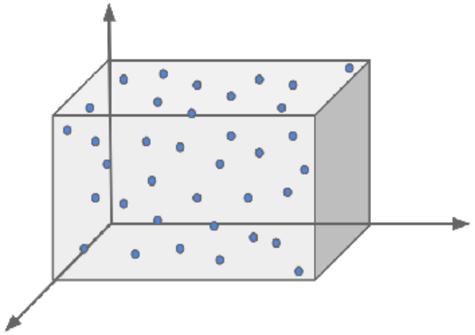


Revisiting the File Downloader

- Download 20 MB on single stream
  - 9 runs, take median download completion time
  - Compare Linux (Multipath) QUIC with (Multipath) QUIC

Mininet setup





Factor	Minimum	Maximum
Capacity [Mbps]	1	50
RTT [ms]	0	50
Queuing Delay [ms]	0	100
Random Loss [%]	0	2.5

253 network scenarios
Vary initial path

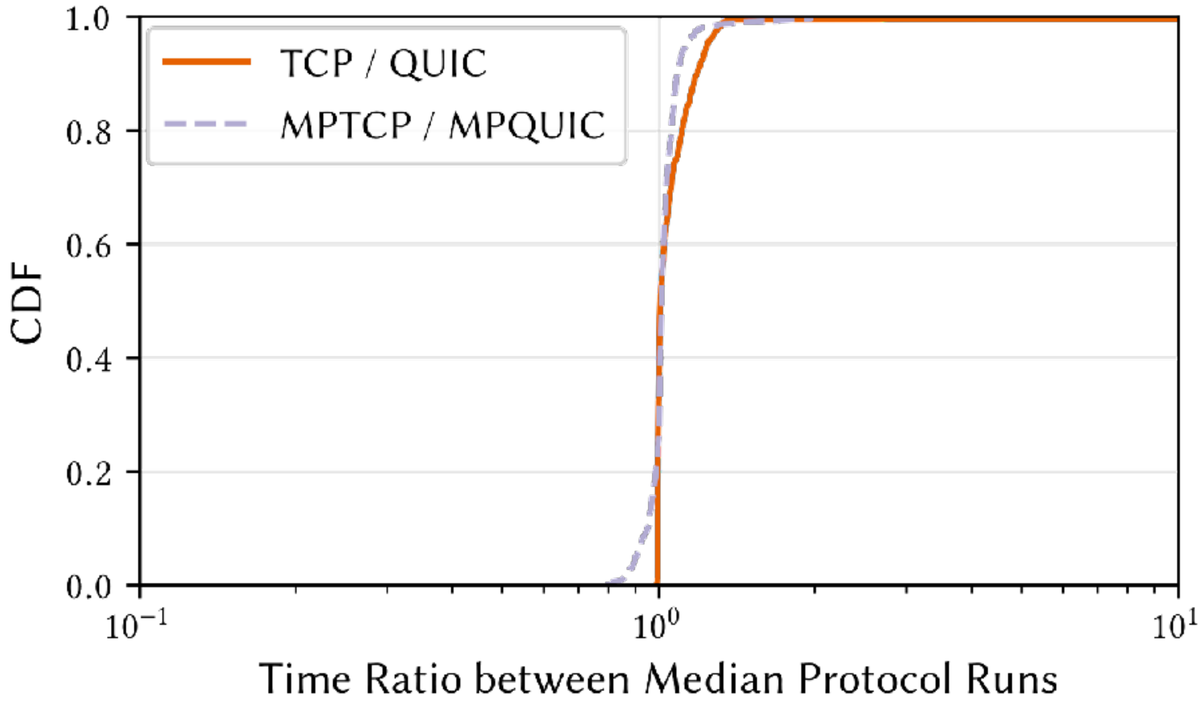
De Coninck, Quentin, and Olivier Bonaventure. "Multipath quic: Design and evaluation." *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 2017.

