Multipath QUIC: Design and Evaluation

Quentin De Coninck, Olivier Bonaventure
quentin.deconinck@uclouvain.be

multipath-quic.org

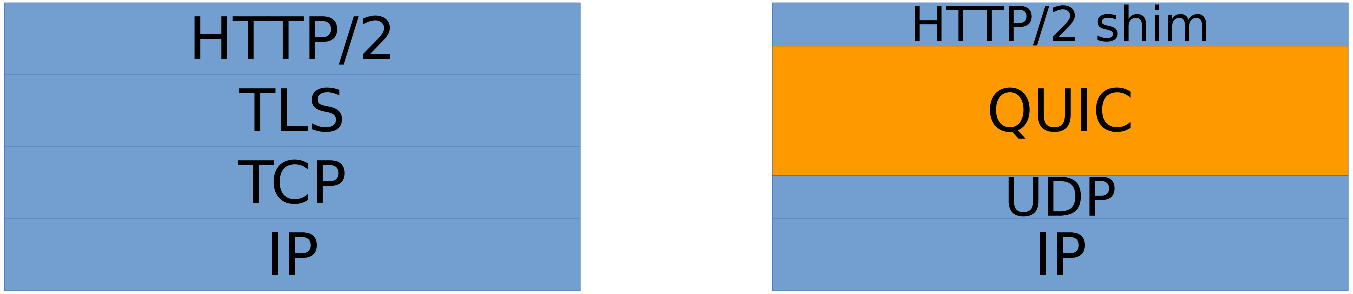
Outline

- The QUIC protocol
- Designing Multipath for QUIC
- Experimental Design Evaluation
- Ongoing Work and Conclusion

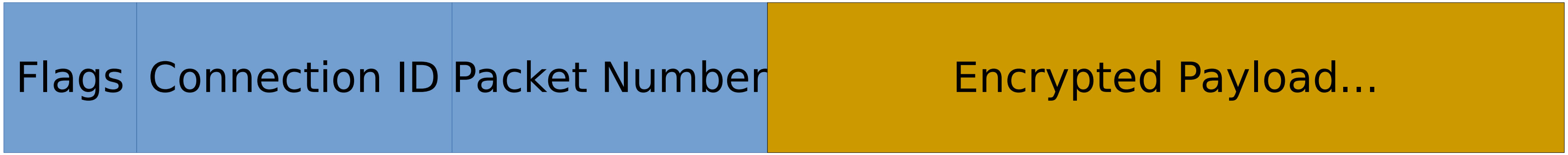
QUIC: watisda?

QUIC = Quick UDP Internet Connection

- **TCP/TLS1.3 atop UDP**
  - >7% of the Internet traffic (YouTube, Chrome,...)
- **Stream multiplexing → HTTP/2 use case**
- **0-RTT establishment (most of the time)**



QUIC Packet



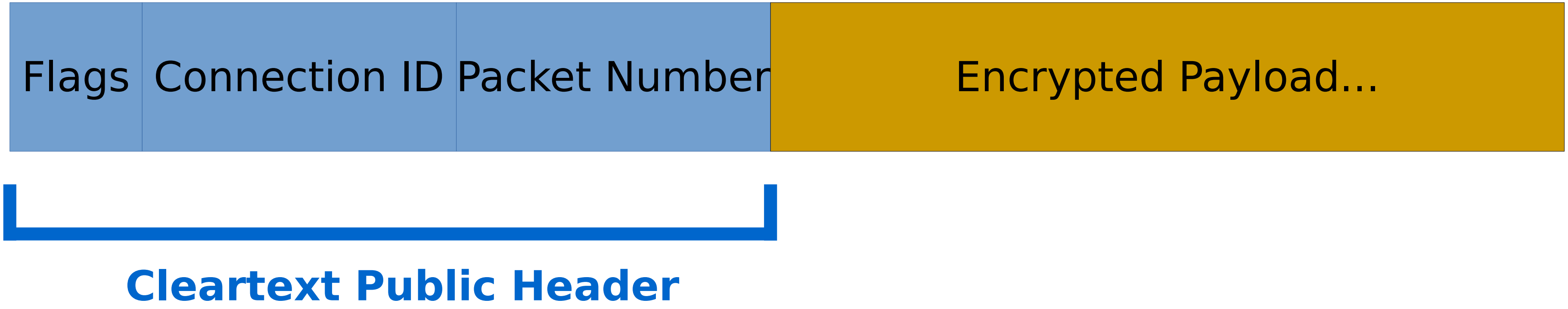
Flags

Connection ID

Packet Number

Encrypted Payload...

QUIC Packet



Flags

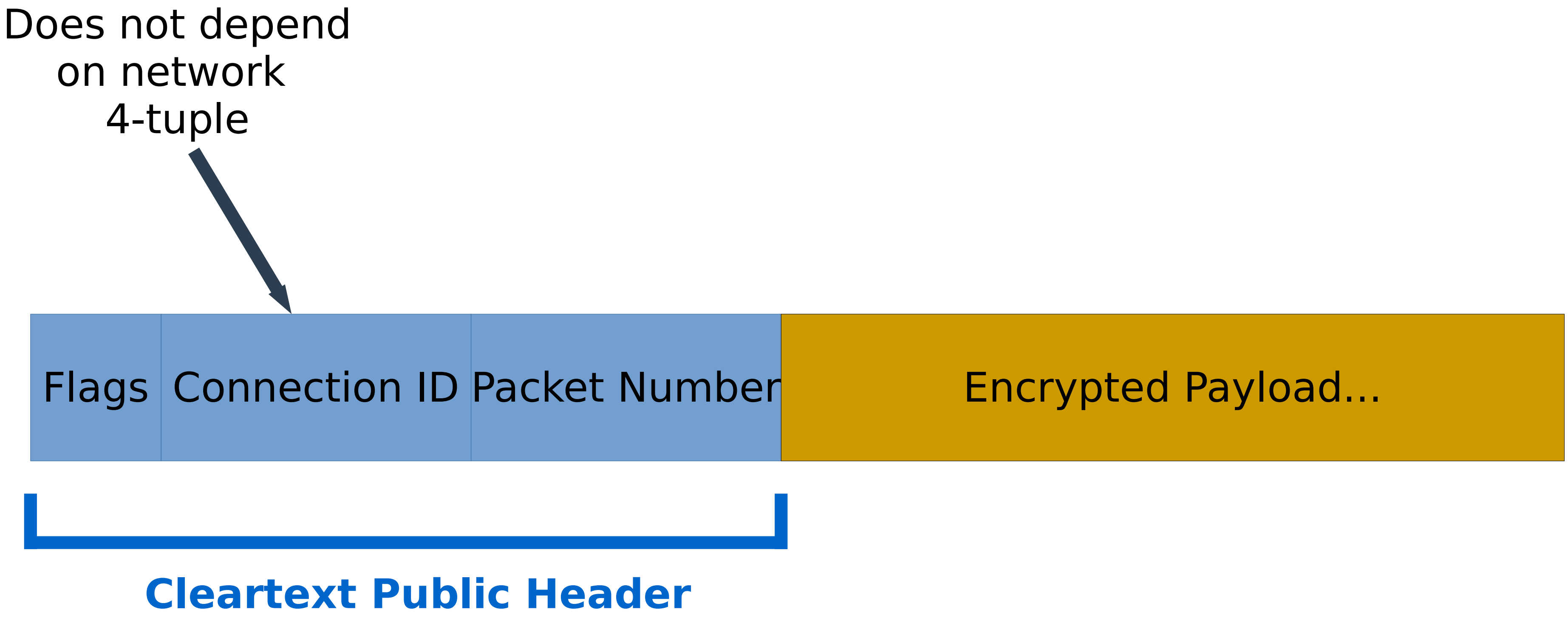
Connection ID

Packet Number

Encrypted Payload...

Cleartext Public Header

QUIC Packet

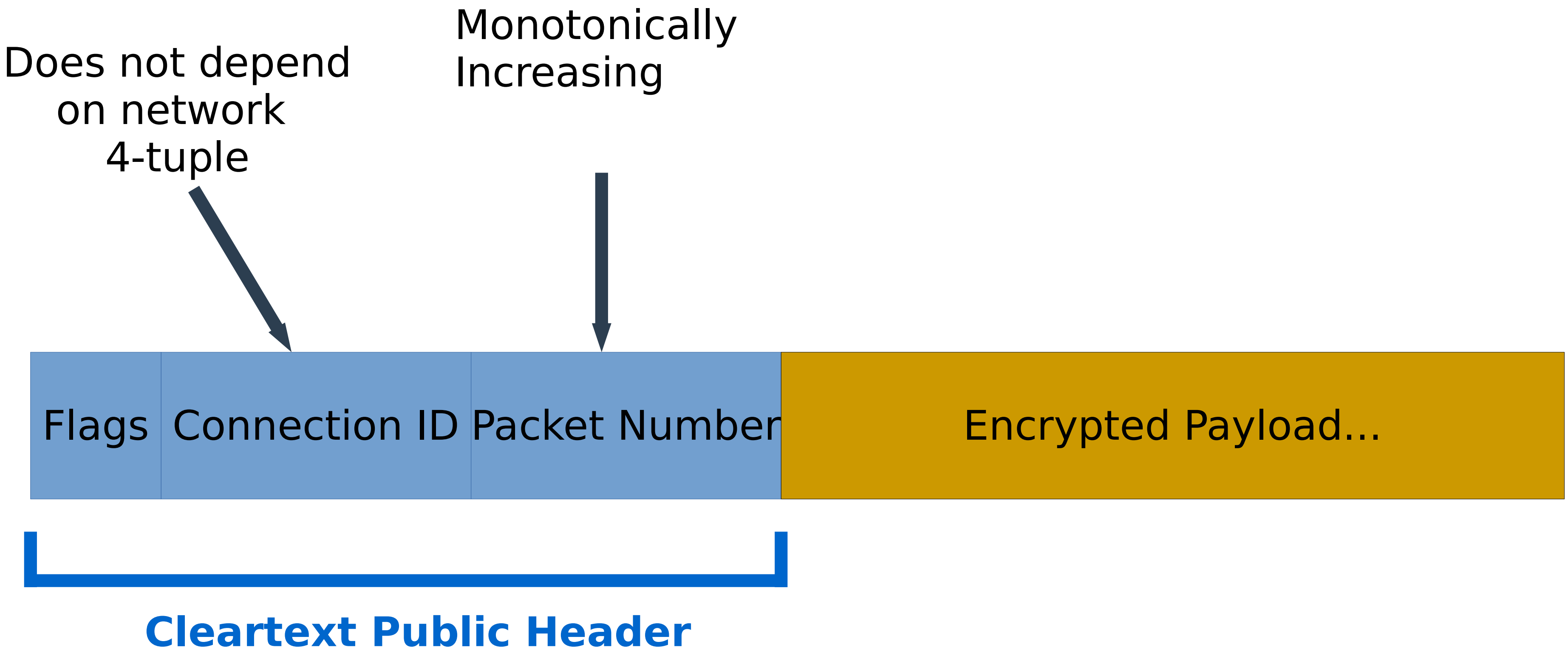


Does not depend on network 4-tuple

Flags | Connection ID | Packet Number | Encrypted Payload...

Cleartext Public Header

QUIC Packet



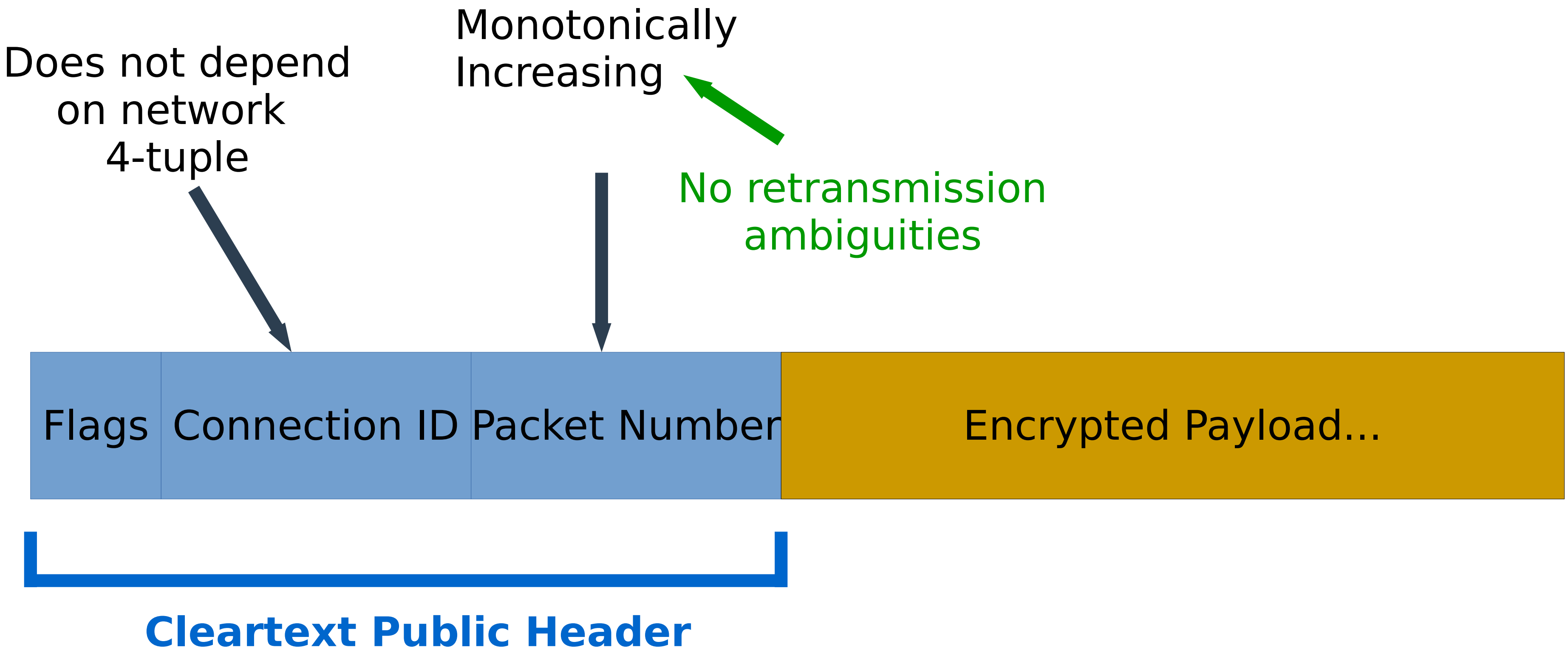
Does not depend on network 4-tuple

Monotonically Increasing

Flags | Connection ID | Packet Number | Encrypted Payload...

Cleartext Public Header

QUIC Packet



Does not depend on network 4-tuple

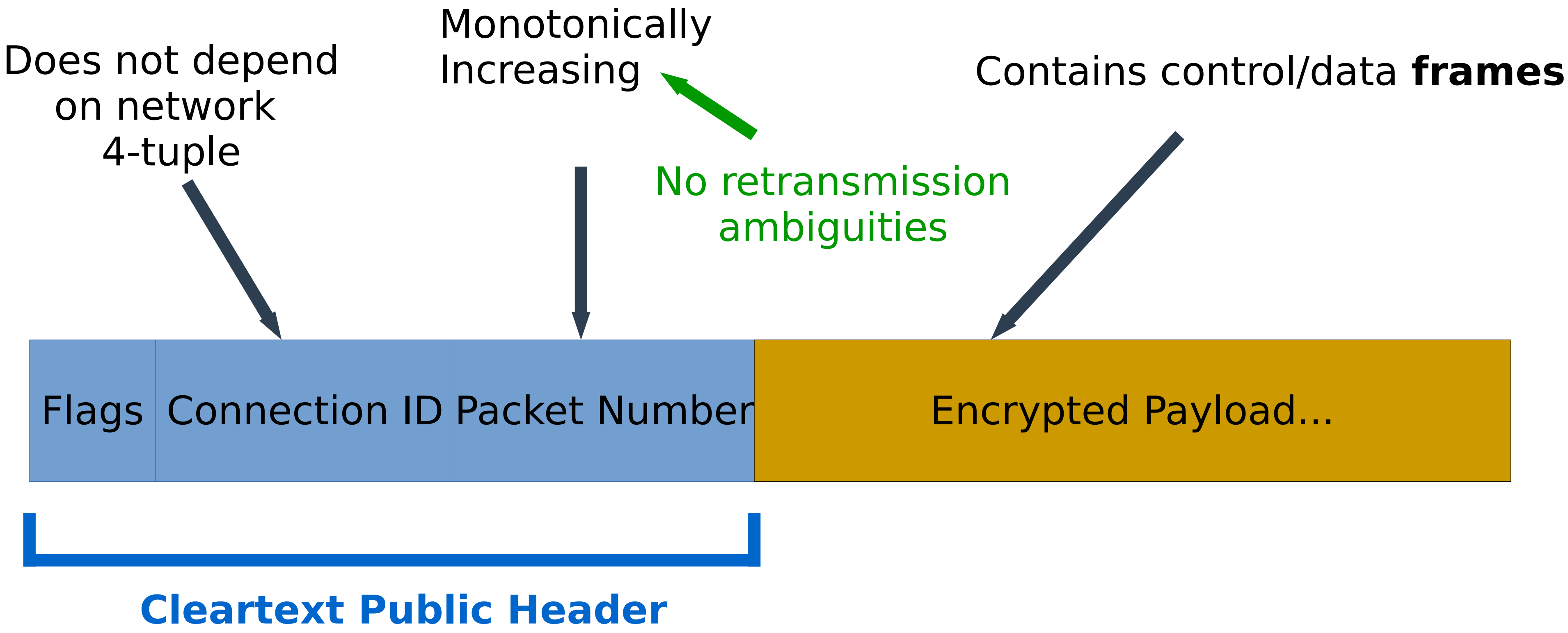
Monotonically Increasing

No retransmission ambiguities

Flags	Connection ID	Packet Number	Encrypted Payload...

Cleartext Public Header

QUIC Packet



Does not depend on network 4-tuple

Monotonically Increasing

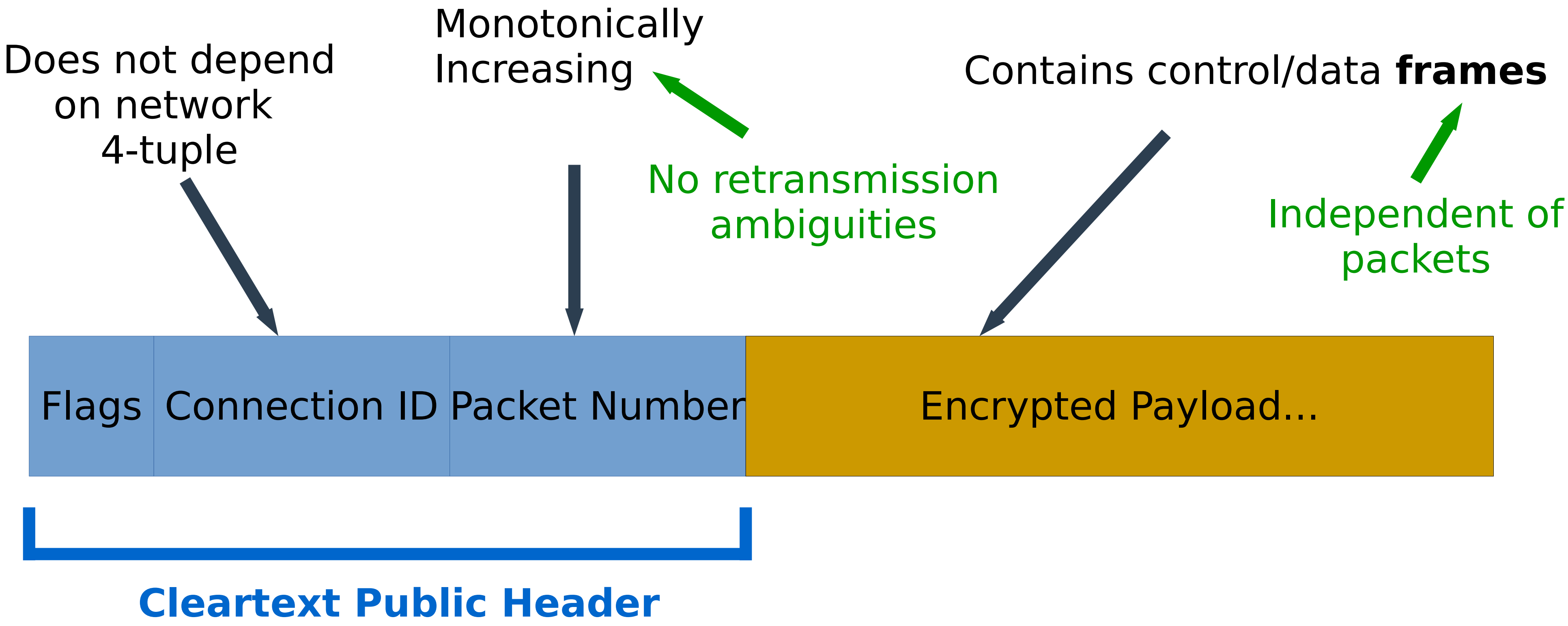
No retransmission ambiguities

Contains control/data **frames**

Flags | Connection ID | Packet Number | Encrypted Payload...

Cleartext Public Header

QUIC Packet



Does not depend on network 4-tuple

Monotonically Increasing

No retransmission ambiguities

Contains control/data **frames**

Independent of packets

Flags

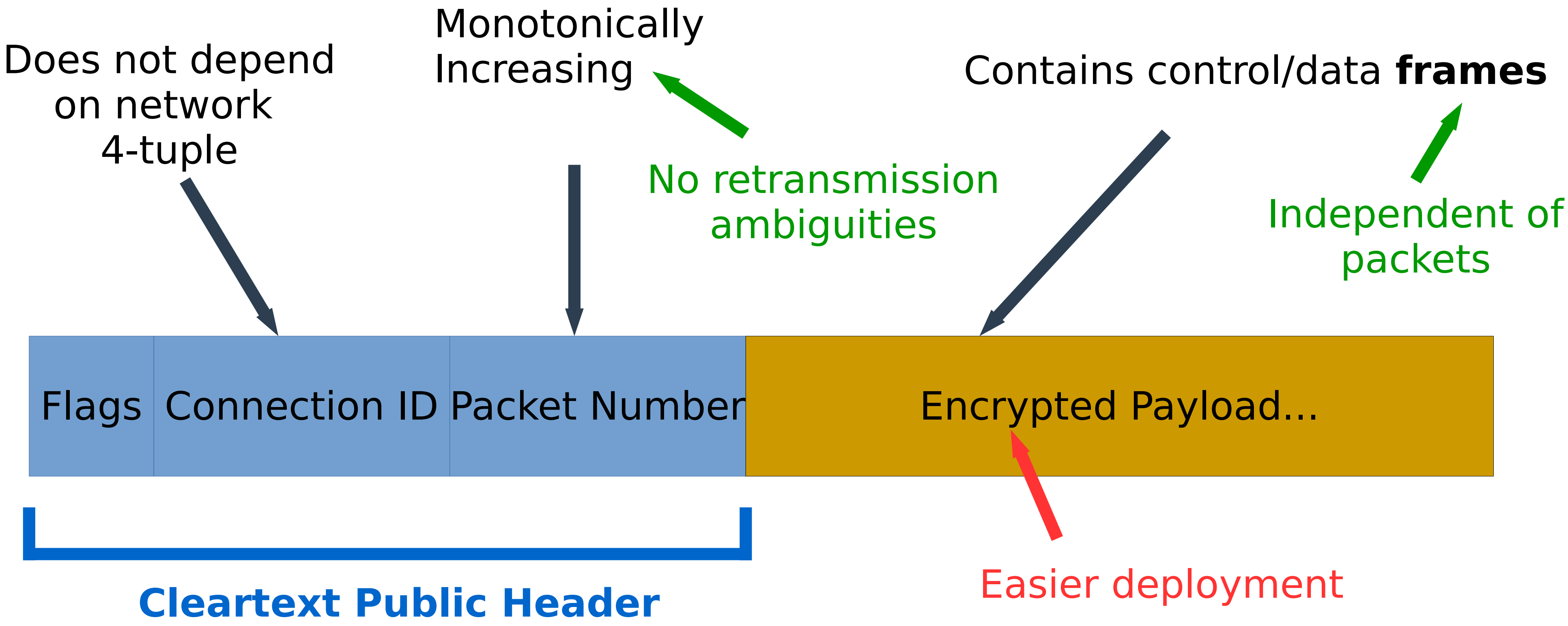
Connection ID

Packet Number

Encrypted Payload...

Cleartext Public Header

QUIC Packet



Does not depend on network 4-tuple

Monotonically Increasing

Contains control/data **frames**

No retransmission ambiguities

Independent of packets

Flags

Connection ID

Packet Number

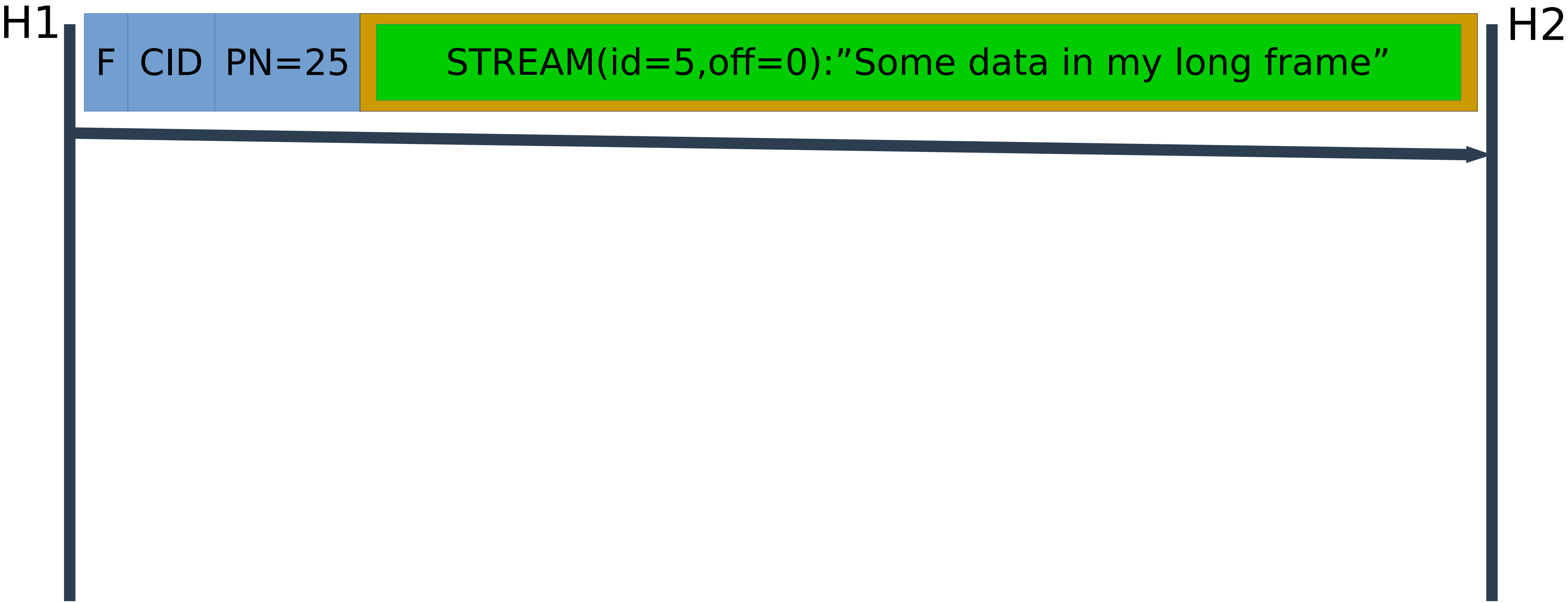
Encrypted Payload...

Cleartext Public Header

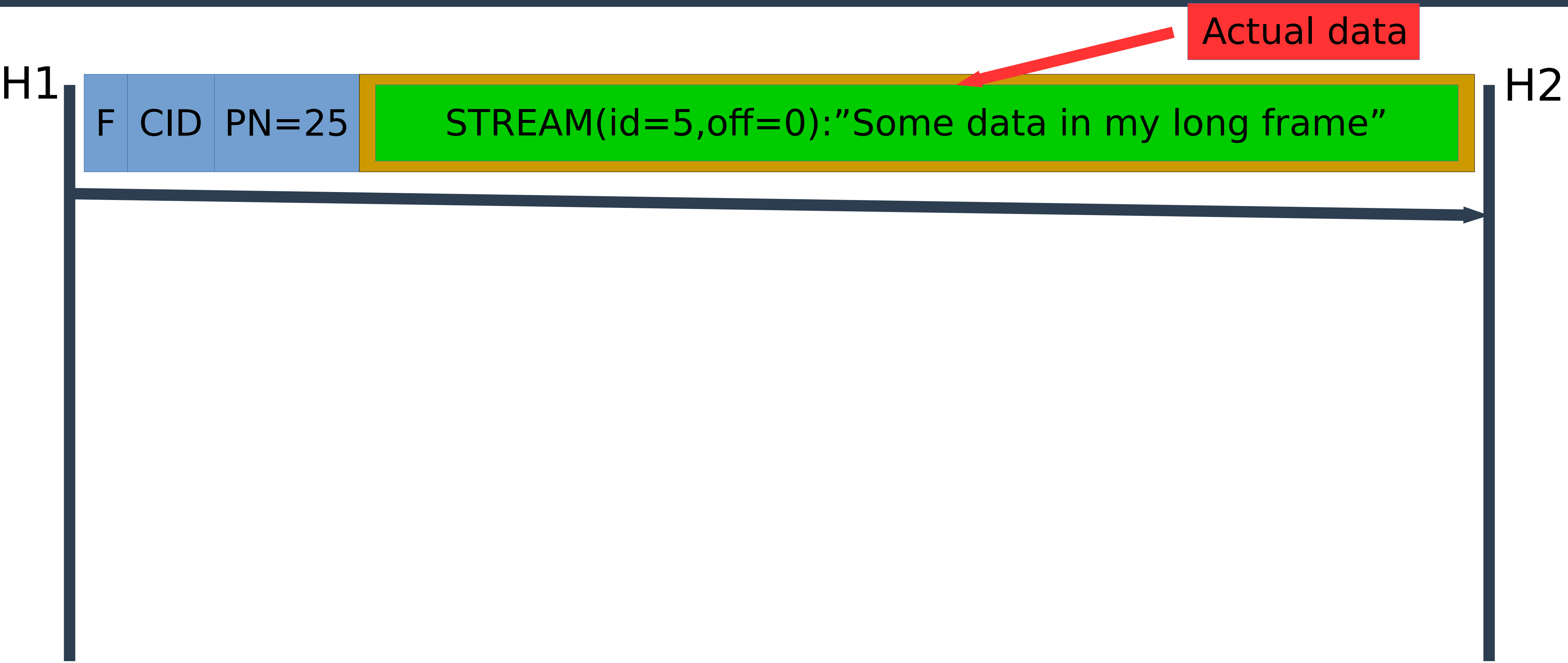
Easier deployment

QUIC Data Transfer

QUIC Data Transfer



QUIC Data Transfer



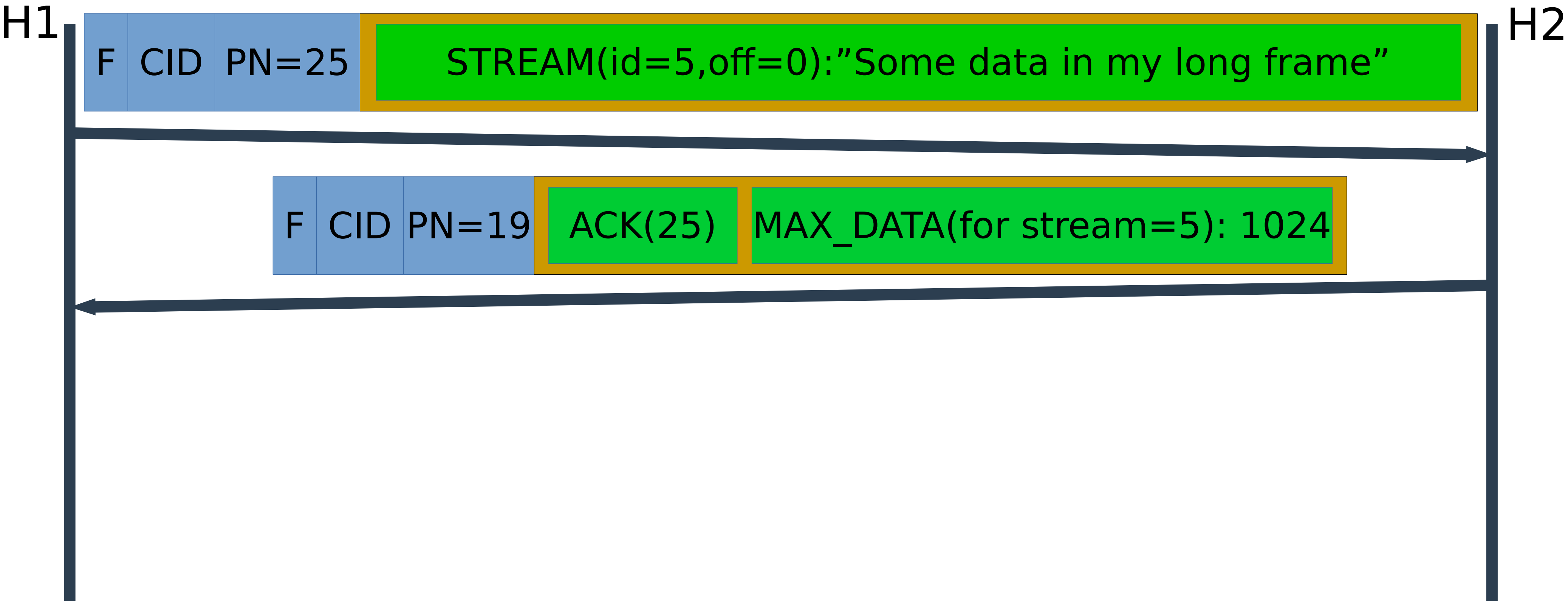
Actual data

H1

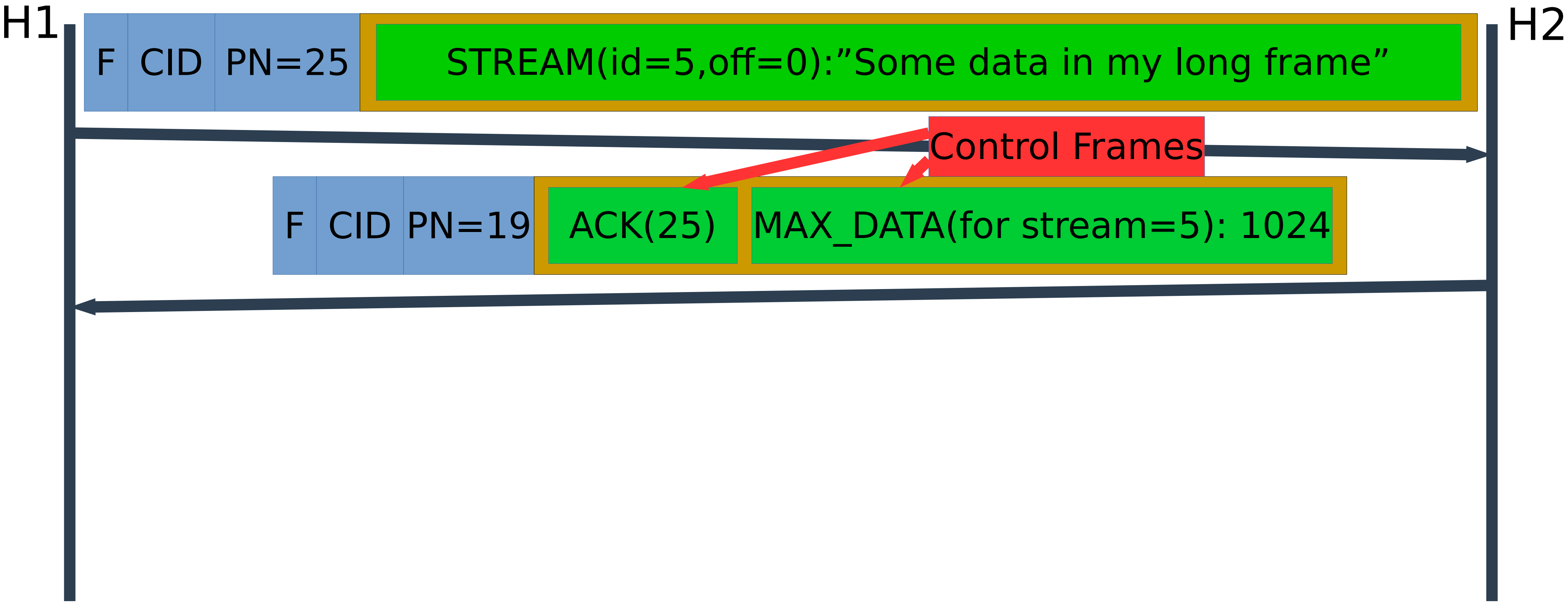
H2

F | CID | PN=25 | STREAM(id=5,off=0):”Some data in my long frame”

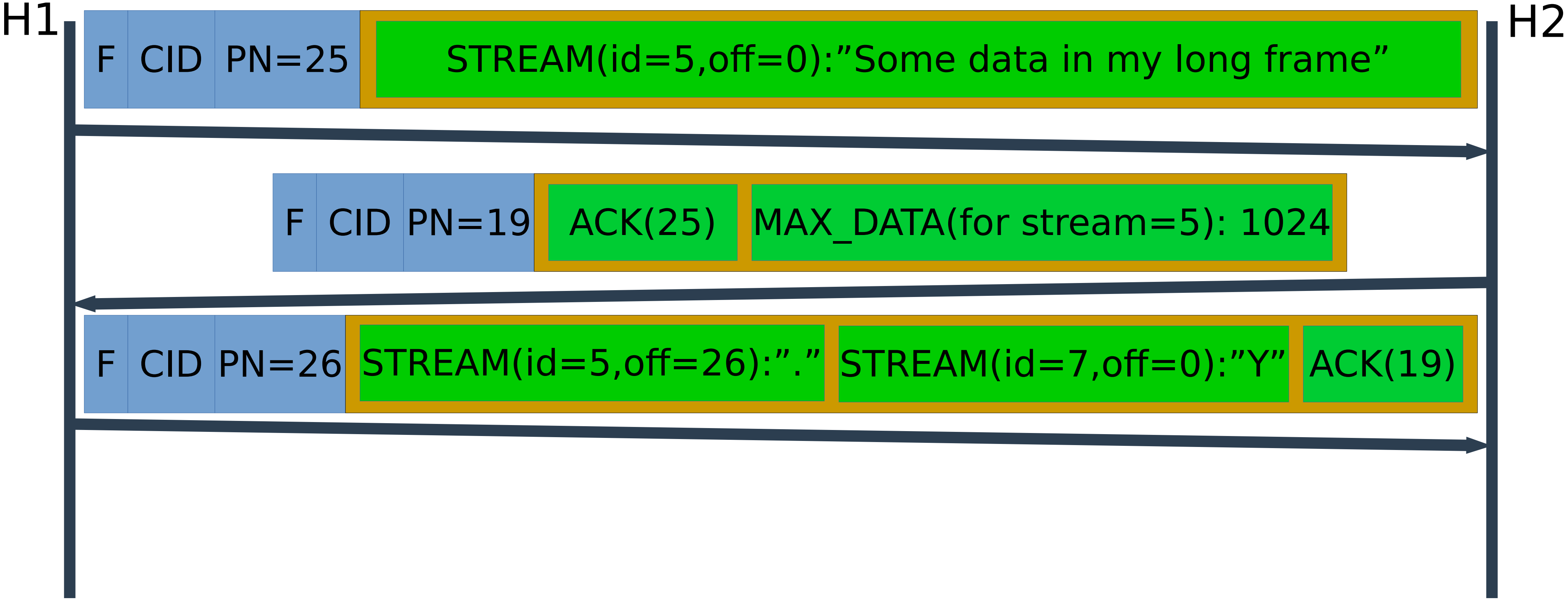
QUIC Data Transfer


H1
H2
F
CID
PN=25
STREAM(id=5,off=0):”Some data in my long frame”
F
CID
PN=19
ACK(25)
MAX_DATA(for stream=5): 1024

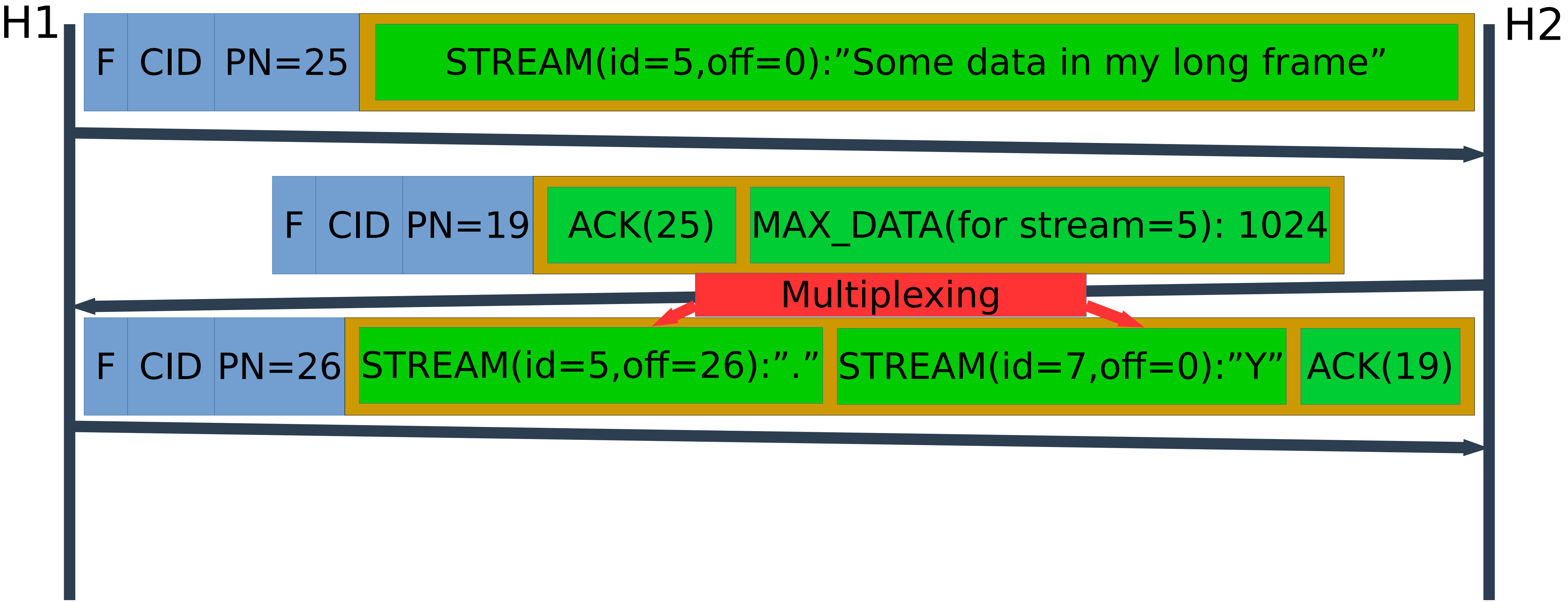
QUIC Data Transfer


H1
H2
F
CID
PN=25
STREAM(id=5,off=0):"Some data in my long frame"
Control Frames
F
CID
PN=19
ACK(25)
MAX_DATA(for stream=5): 1024