Fullmesh path manager
Lowest-latency first scheduler

Speedup Ratio of QUIC vs. TCP

Completion time (MP)TCP / Completion Time (MP)QUIC

No random loss case



Same algorithms, same results

Hanshake slightly faster in QUIC

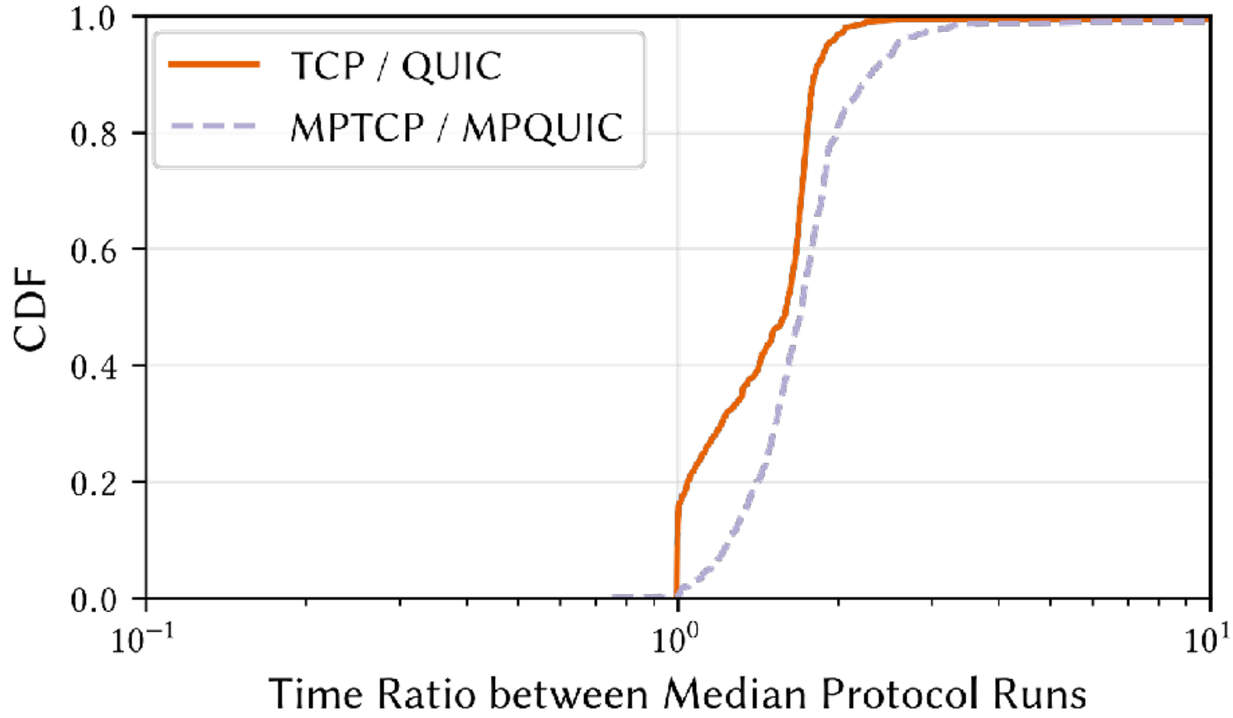
De Coninck, Quentin, and Olivier Bonaventure. "Multipath quic: Design and evaluation." *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 2017.