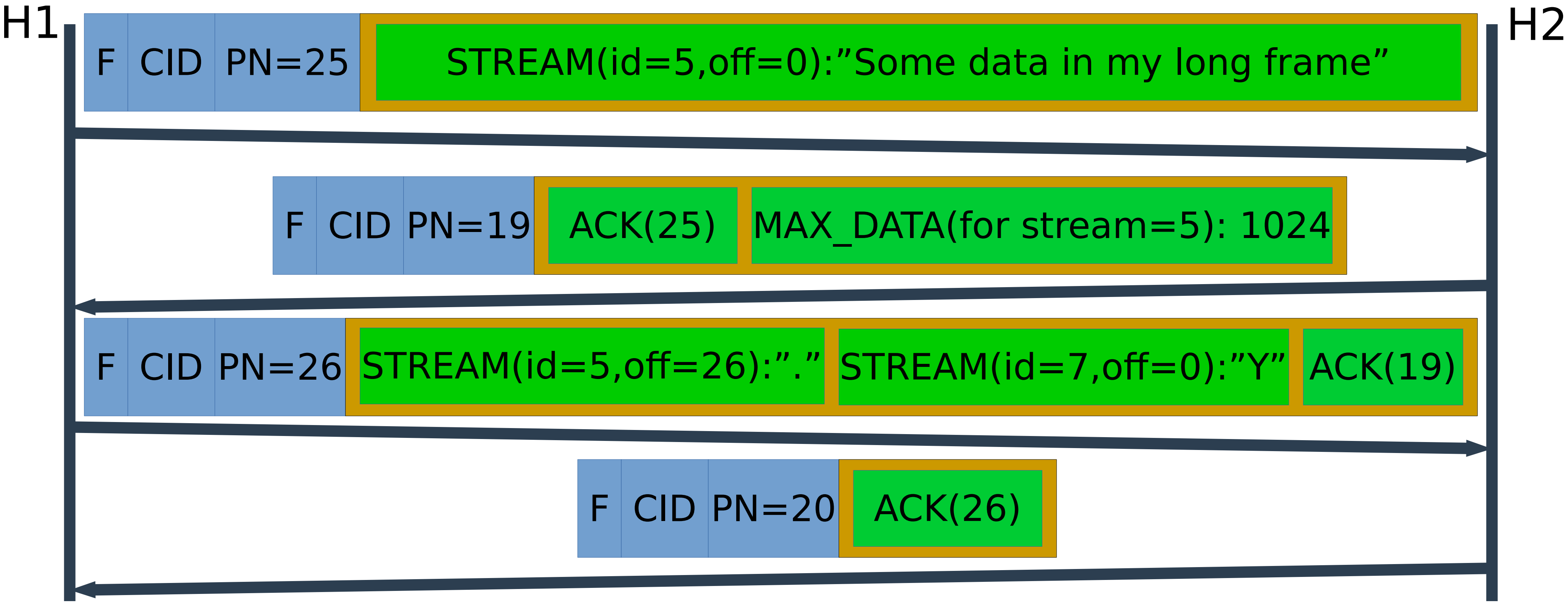
QUIC Data Transfer



QUIC Data Transfer

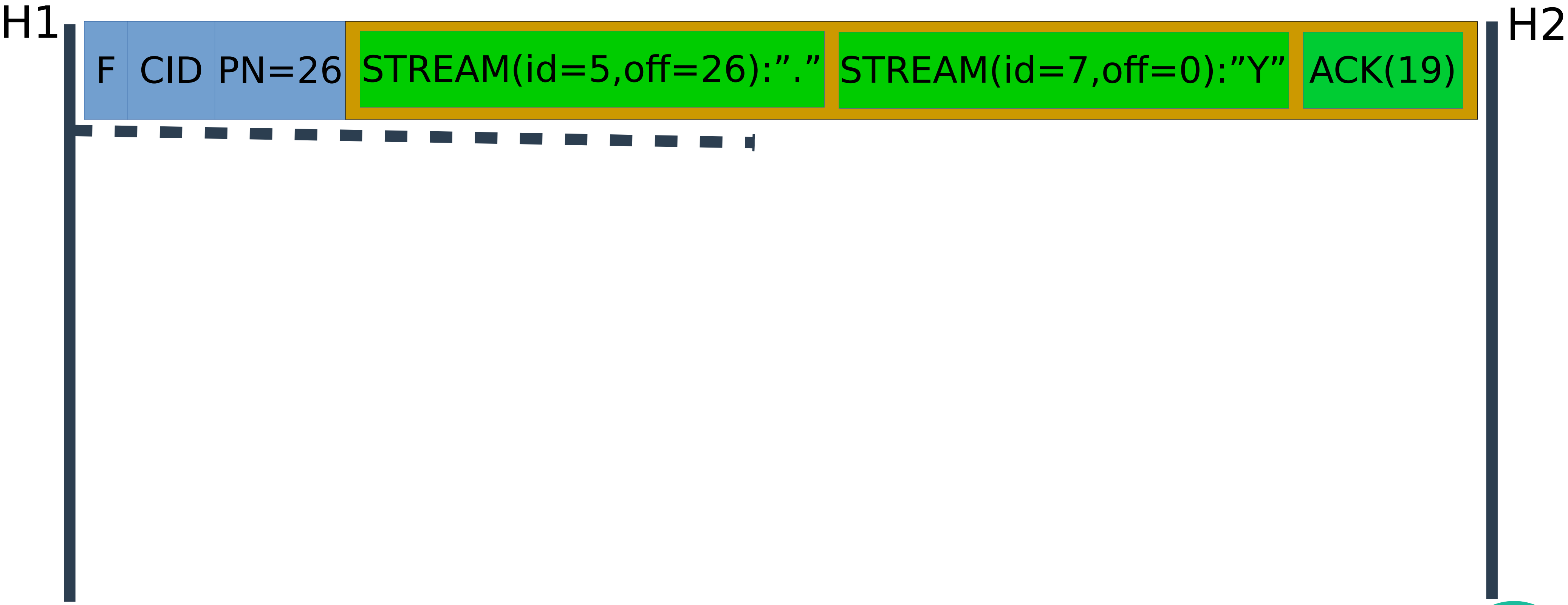


QUIC Data Transfer

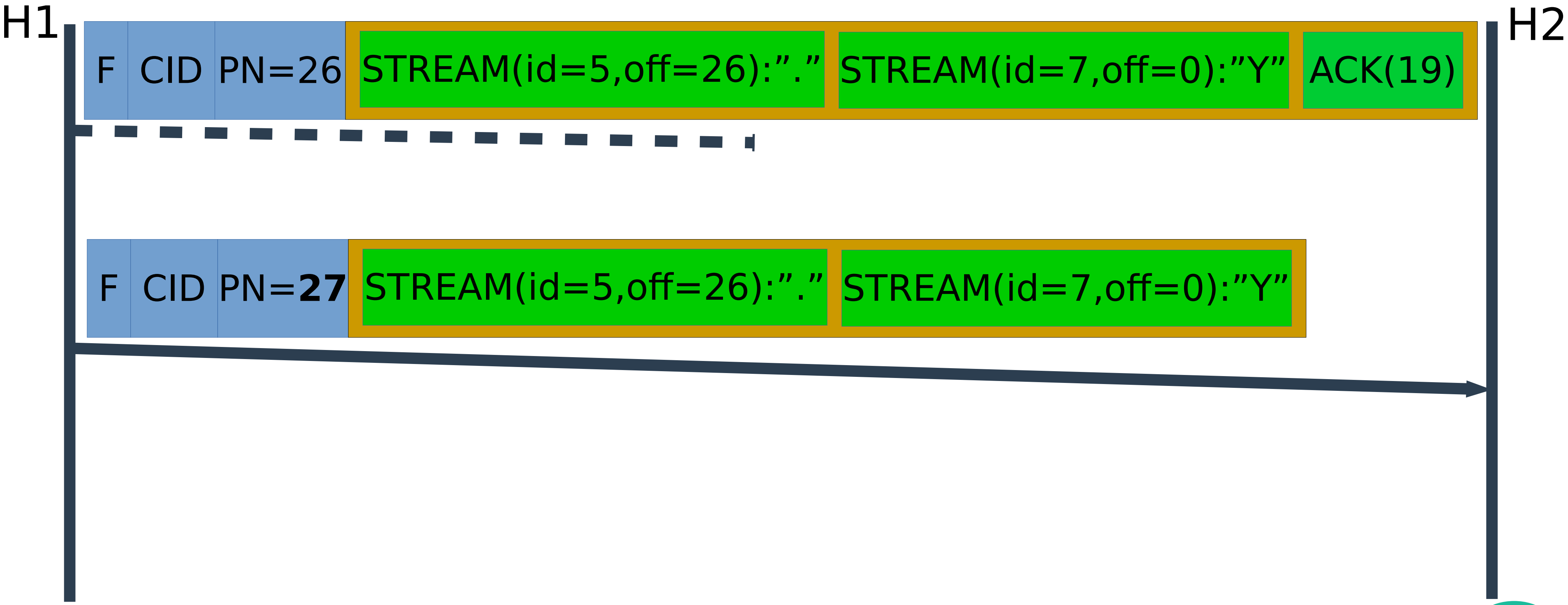


QUIC and Packet Losses

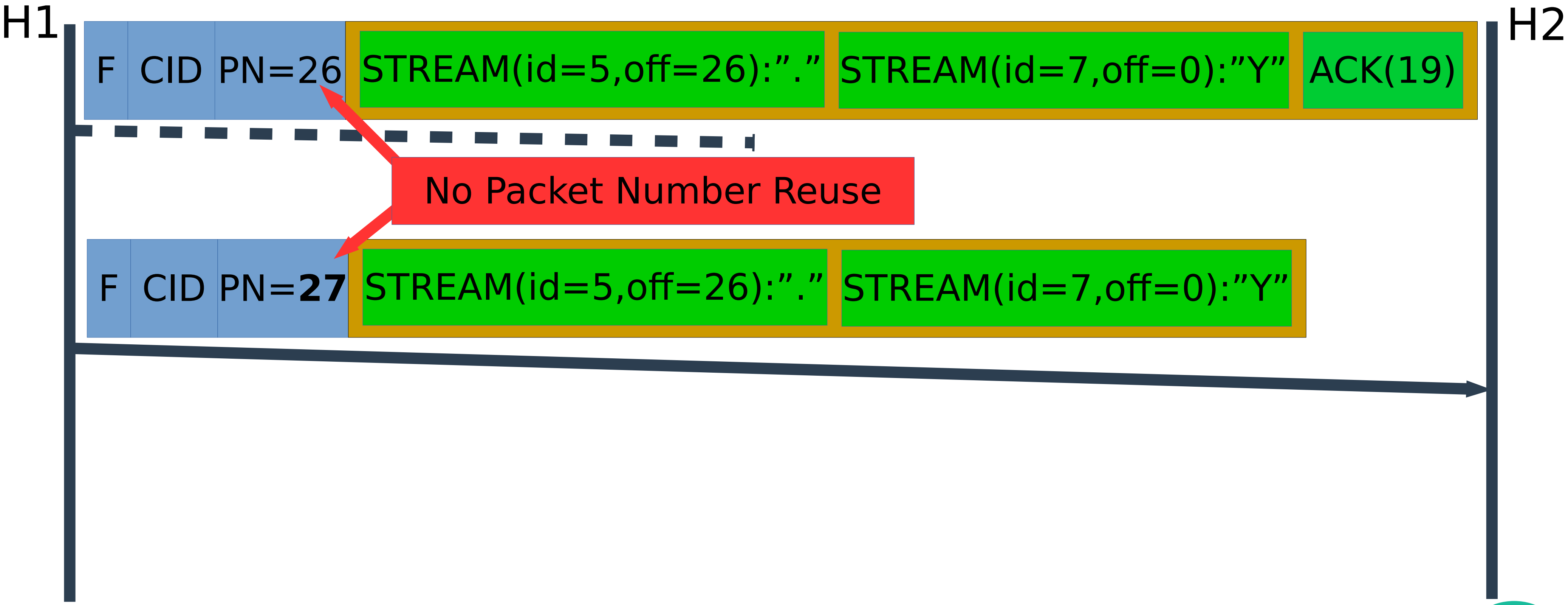
QUIC and Packet Losses



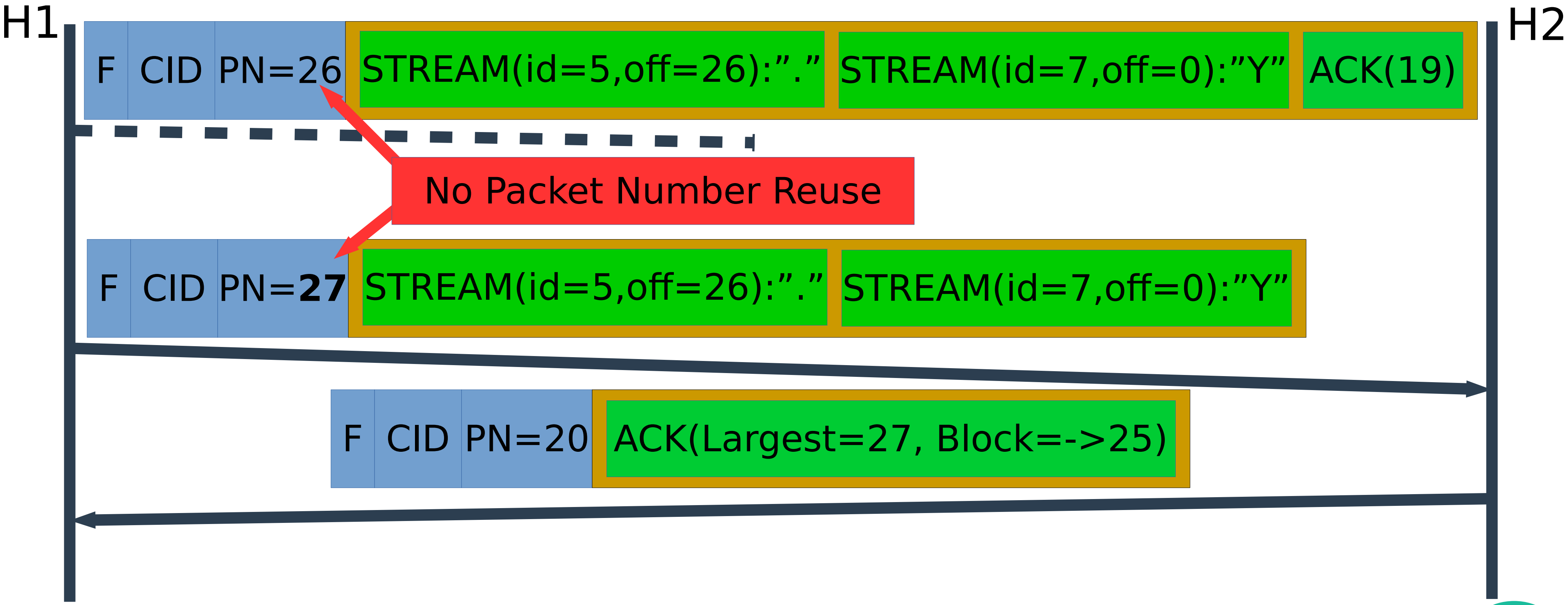
QUIC and Packet Losses



QUIC and Packet Losses



QUIC and Packet Losses



What about Multipath?

Why Multipath QUIC?

- **QUIC assumes a single-path flow**

Why Multipath QUIC?

- QUIC assumes a single-path flow

Why Multipath QUIC?

- **QUIC assumes a single-path flow**

Why Multipath QUIC?

- **QUIC assumes a single-path flow**

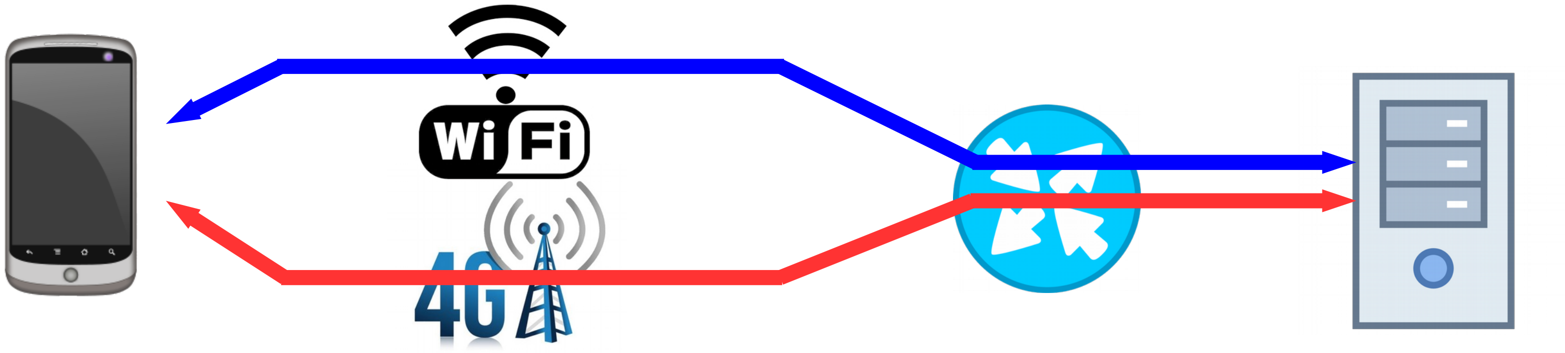
- **Multipath QUIC**
  - Bandwidth aggregation
  - Seamless network handover
    - Can try new WiFi while keeping using LTE

Design of Multipath QUIC

- Connection is composed of a set of paths

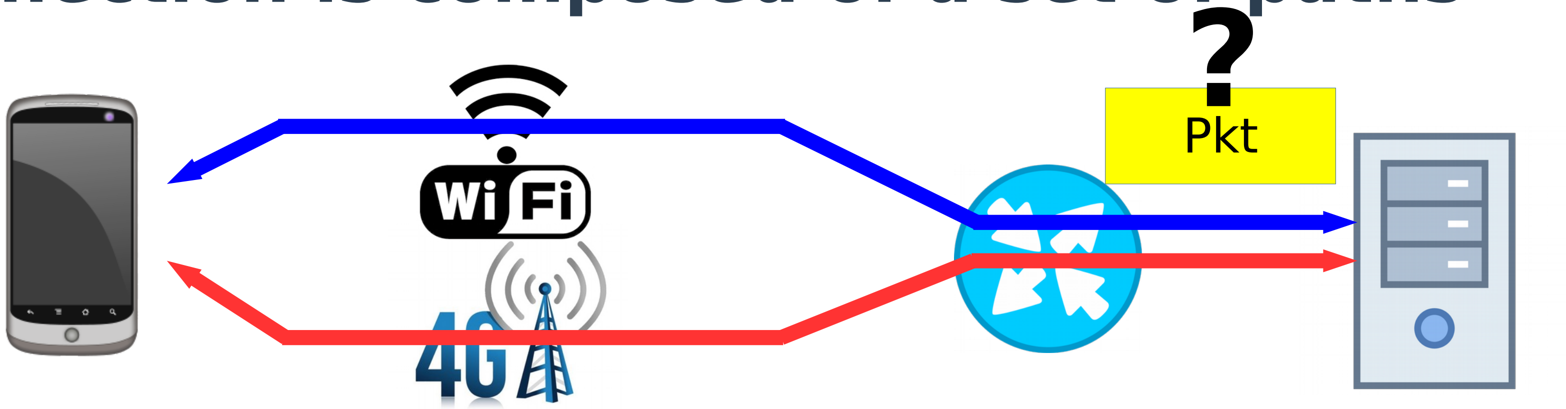
Design of Multipath QUIC

- Connection is composed of a set of paths



Design of Multipath QUIC

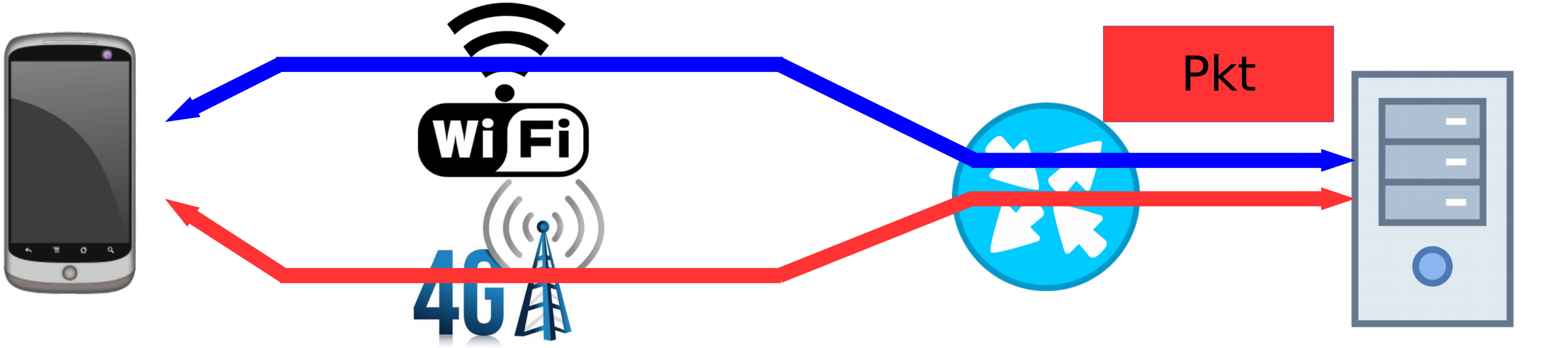
- **Connection is composed of a set of paths**



Performance monitoring?
Loss detection?
Path congestion control?

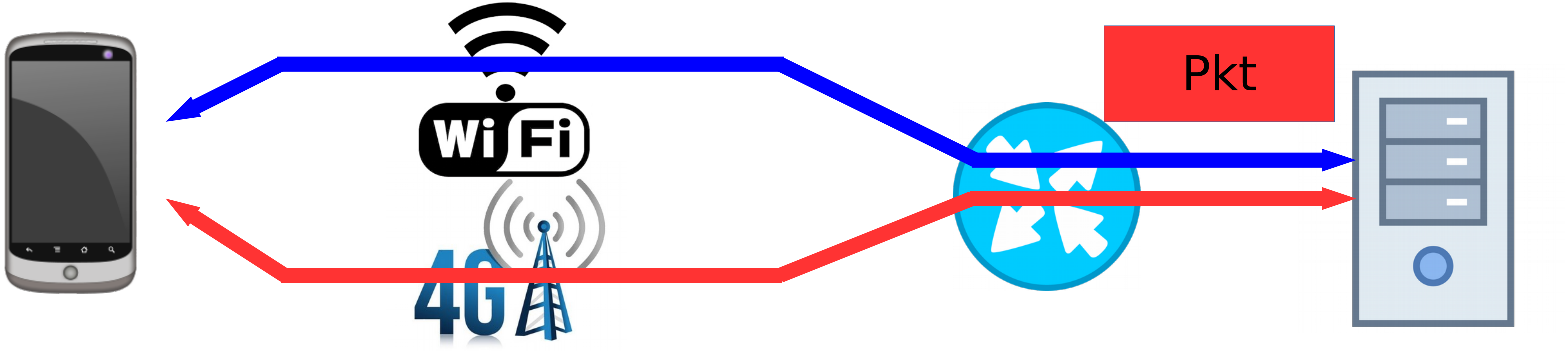
Design of Multipath QUIC

- **Connection is composed of a set of paths**



Design of Multipath QUIC

- **Connection is composed of a set of paths**

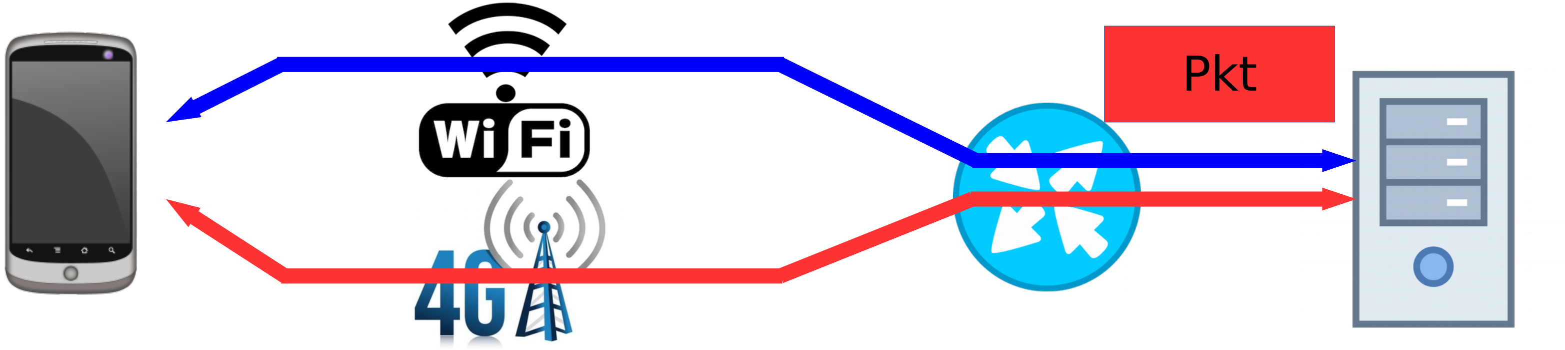


Flags	Connection ID	Path ID	Packet Number	Encrypted Payload...

Explicit path
identification

Design of Multipath QUIC

- **Connection is composed of a set of paths**

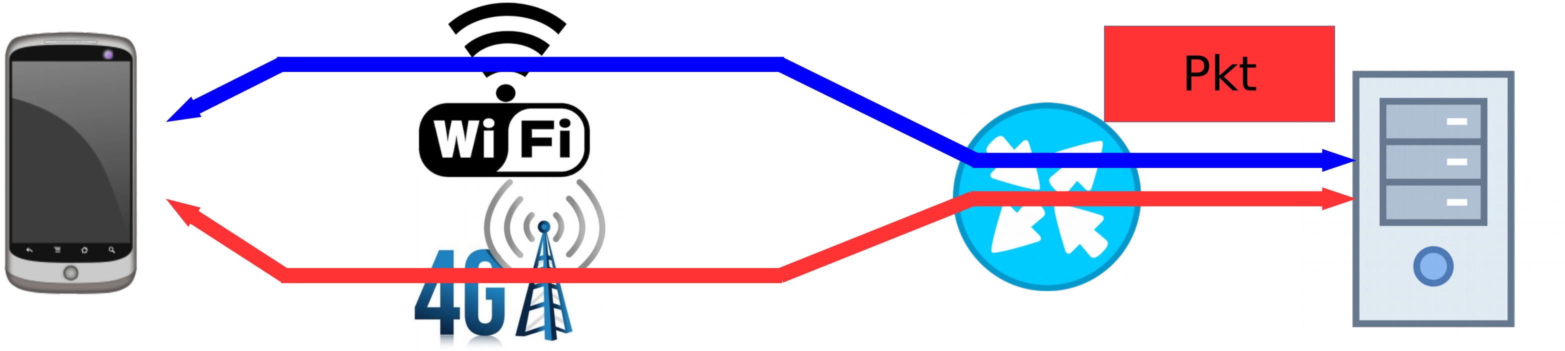


Flags	Connection ID	Path ID	Packet Number	Encrypted Payload...

Explicit path identification → **No path handshake**

Design of Multipath QUIC

- **Connection is composed of a set of paths**

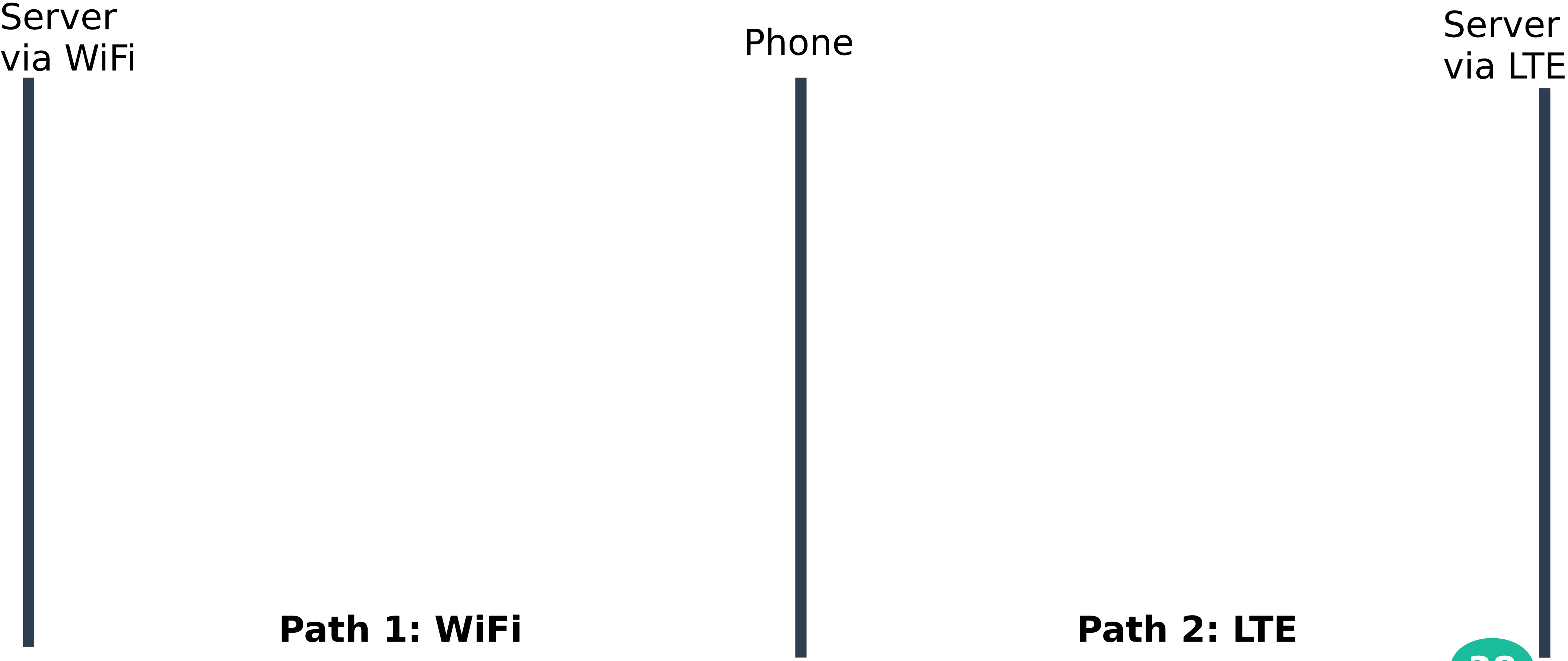


Flags	Connection ID	Path ID	Packet Number	Encrypted Payload...

Explicit path identification → **No path handshake**

Per-path numbering space

Multipath QUIC Data Transfer



Server
via WiFi

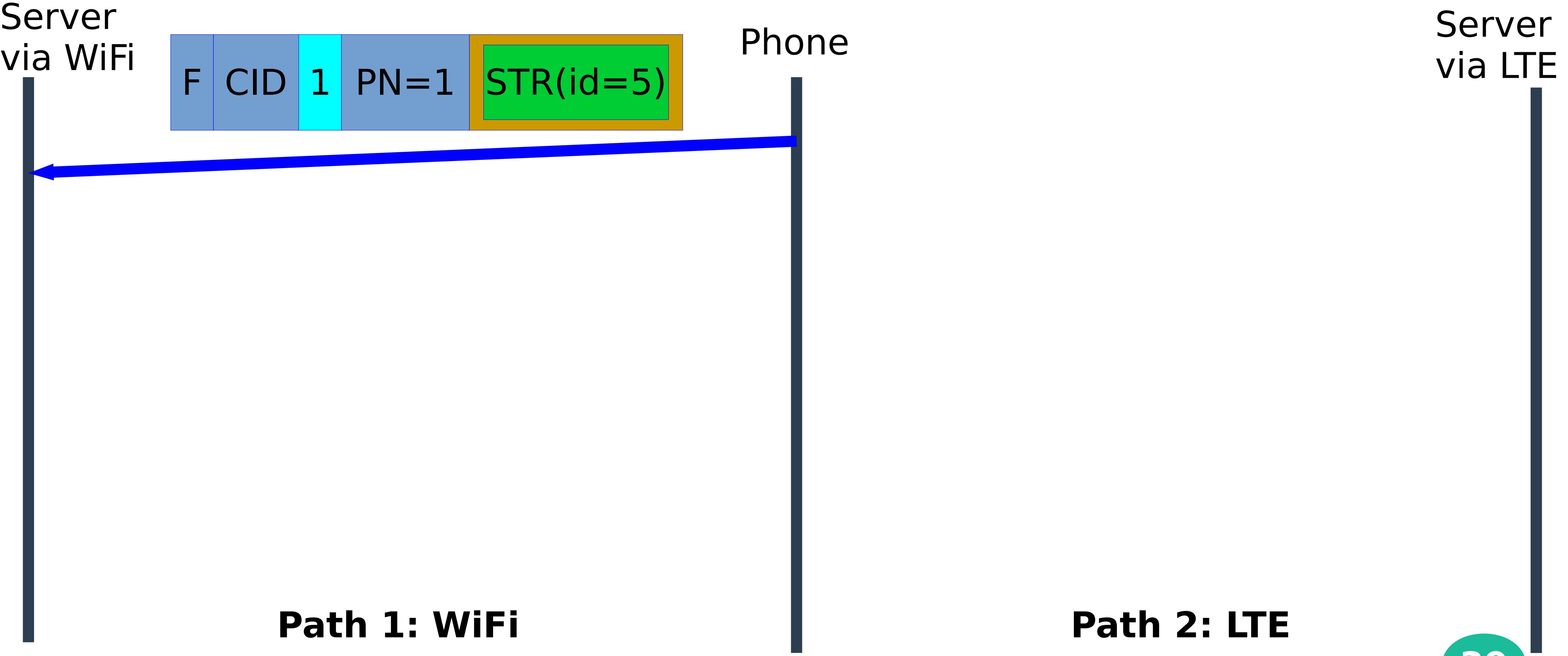
Phone

Server
via LTE

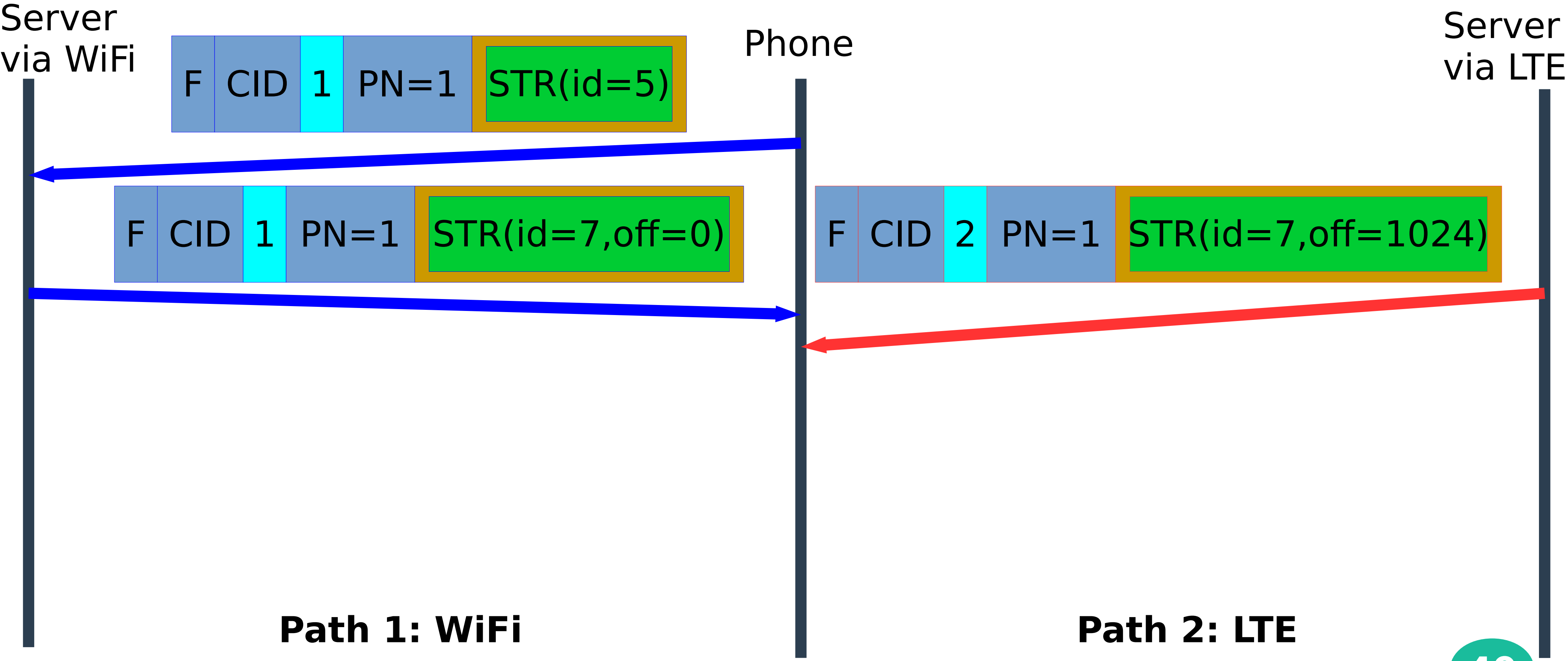
Path 1: WiFi

Path 2: LTE

Multipath QUIC Data Transfer



Multipath QUIC Data Transfer



Server via WiFi

Phone

Server via LTE

F | CID | 1 | PN=1 | STR(id=5)

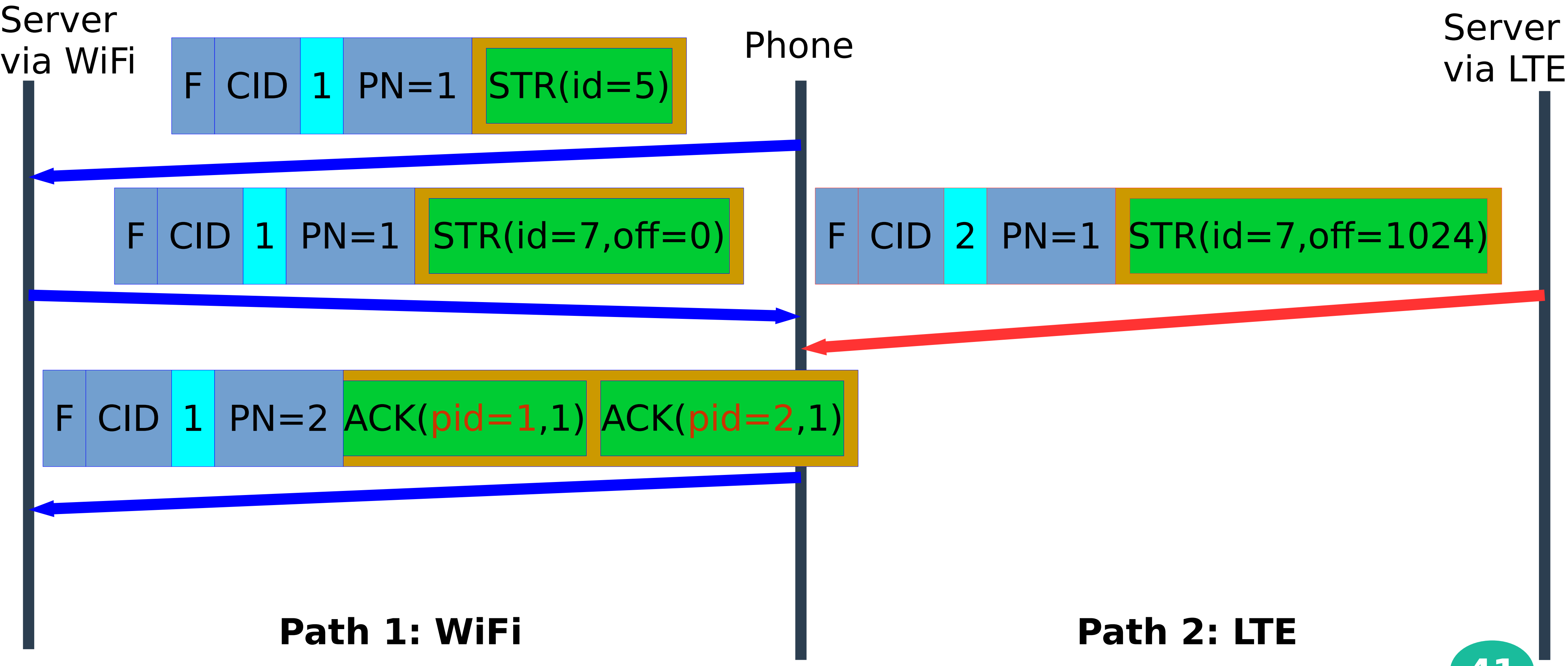
F | CID | 1 | PN=1 | STR(id=7,off=0)

F | CID | 2 | PN=1 | STR(id=7,off=1024)

Path 1: WiFi

Path 2: LTE

Multipath QUIC Data Transfer



Server via WiFi

Phone

Server via LTE

F | CID | 1 | PN=1 | STR(id=5)

F | CID | 1 | PN=1 | STR(id=7,off=0)

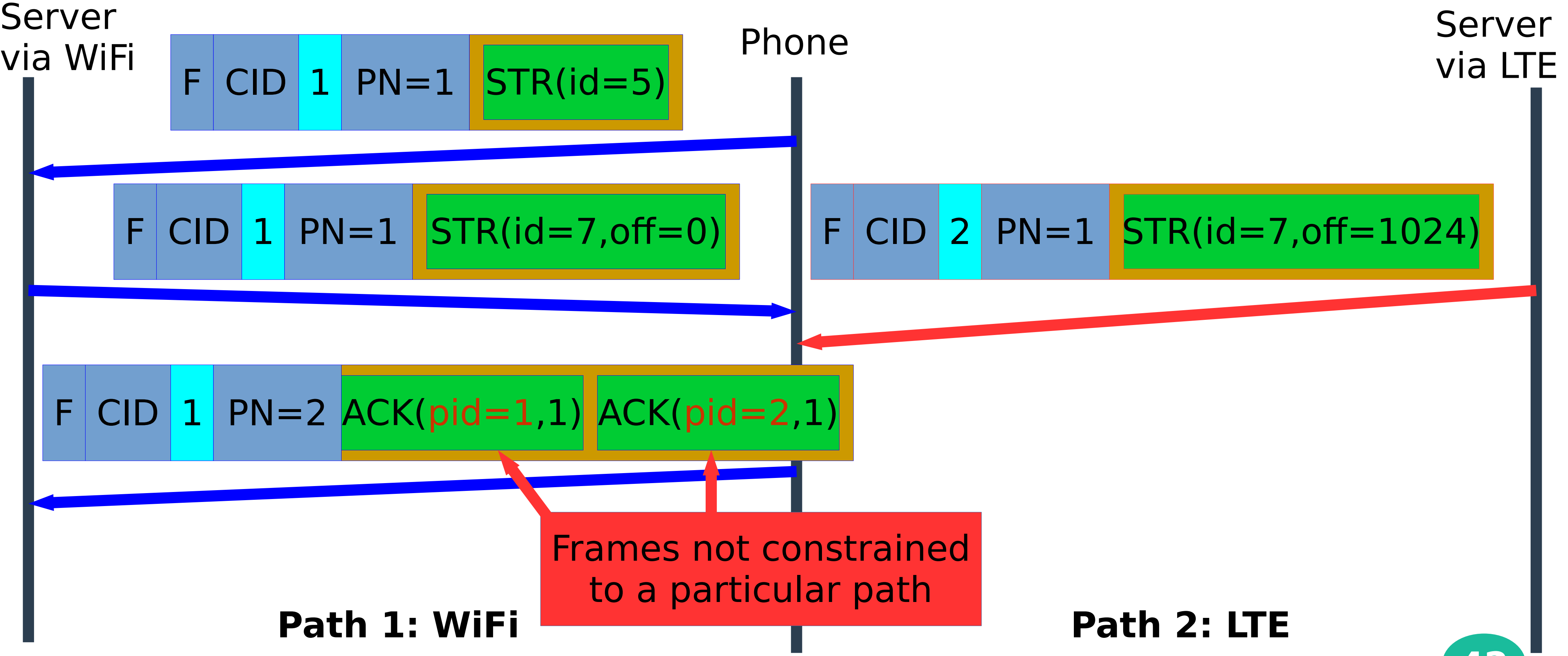
F | CID | 2 | PN=1 | STR(id=7,off=1024)

F | CID | 1 | PN=2 | ACK(pid=1,1) | ACK(pid=2,1)