Fullmesh path manager
Lowest-latency first scheduler

Speedup Ratio of QUIC vs. TCP

Completion time (MP)TCP / Completion Time (MP)QUIC

With random losses

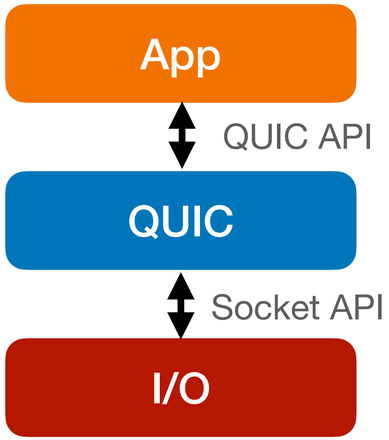


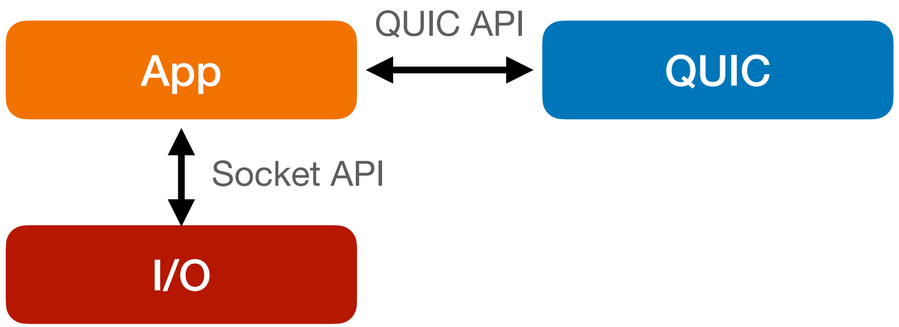
- QUIC ACK range (>400 entries) >>> TCP SACK range (2-3 due to TCP options)
- QUIC does not need to (re)transmit original packet if delivered on another path

De Coninck, Quentin, and Olivier Bonaventure. "Multipath quic: Design and evaluation." *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 2017.

What About Short Files?

- If a slow path is used at the end of a transfer, still have the issue
  - But application can more easily integrate/tweak the MPQUIC scheduler
    - If transfer size is known, avoid using the slow path when not advantageous

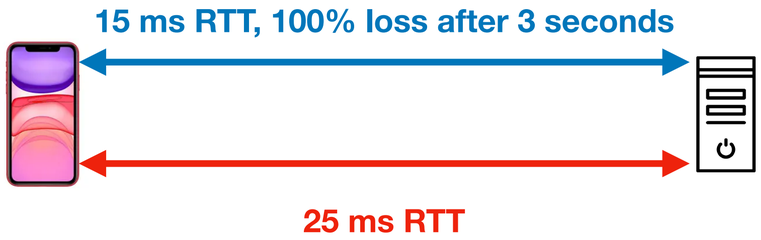




De Coninck, Quentin, and Olivier Bonaventure. "Multipath quic: Design and evaluation." *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 2017.

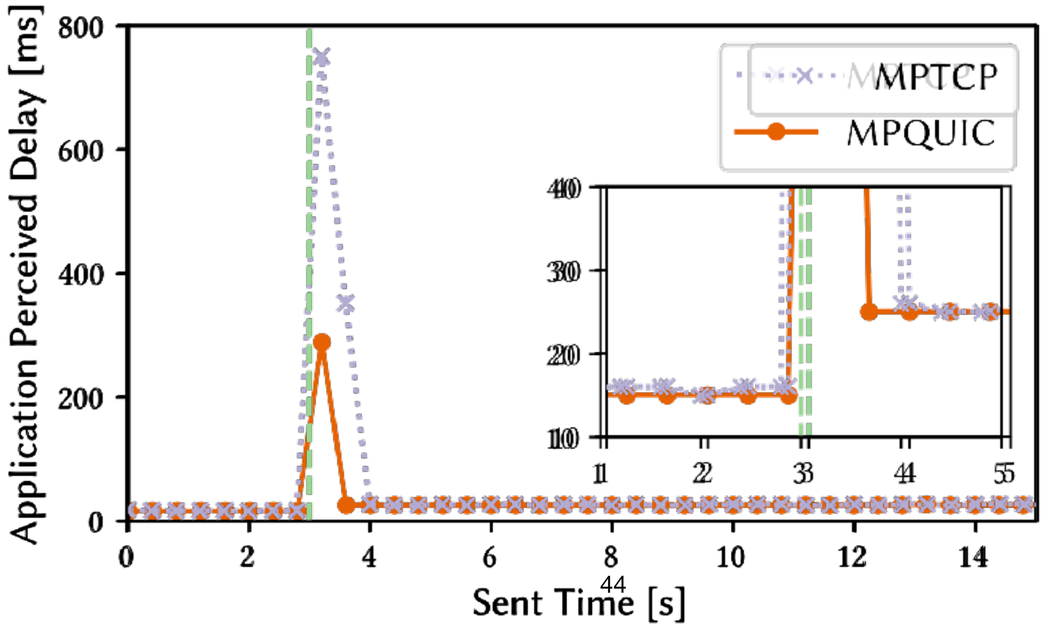
Revisiting the Interactive Traffic with Mobility