Path 1: WiFi

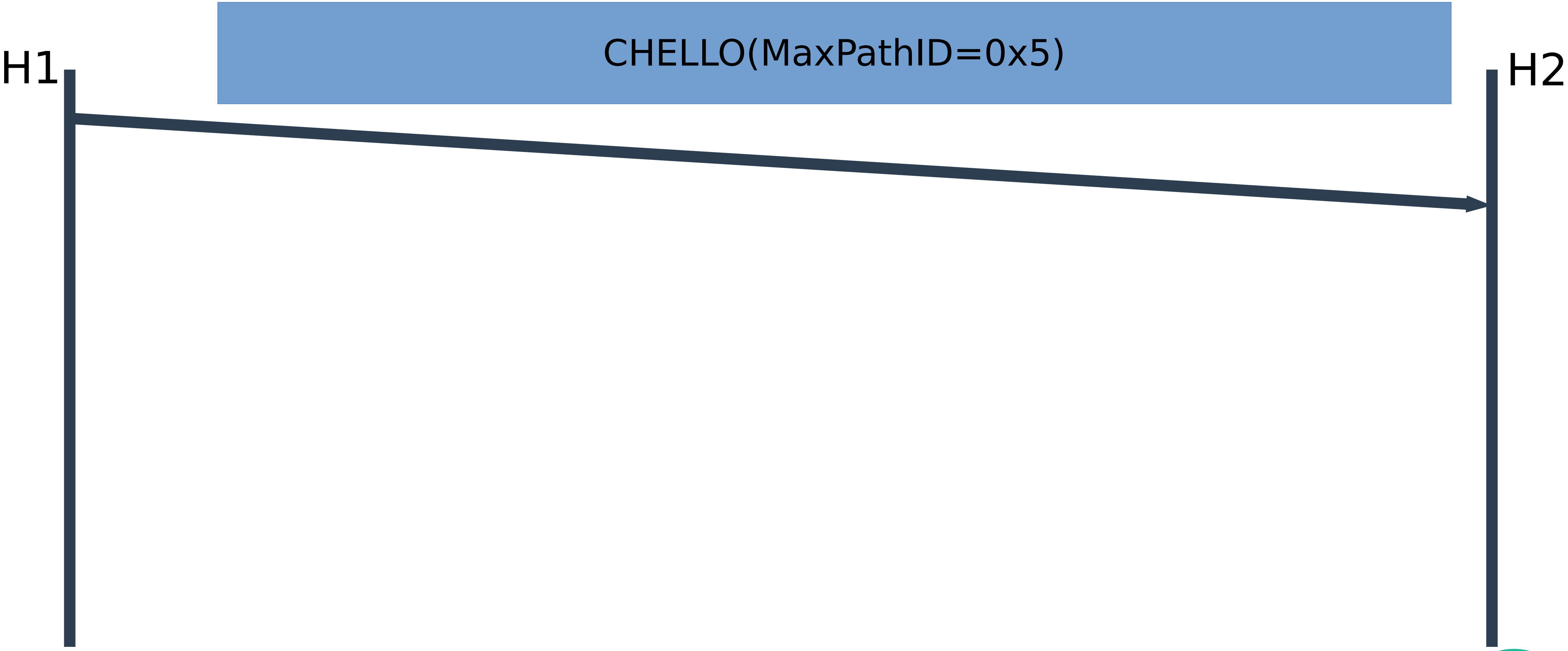
Path 2: LTE

Multipath QUIC Data Transfer


Server
via WiFi
Phone
Server
via LTE
F | CID | 1 | PN=1 | STR(id=5)
F | CID | 1 | PN=1 | STR(id=7,off=0)
F | CID | 2 | PN=1 | STR(id=7,off=1024)
F | CID | 1 | PN=2 | ACK(pid=1,1) | ACK(pid=2,1)
Frames not constrained to a particular path
Path 1: WiFi
Path 2: LTE

Multipath Negotiation

Multipath Negotiation

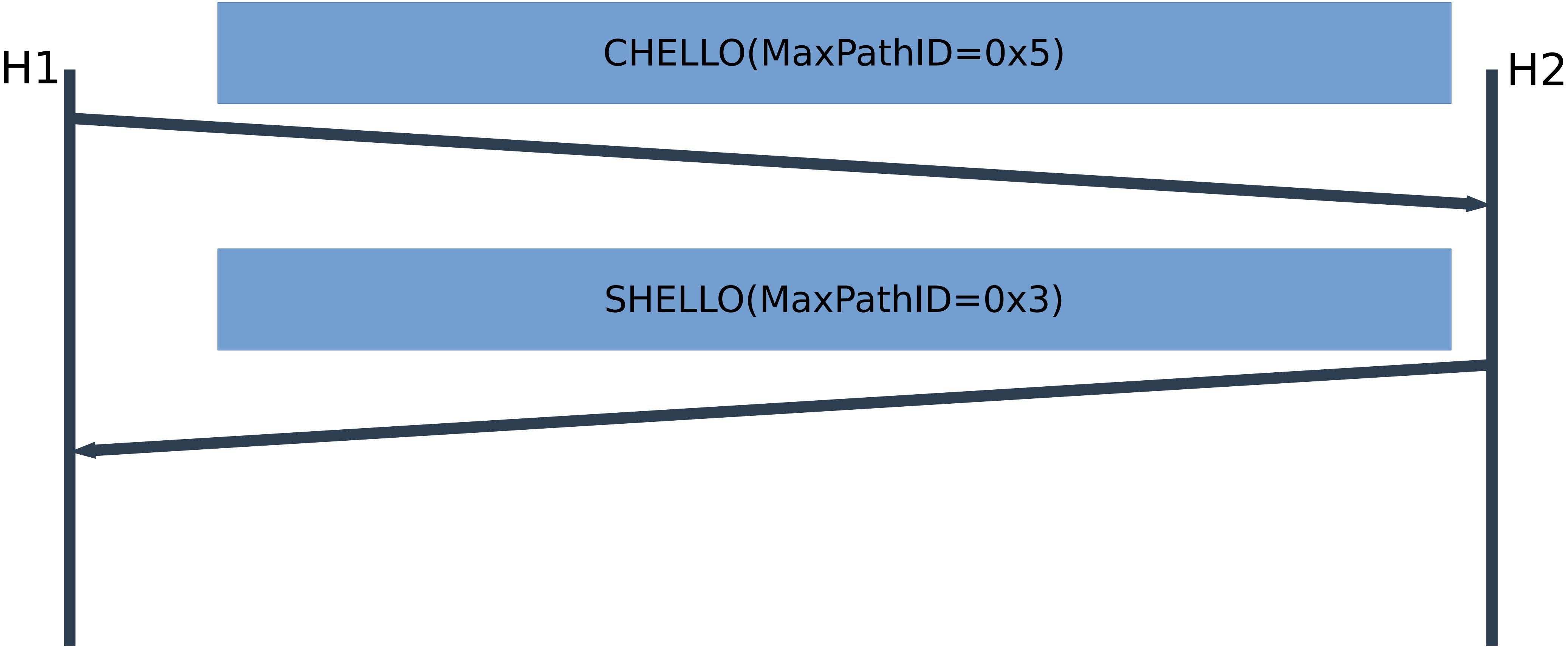


H1

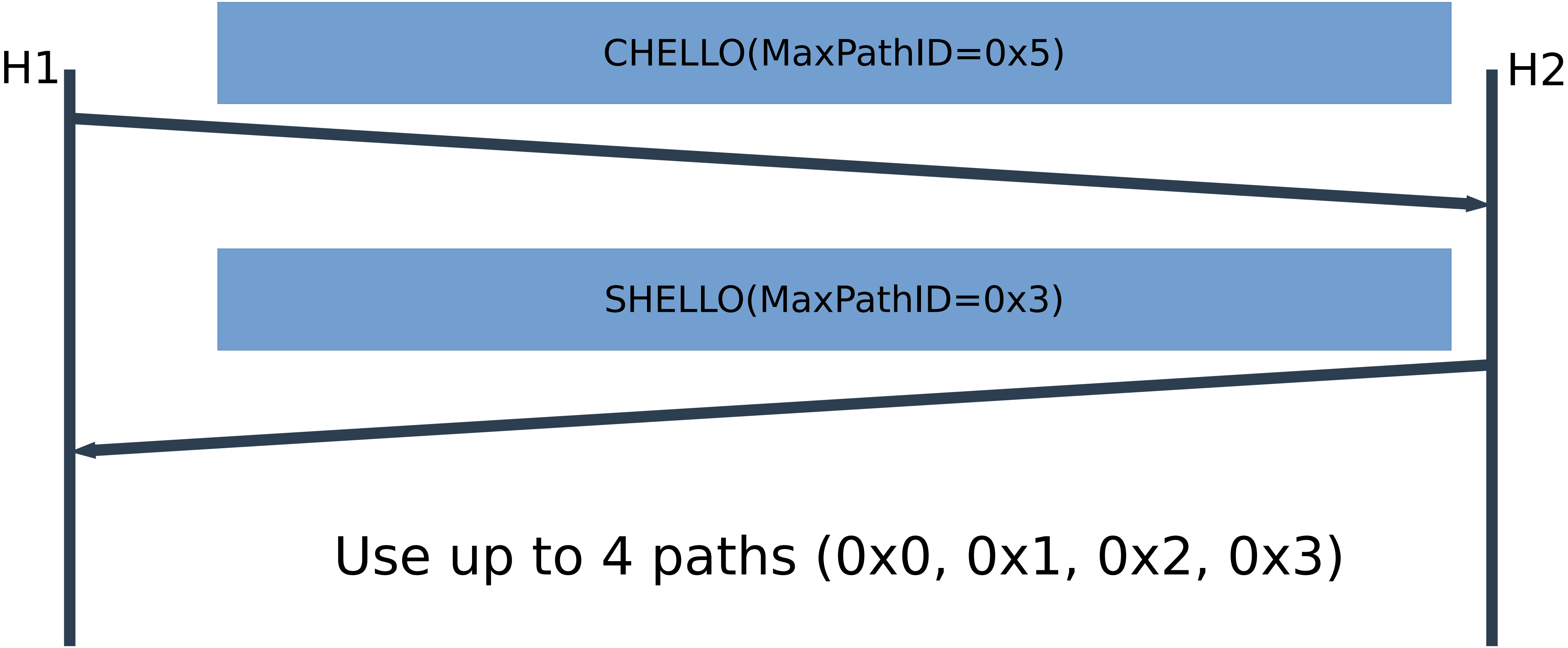
CHELLO(MaxPathID=0x5)

H2

Multipath Negotiation



Multipath Negotiation



H1

H2

CHELLO(MaxPathID=0x5)

SHELLO(MaxPathID=0x3)

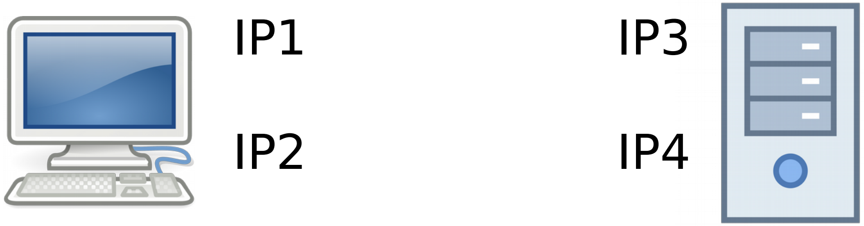
Use up to 4 paths (0x0, 0x1, 0x2, 0x3)

Multipath Mechanisms

- Path management
- Packet scheduling
- Congestion control

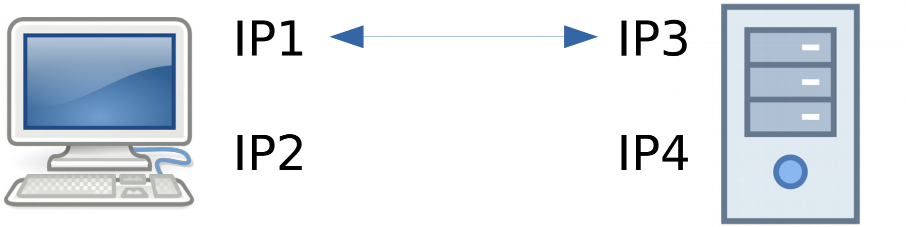
Path Management

- **How and when paths are established?**



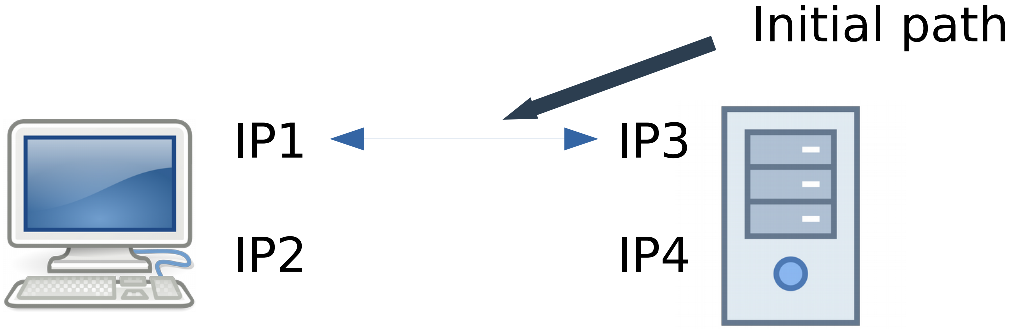
Path Management

- **How and when paths are established?**



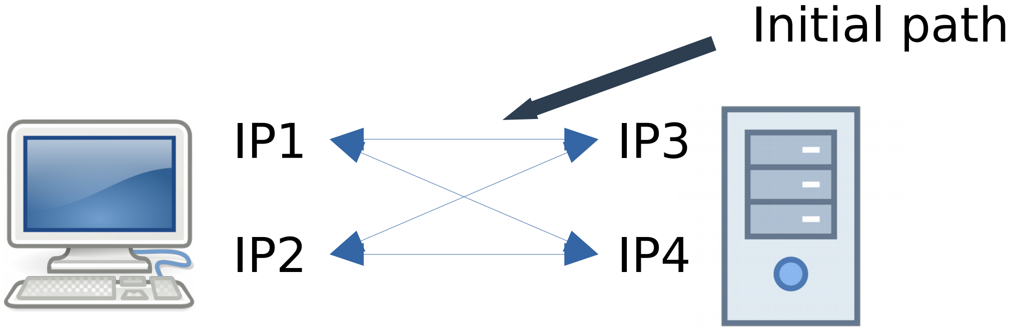
Path Management

- **How and when paths are established?**



Path Management

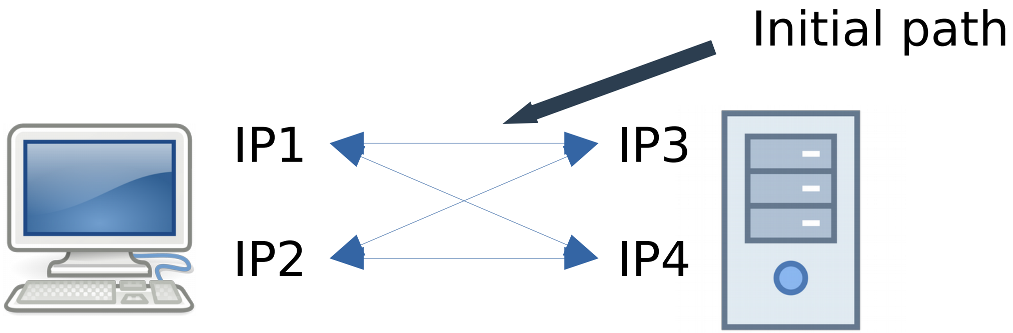
- **How and when paths are established?**



- **Fullmesh fashion**

Path Management

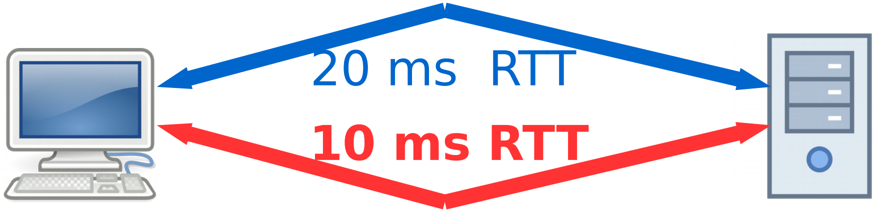
- **How and when paths are established?**



- **Fullmesh fashion**
- **ADD_ADDRESS + REMOVE_ADDRESS frames**

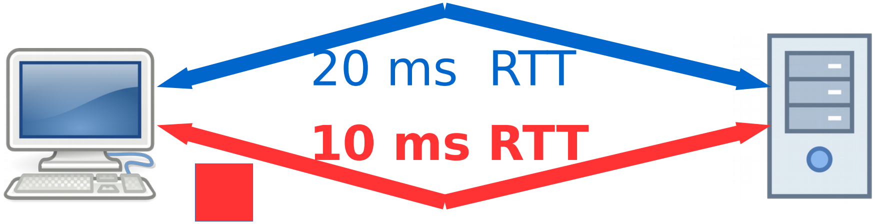
Packet Scheduling

- **Lowest-latency first**



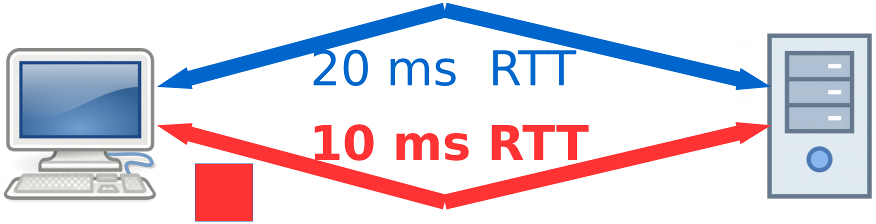
Packet Scheduling

- **Lowest-latency first**

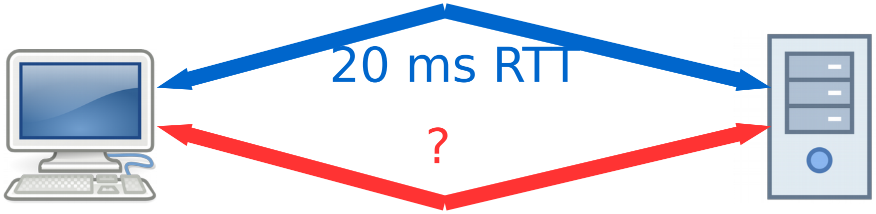


Packet Scheduling

- **Lowest-latency first**

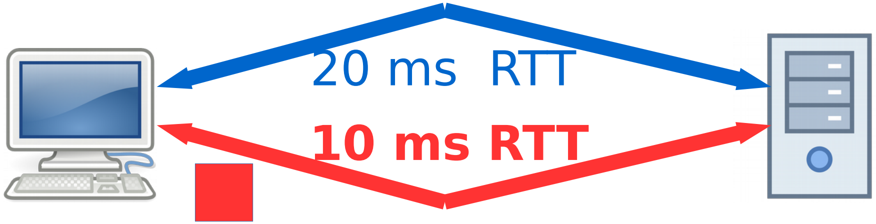


- **What about when starting using a new path?**

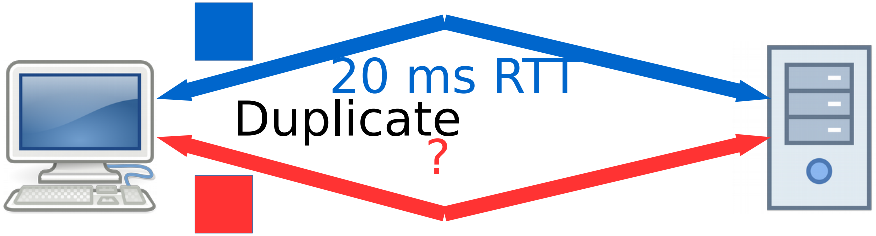


Packet Scheduling

- **Lowest-latency first**

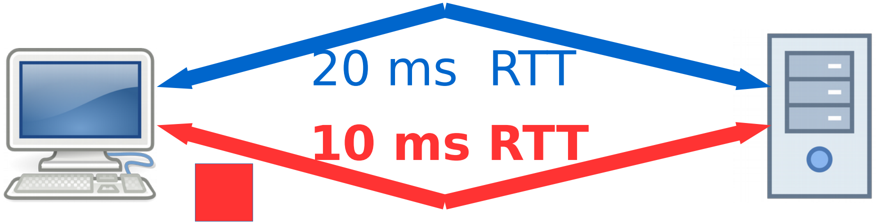


- **What about when starting using a new path?**

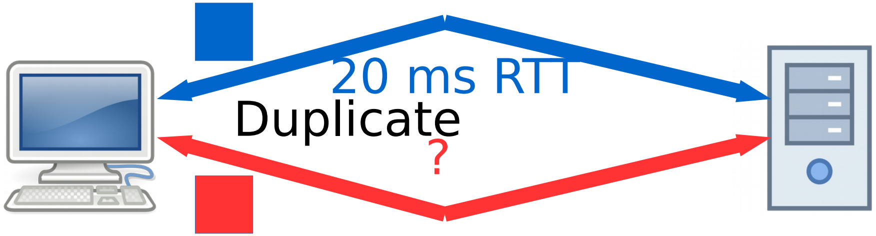


Packet Scheduling

- **Lowest-latency first**



- **What about when starting using a new path?**



- **Schedule all frames (not only STREAM)**

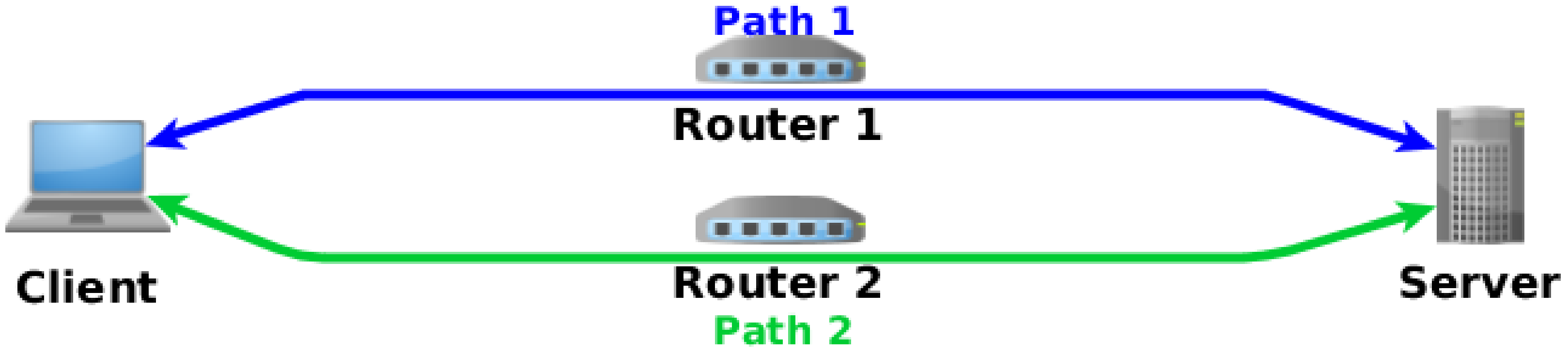
Congestion Control

- **Multipath = need for coupled CC**
  - CUBIC would be unfair
- **Opportunistic Linked Increase Algorithm**
  - MPTCP state-of-the-art

How well does Multipath QUIC perform?

Evaluation of Multipath QUIC

- **(Multipath) QUIC vs. (Multipath) TCP**
  - Multipath QUIC: quic-go
  - Linux Multipath TCP v0.91 with default settings
- **Mininet environment with 2 paths**

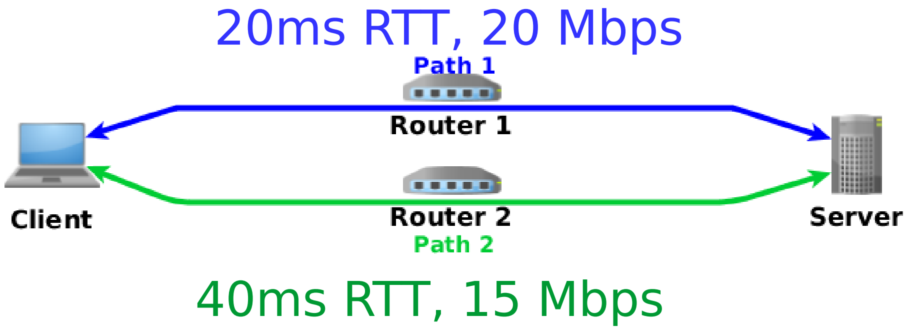


Evaluating Bandwith Aggregation

- **Download of 20 MB file**
  - Over a single stream
  - Collect the transfer time

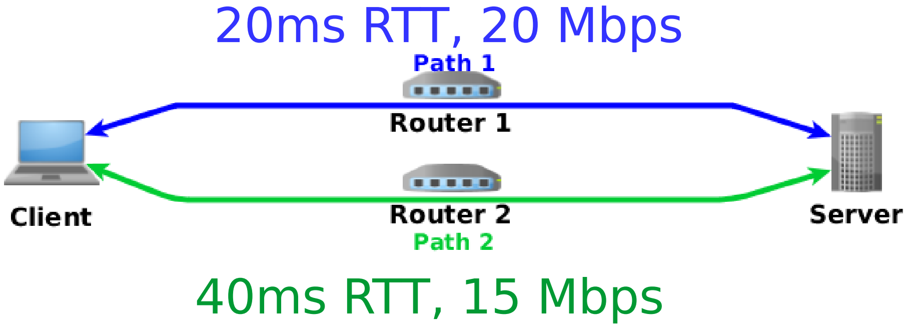
Evaluating Bandwith Aggregation

- **Download of 20 MB file**
  - Over a single stream
  - Collect the transfer time
- **For a loss-free scenario**



Evaluating Bandwith Aggregation

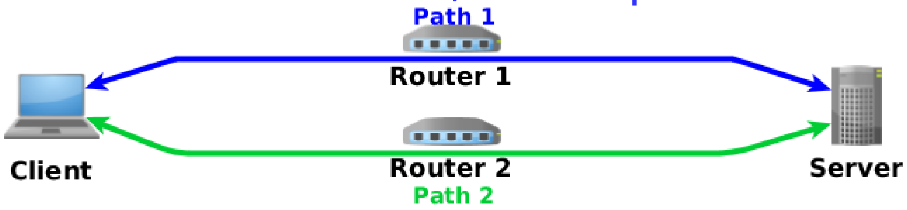
- **Download of 20 MB file**
  - Over a single stream
  - Collect the transfer time
- **For a loss-free scenario**
  - MPQUIC has 13% speedup compared to MPTCP



Evaluating Bandwith Aggregation

- **Download of 20 MB file**
  - Over a single stream
  - Collect the transfer time
- **For a loss-free scenario**
  - MPQUIC has 13% speedup compared to MPTCP
    - MPQUIC less bursty than MPTCP
    - Probably due to CC skew on initial path in MPTCP

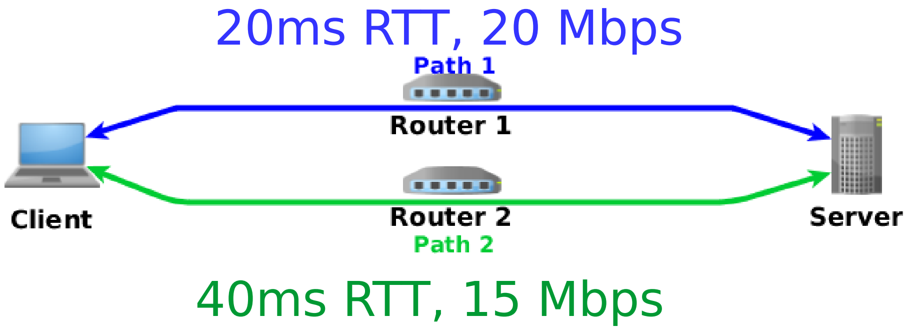
20ms RTT, 20 Mbps



40ms RTT, 15 Mbps

Evaluating Bandwith Aggregation

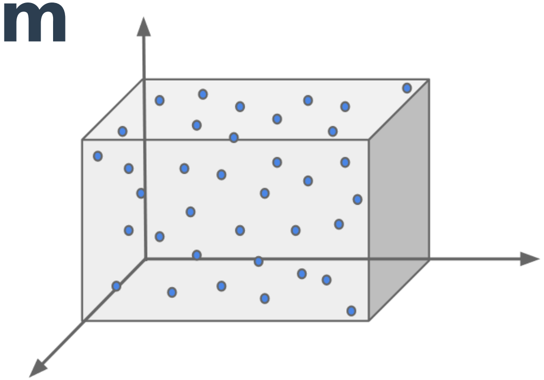
- **Download of 20 MB file**
  - Over a single stream
  - Collect the transfer time



- **For a loss-free scenario**
  - MPQUIC has 13% speedup compared to MPTCP
    - MPQUIC less bursty than MPTCP
    - Probably due to CC skew on initial path in MPTCP
- **But what about other topologies?**

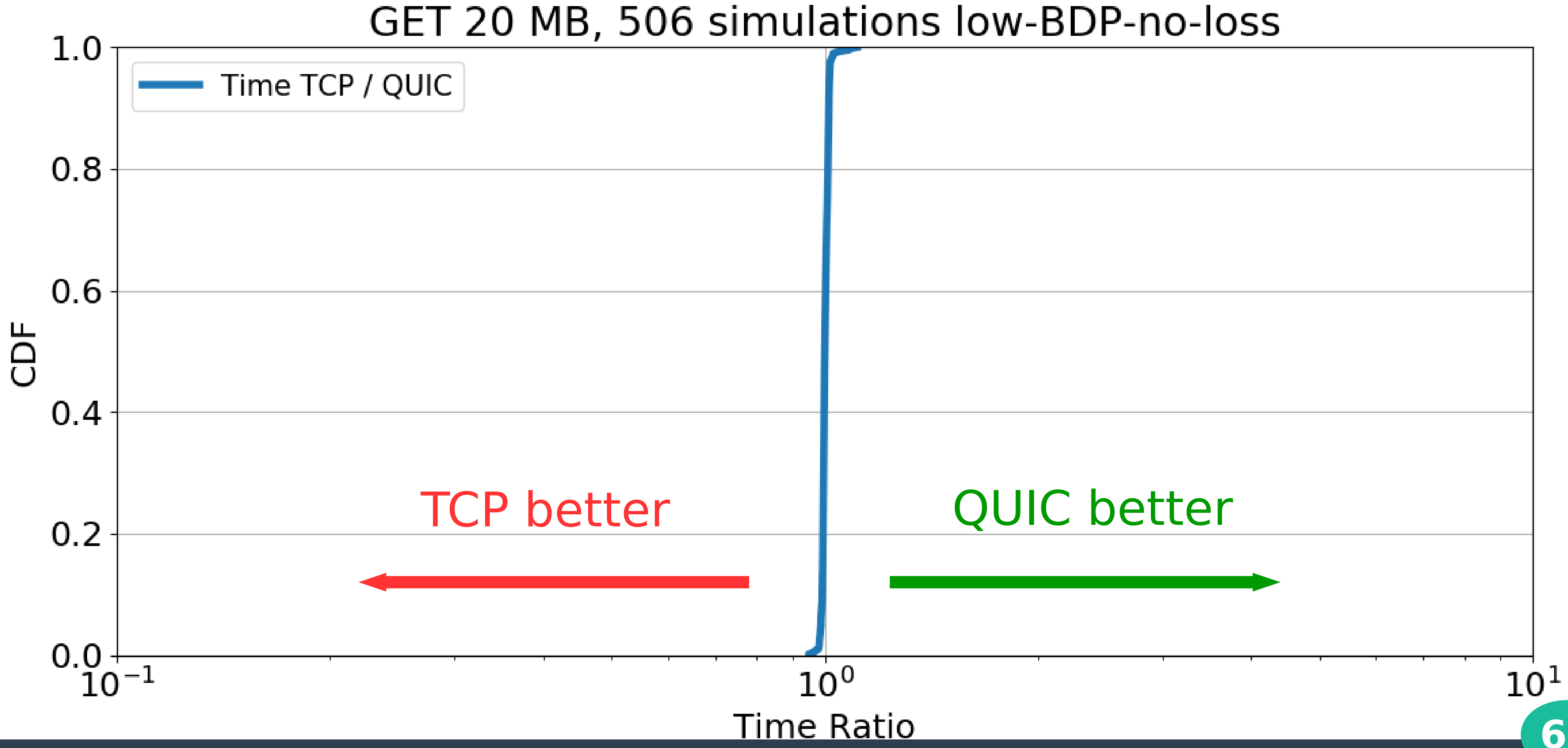
Evaluating Bandwidth Aggregation

- **Experimental design, WSP algorithm**
- **2x253 network scenarios**
  - Vary the initial path
- **Median over 15 runs**

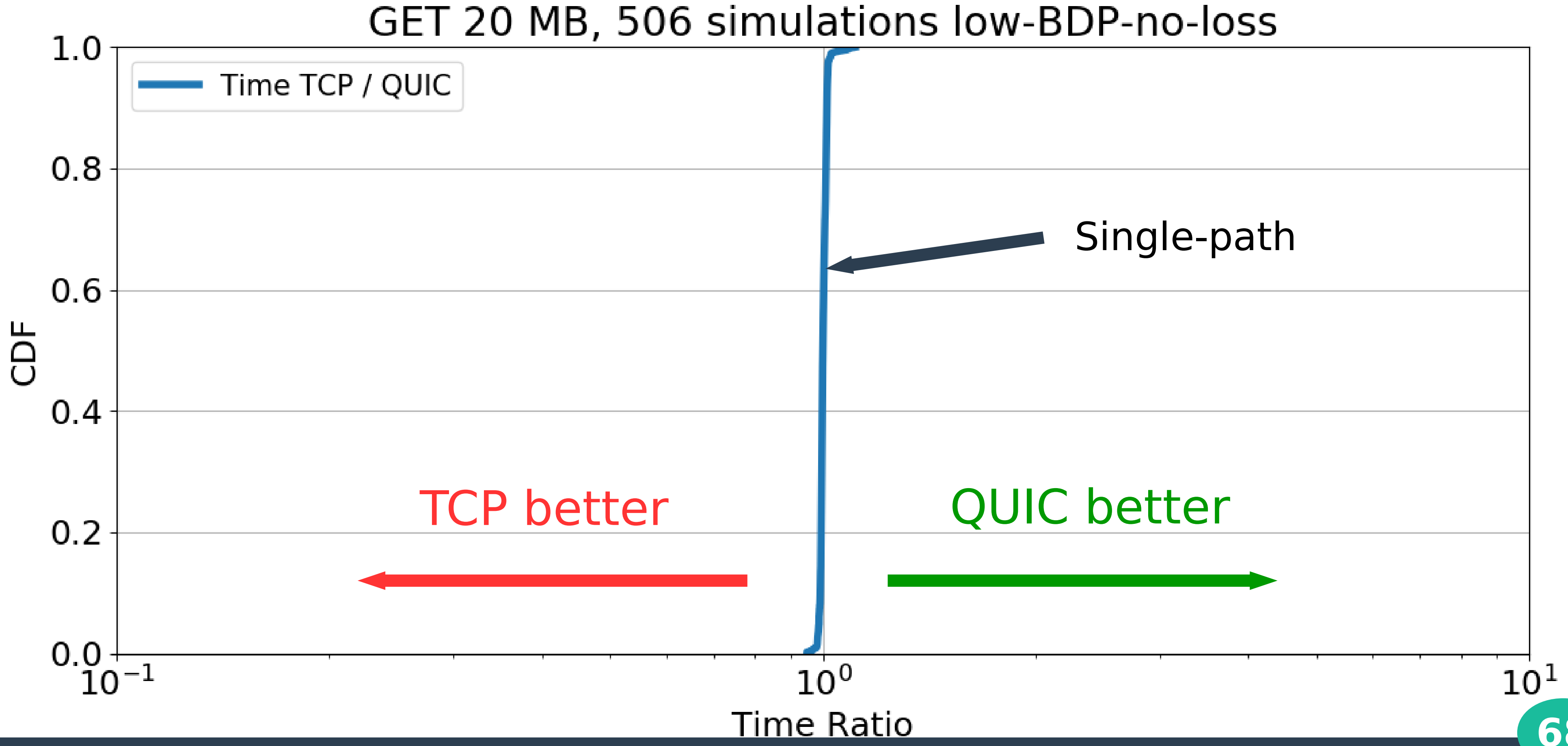


Factor	Minimum	Maximum
Capacity [Mbps]	0.1	100
Round-Trip-Time [ms]	0	50
Queuing Delay [ms]	0	100
Random Loss [%]	0	2.5