With Multipath QUIC



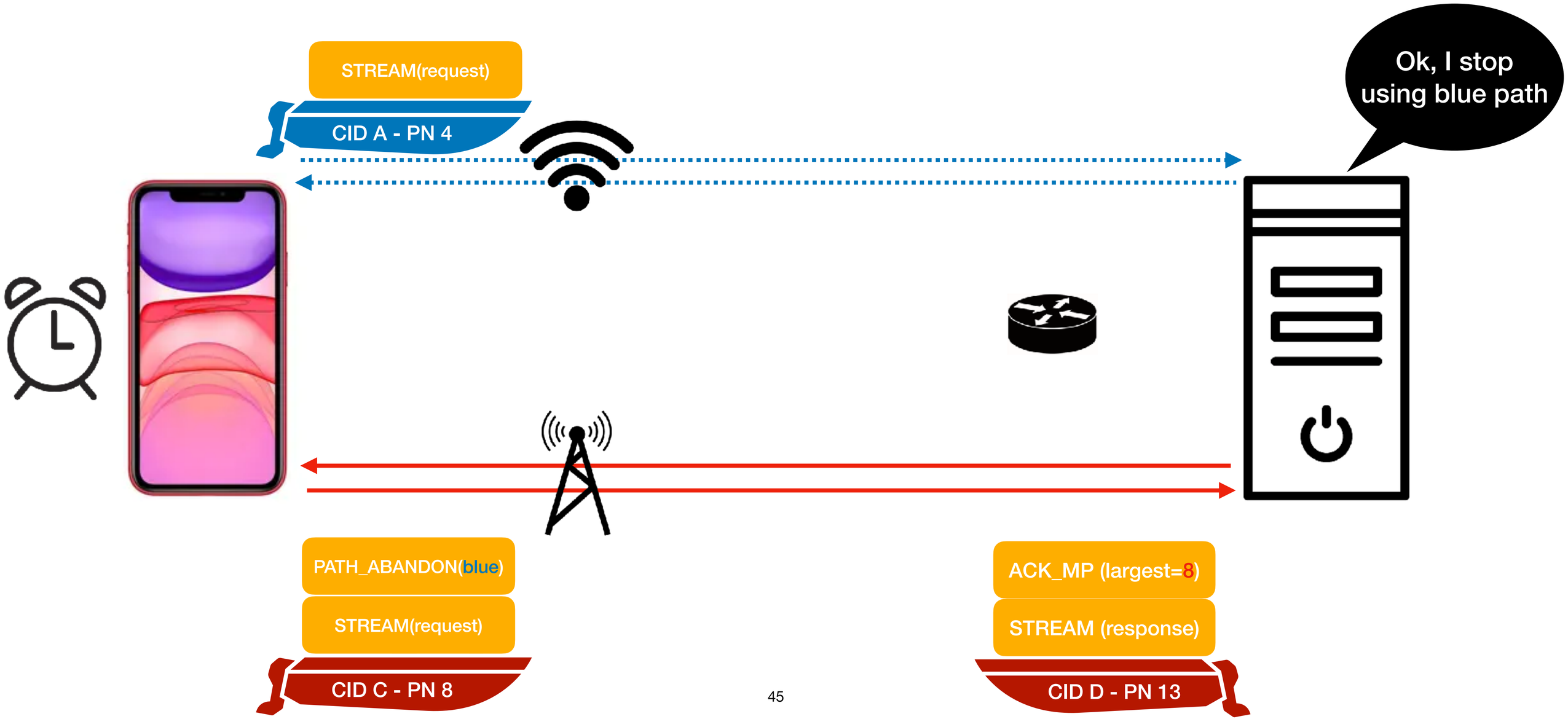
Request/Response traffic

- 750 bytes request/response every 400 ms
- Measure application delay seen by client



Understanding the Difference

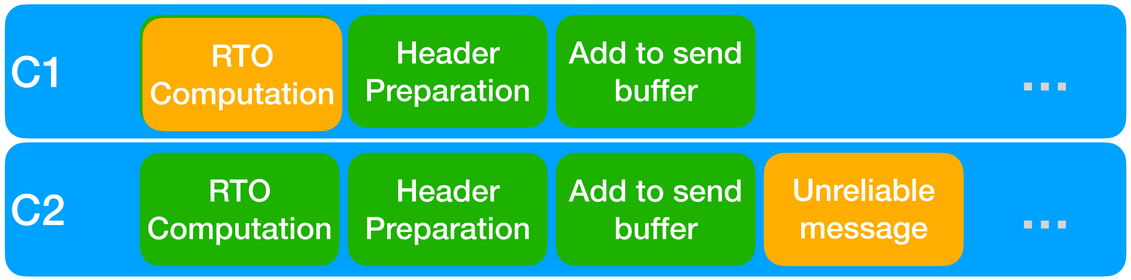
Fullmesh path manager
Lowest-latency first scheduler

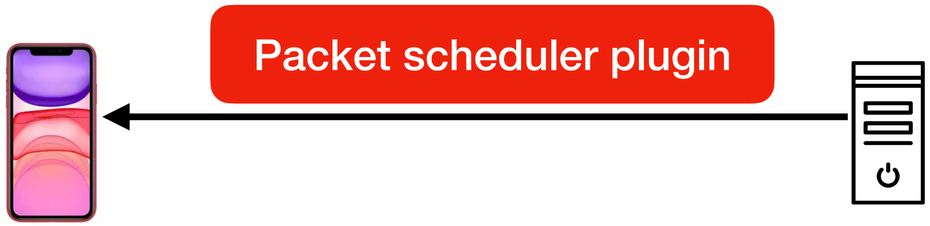


De Coninck, Quentin, et al. "Pluginizing quic." *Proceedings of the ACM Special Interest Group on Data Communication*. 2019. 59-74.

Supporting Different Use Cases?

- Closer application integration ≠ immediate deployment
- What about letting applications tweak their network stack on a per-connection basis?
- Pluginize your protocol (e.g., QUIC)





And What About Video Streaming?

What About Video Streaming?

- Application has both aggregation bandwidth + latency/buffer constraints
  - Especially (near-)live streaming

[CoNext'16] Han, Bo, et al. "MP-DASH: Adaptive video streaming over preference-aware multipath." *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. 2016.

Where Multipath Transport Can Help

- Audio streaming
  - Handle mobile cases with multiple paths
- Pre-chuncked videos (e.g., DASH)
  - Higher video quality through adaptive rate algorithms [CoNext'16]
  - Resilience to network quality changes

[Networking'19] Michel, François, Quentin De Coninck, and Olivier Bonaventure. "QUIC-FEC: Bringing the benefits of Forward Erasure Correction to QUIC." *2019 IFIP Networking Conference (IFIP Networking). IEEE, 2019.*

Where Multipath Transport Can Help

- Multipath-aware video coding
  - I-frames delivery more important than B- and P-ones
  - Multistreaming of (Multipath) QUIC can help here
  - Doing QUIC Forward Erasure Correction [Networking'19] on side paths

Conclusion

IETF 115 - November 7th, 2022

- Multipath transports now available (more than 10 years of efforts)
- Good application performance requires algorithms adaptations
  - Bandwidth aggregation goal ≠ lowest-latency goal
- Video streaming: identify your goals & implement algorithms
  - Multipath QUIC more powerful/deployable than Multipath TCP
    - At least 3 open-source implementations (Cloudflare’s quiche, picoquic,…)

quentin.deconinck@uclouvain.be