Large File Download – No Loss



Large File Download – No Loss

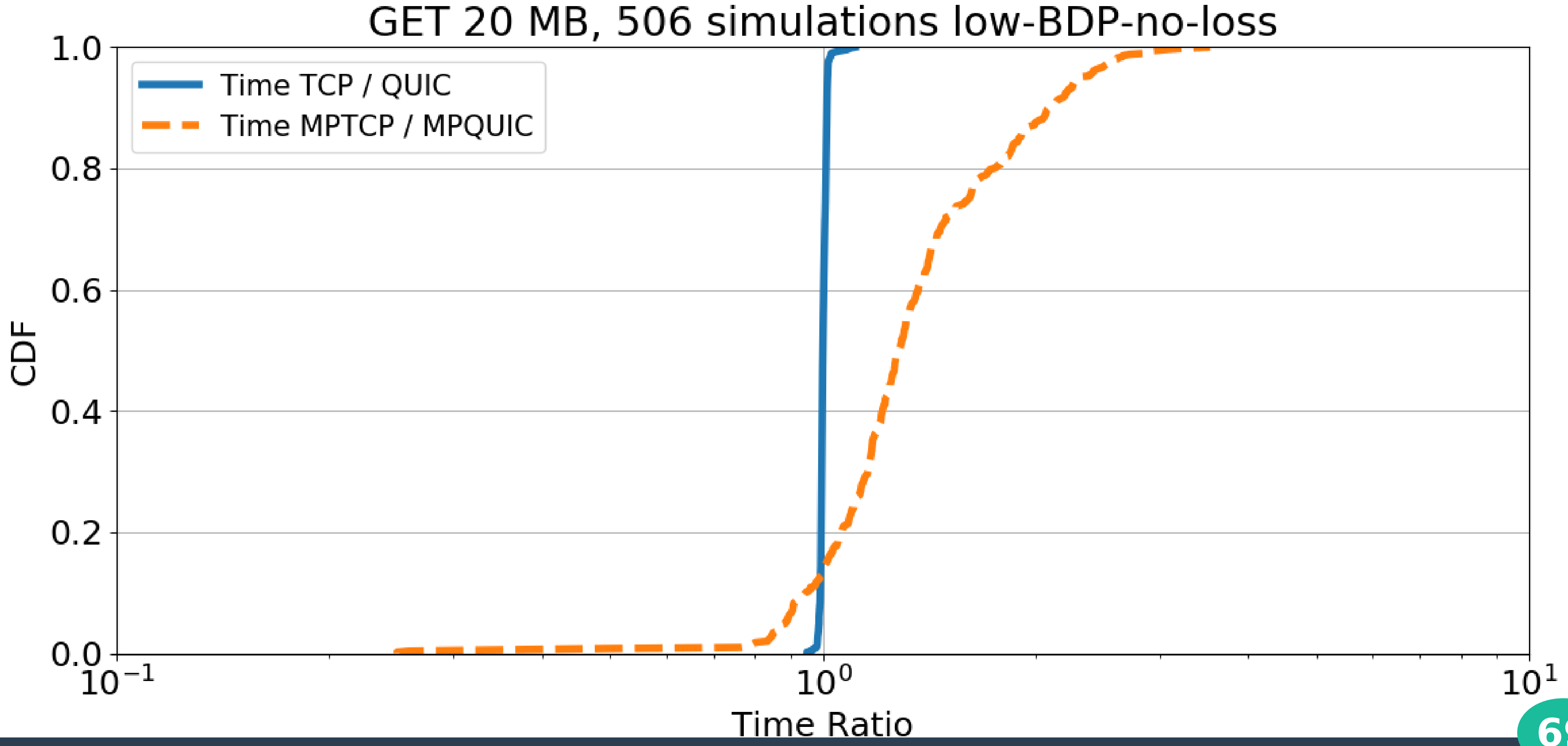


Single-path

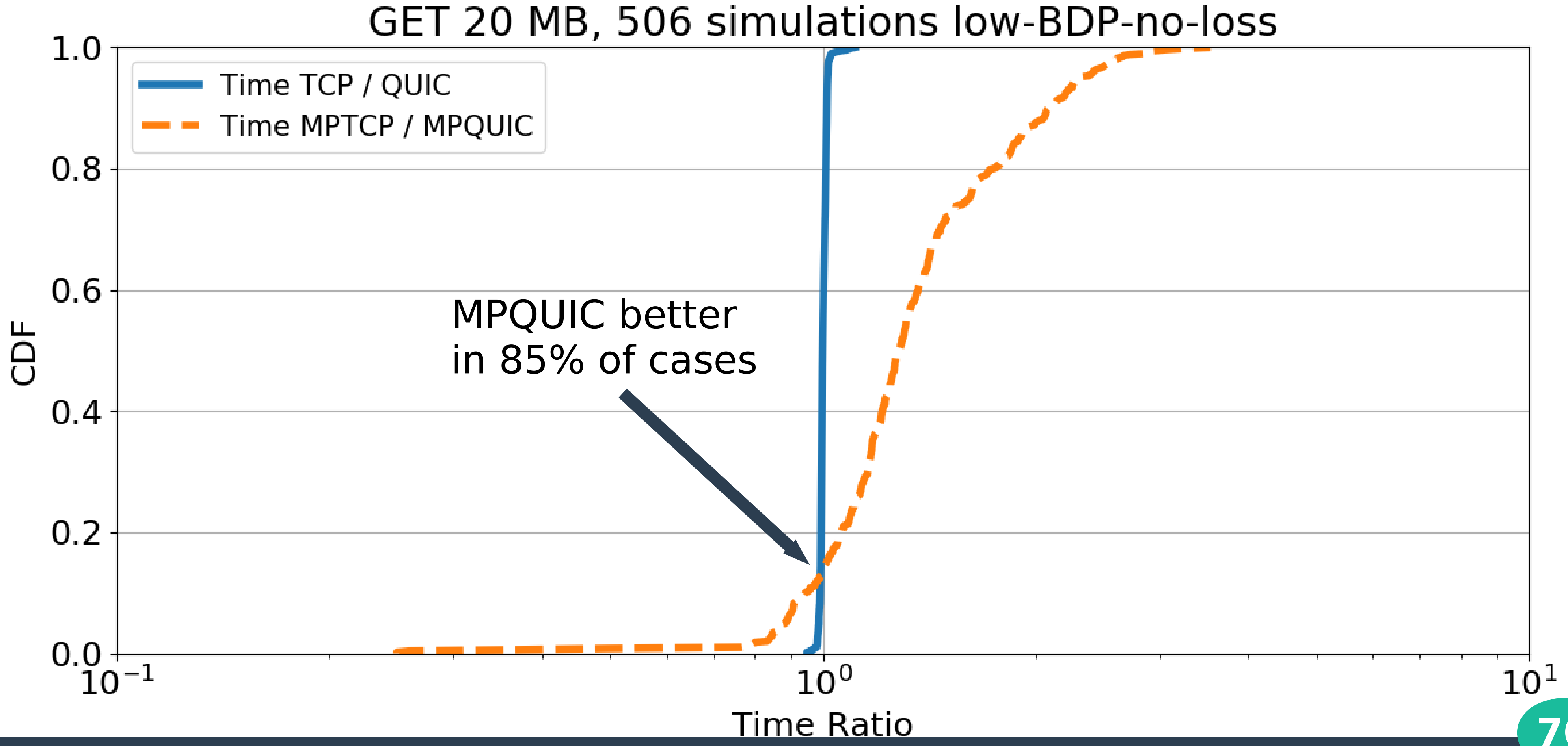
TCP better

QUIC better

Large File Download – No Loss

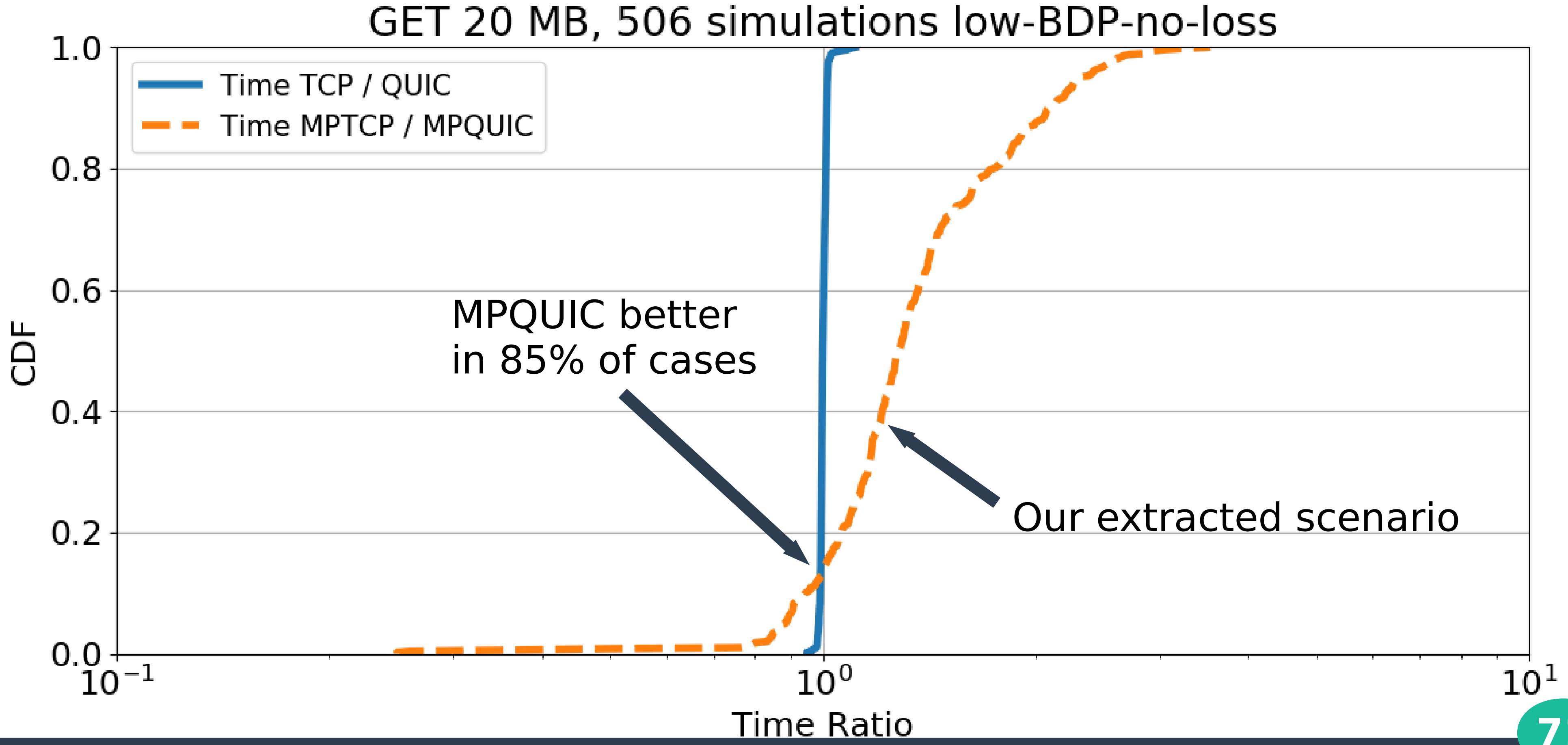


Large File Download – No Loss



MPQUIC better
in 85% of cases

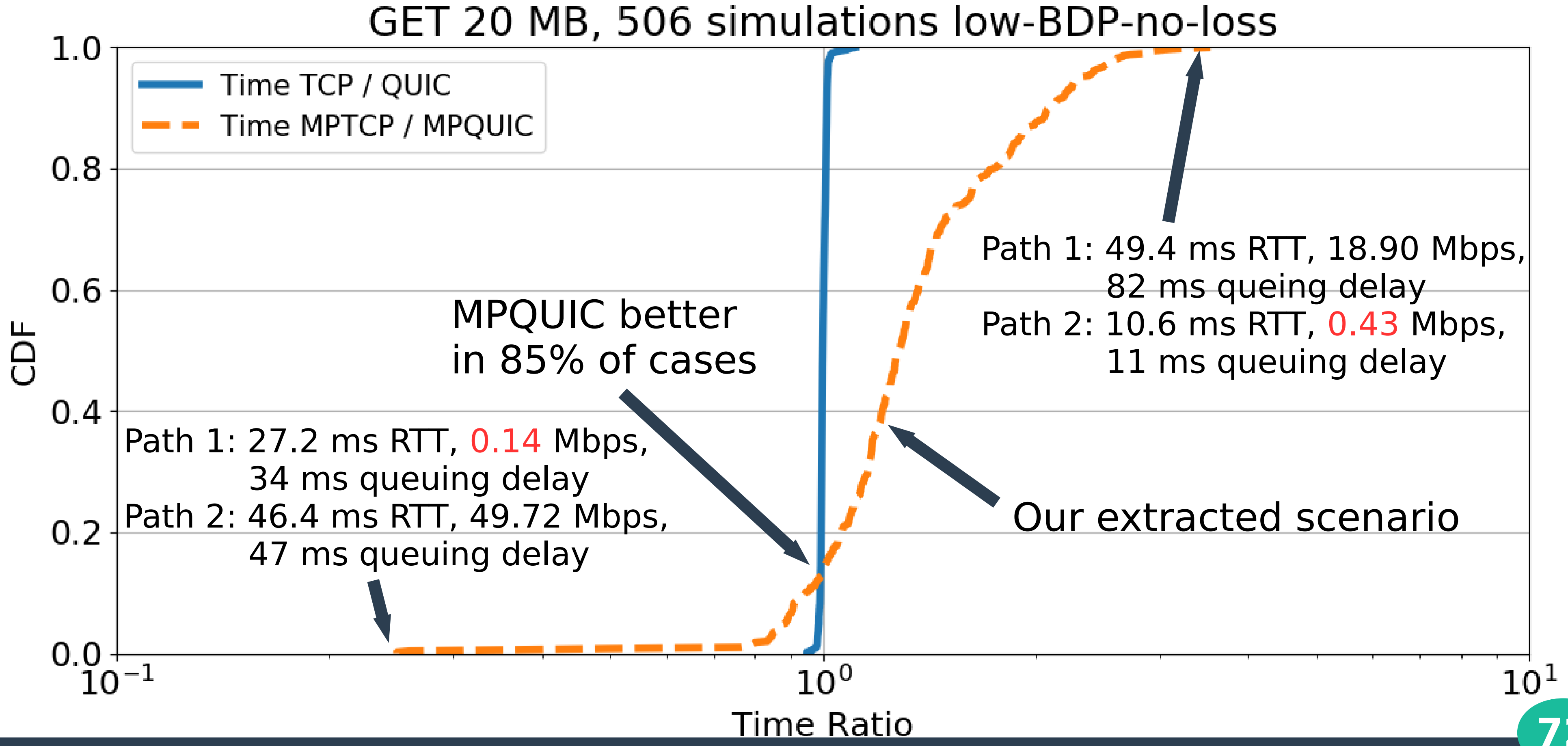
Large File Download – No Loss



MPQUIC better in 85% of cases

Our extracted scenario

Large File Download – No Loss



GET 20 MB, 506 simulations low-BDP-no-loss

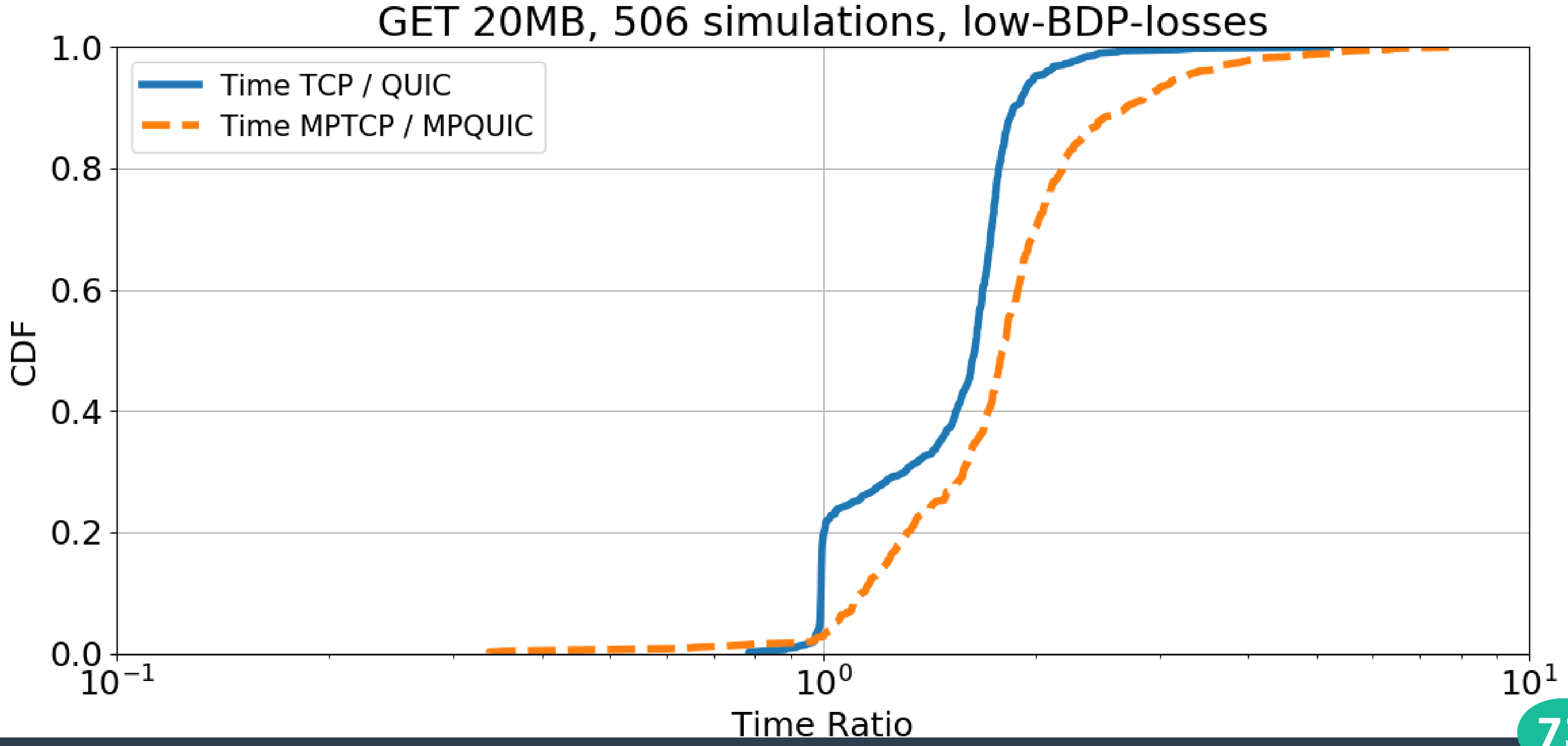
MPQUIC better in 85% of cases

Path 1: 49.4 ms RTT, 18.90 Mbps, 82 ms queing delay
Path 2: 10.6 ms RTT, 0.43 Mbps, 11 ms queuing delay

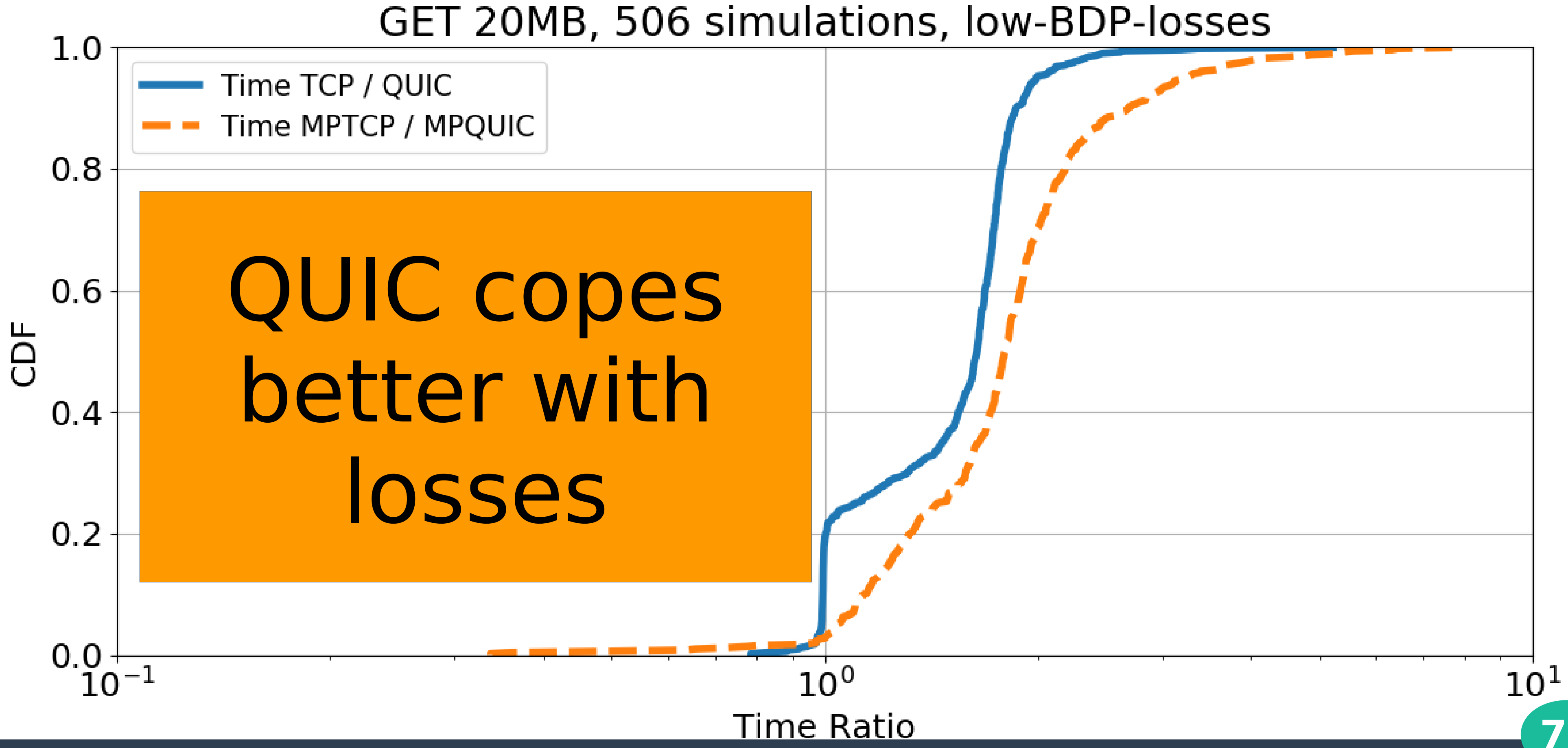
Path 1: 27.2 ms RTT, 0.14 Mbps, 34 ms queuing delay
Path 2: 46.4 ms RTT, 49.72 Mbps, 47 ms queuing delay

Our extracted scenario

Large File Download – Losses

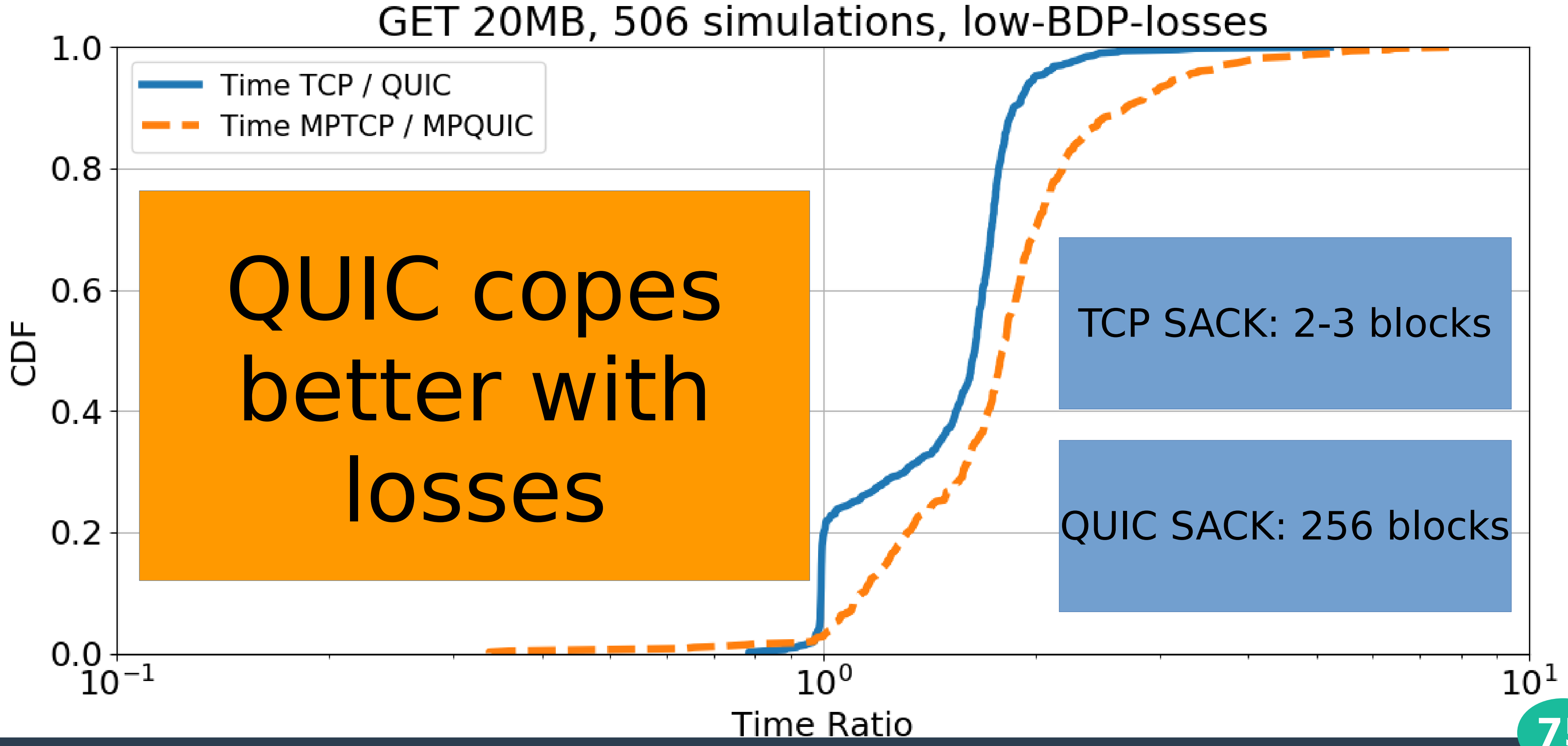


Large File Download – Losses



QUIC copes better with losses

Large File Download – Losses



QUIC copes better with losses

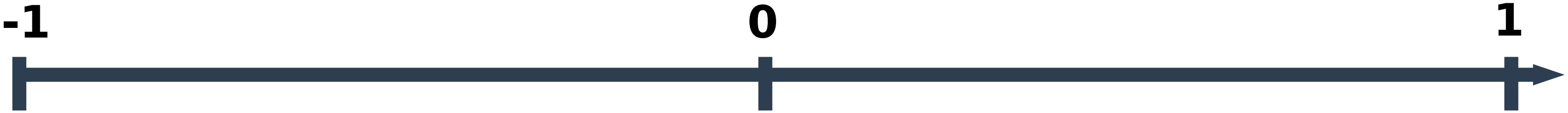
TCP SACK: 2-3 blocks

QUIC SACK: 256 blocks

What is the actual benefit of Multipath to QUIC?

Actual Multipath Benefit

- **Experimental Aggregation Benefit**
  - Multipath QUIC/TCP vs. single-path QUIC/TCP



Zero goodput

MP gives 0 Mbps

= best single path

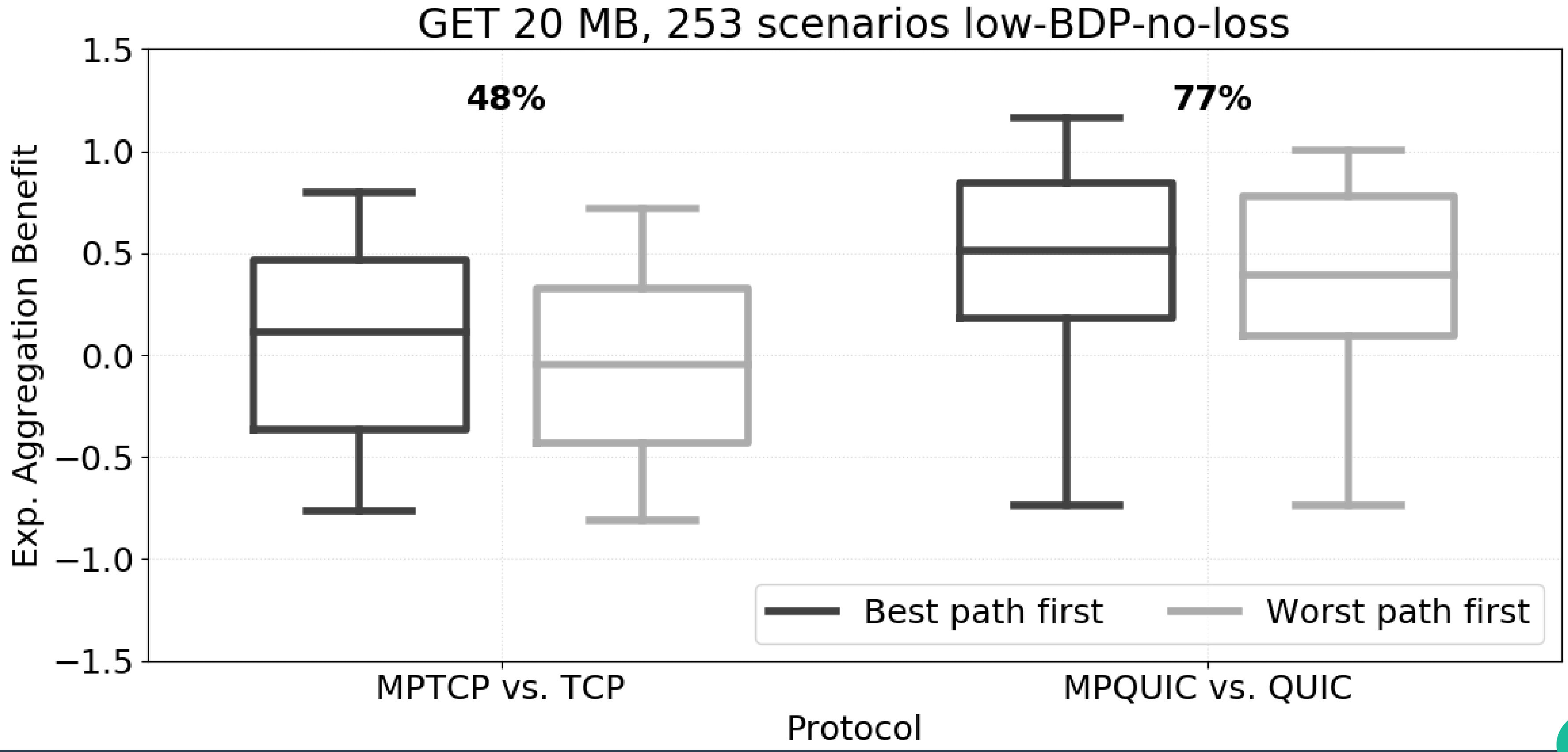
3 Mbps + 5 Mbps paths
MP gives 5 Mbps

= aggregation of all paths

3 Mbps + 5 Mbps paths
MP gives 8 Mbps

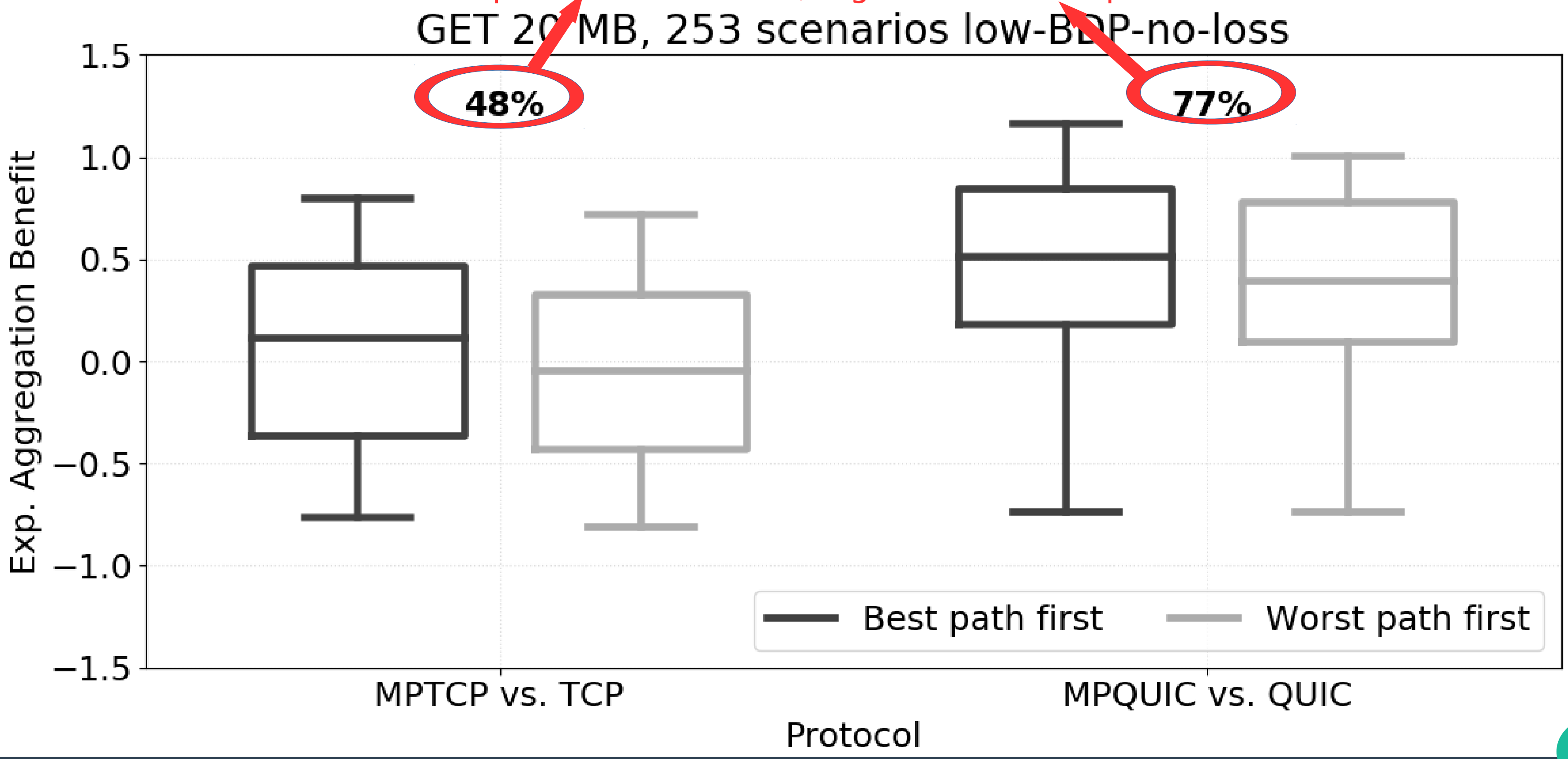
- **Results depends on the first path used**
  - Handshake latency over initial path

Benefits of Multipath – No Loss

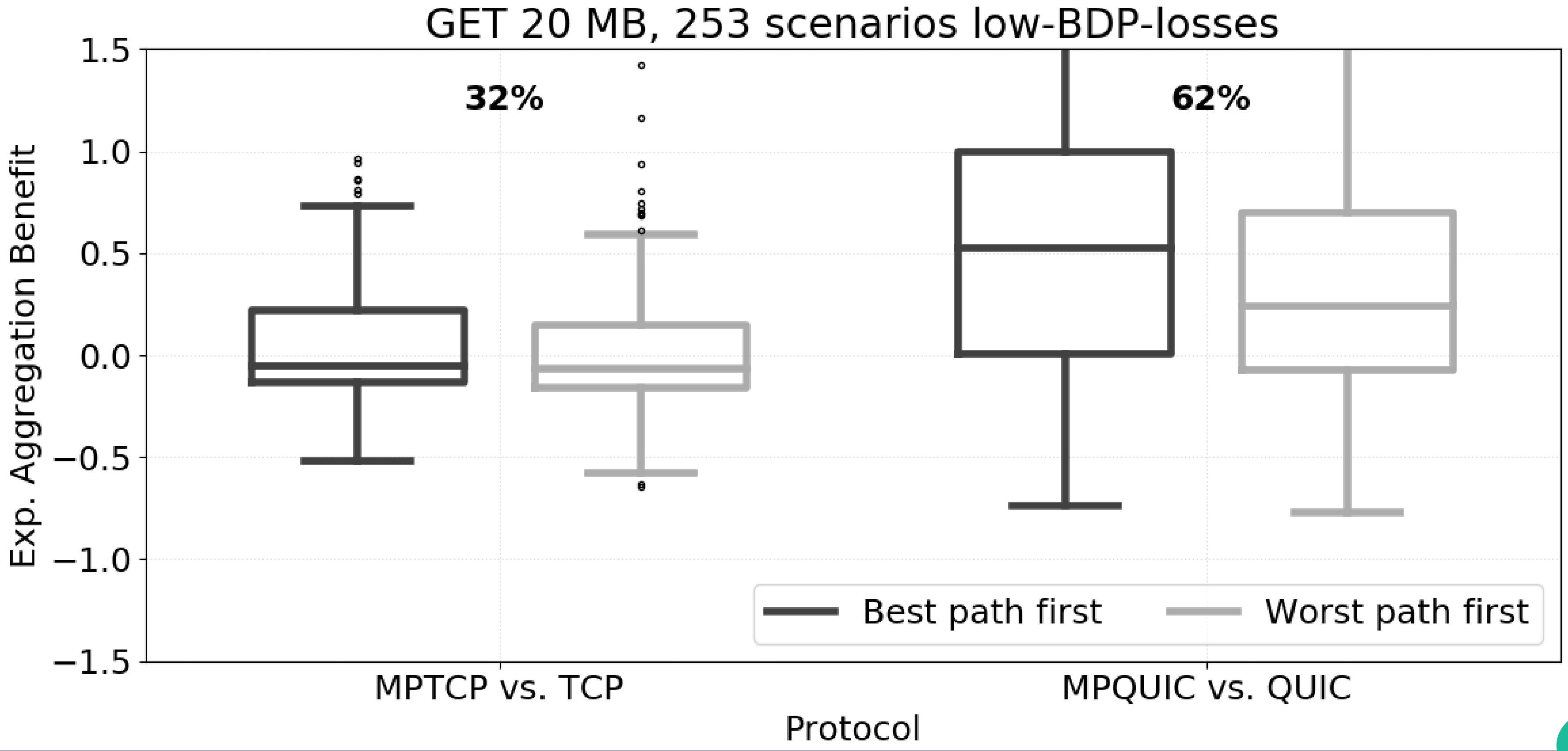


Benefits of Multipath – No Loss

% scenarios multipath has EAB >= 0, regardless of first path used



Benefits of Multipath - Losses

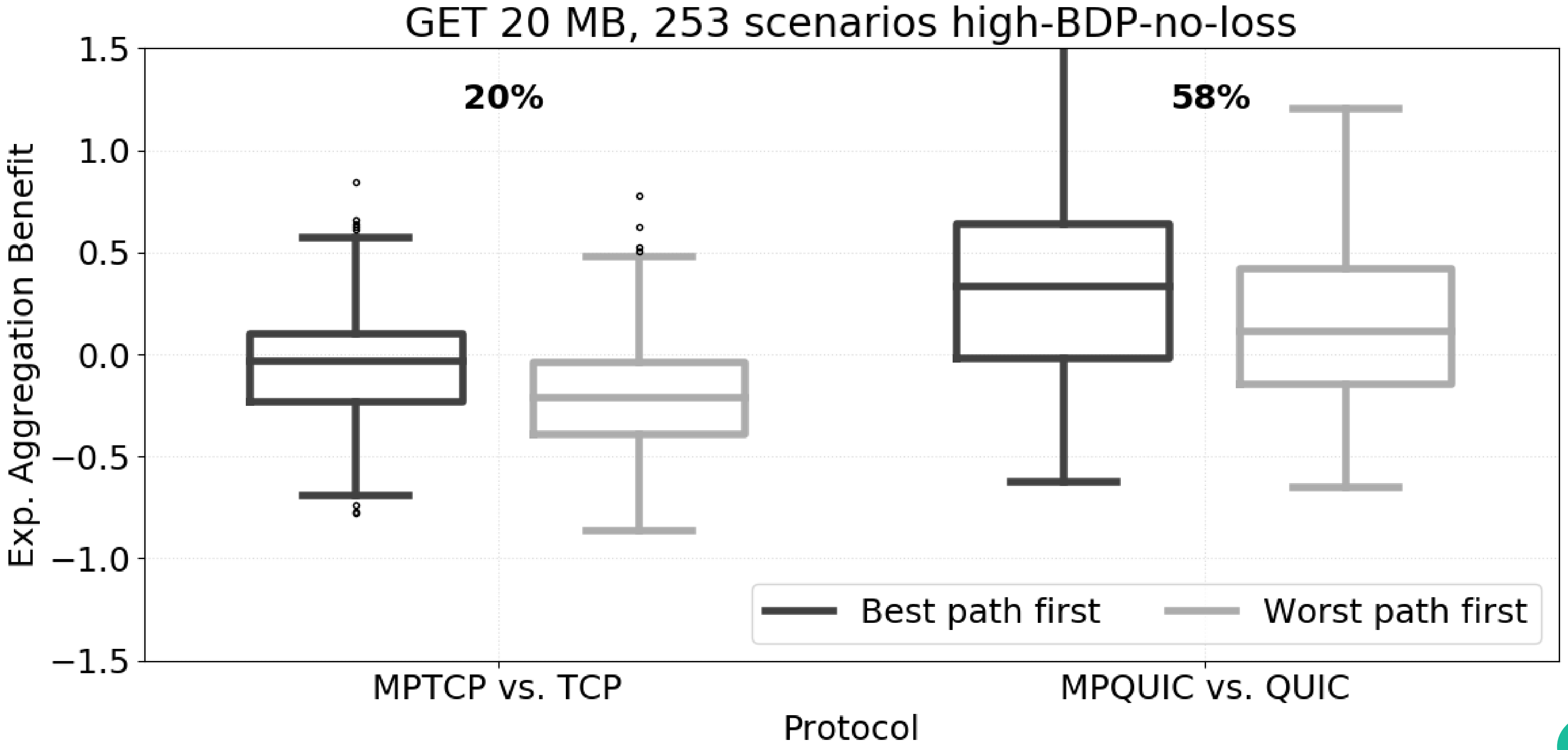


What about congestion-prone networks?

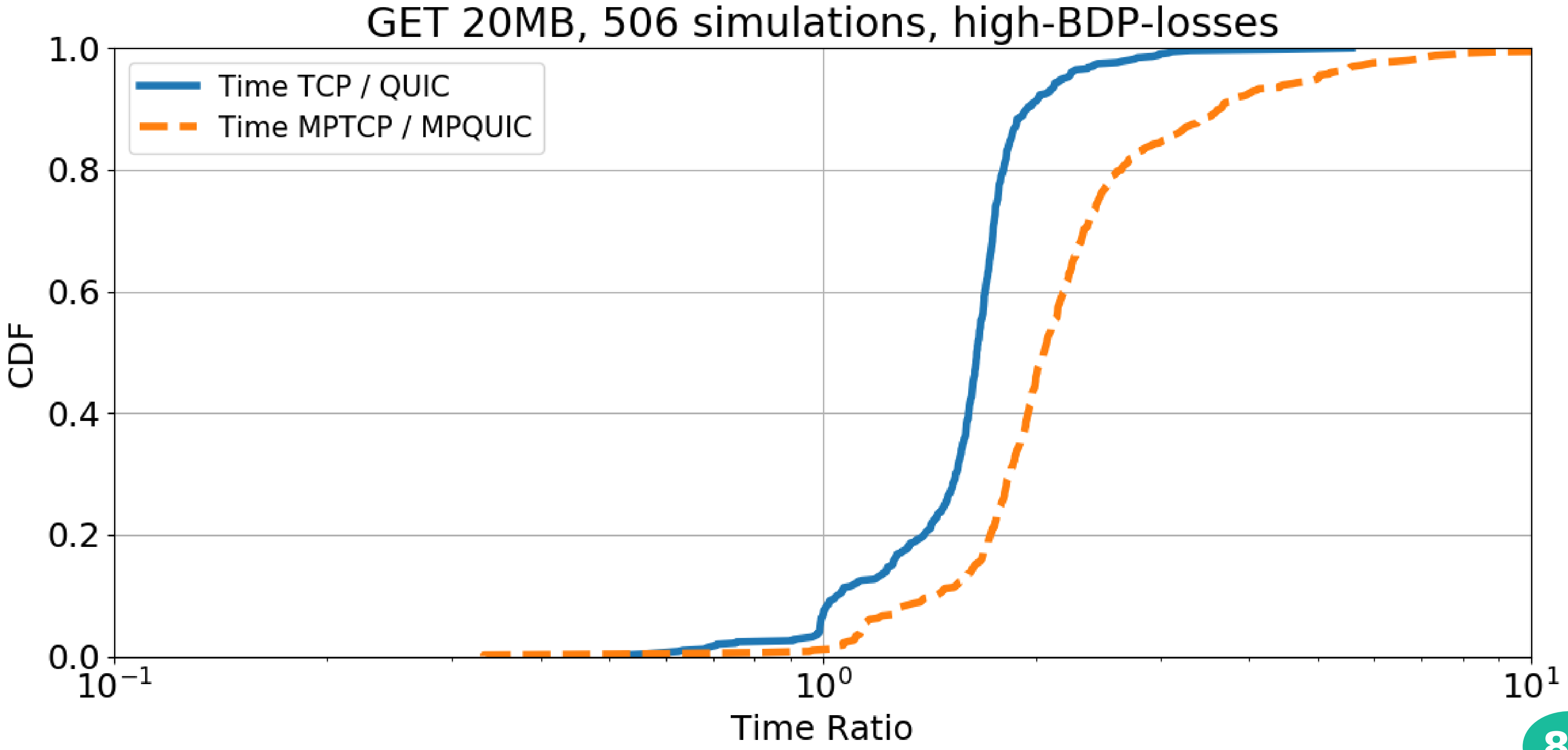
Experimental Design with High-BDP Networks

Factor	Minimum	Maximum
Capacity [Mbps]	0.1	100
Round-Trip-Time [ms]	0	**400**
Queuing Delay [ms]	0	**2000**
Random Loss [%]	0	2.5

Multipath Benefits without Losses



Completion Time Ratio with Losses



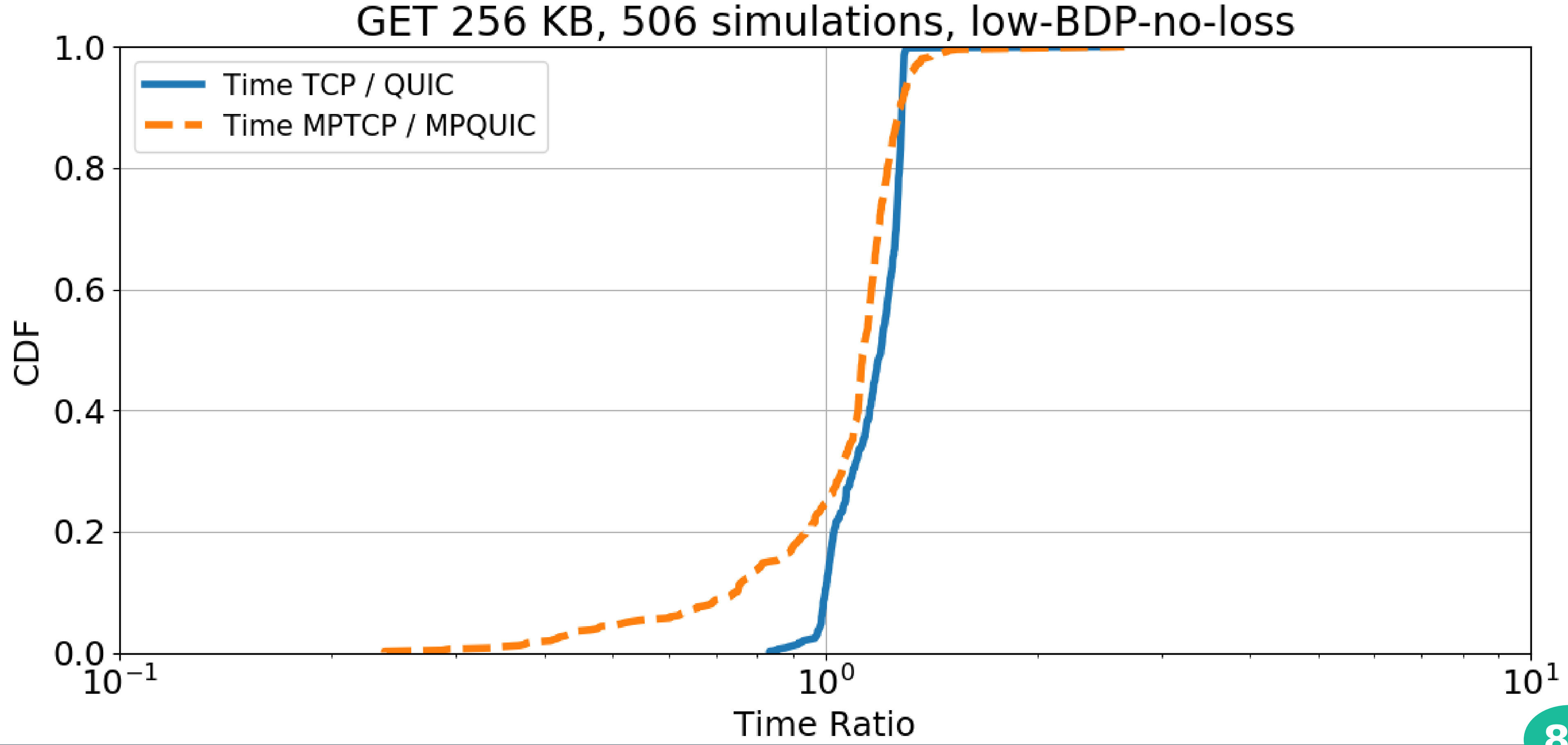
What about short transfers?

Short Transfer Evaluation with Low-BDP

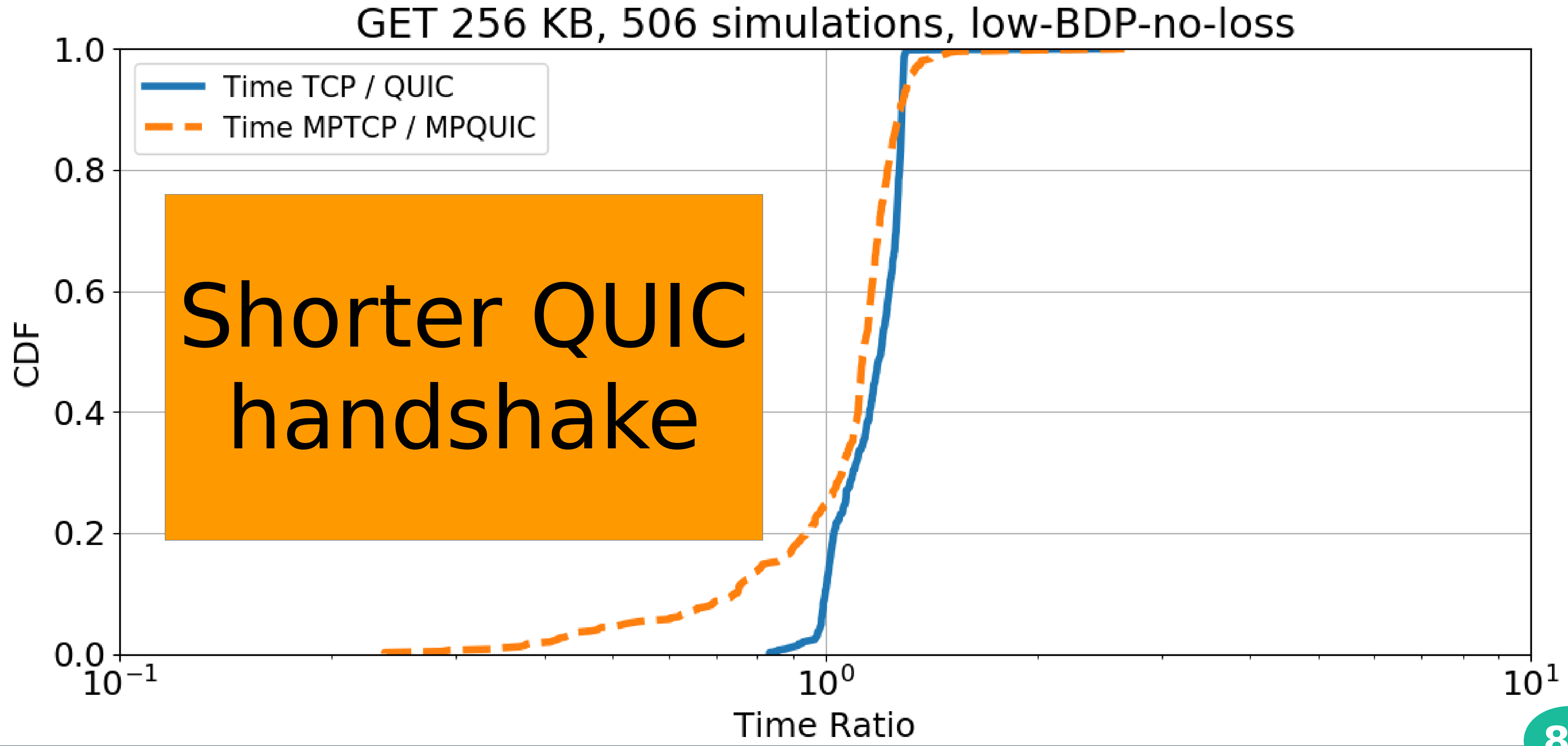
- **Download of a 256 KB file**
  - Collect transfer time
- **Median over 3 runs**

Factor	Minimum	Maximum
Capacity [Mbps]	0.1	100
Round-Trip-Time [ms]	0	50
Queuing Delay [ms]	0	100

Comparison QUIC vs. TCP

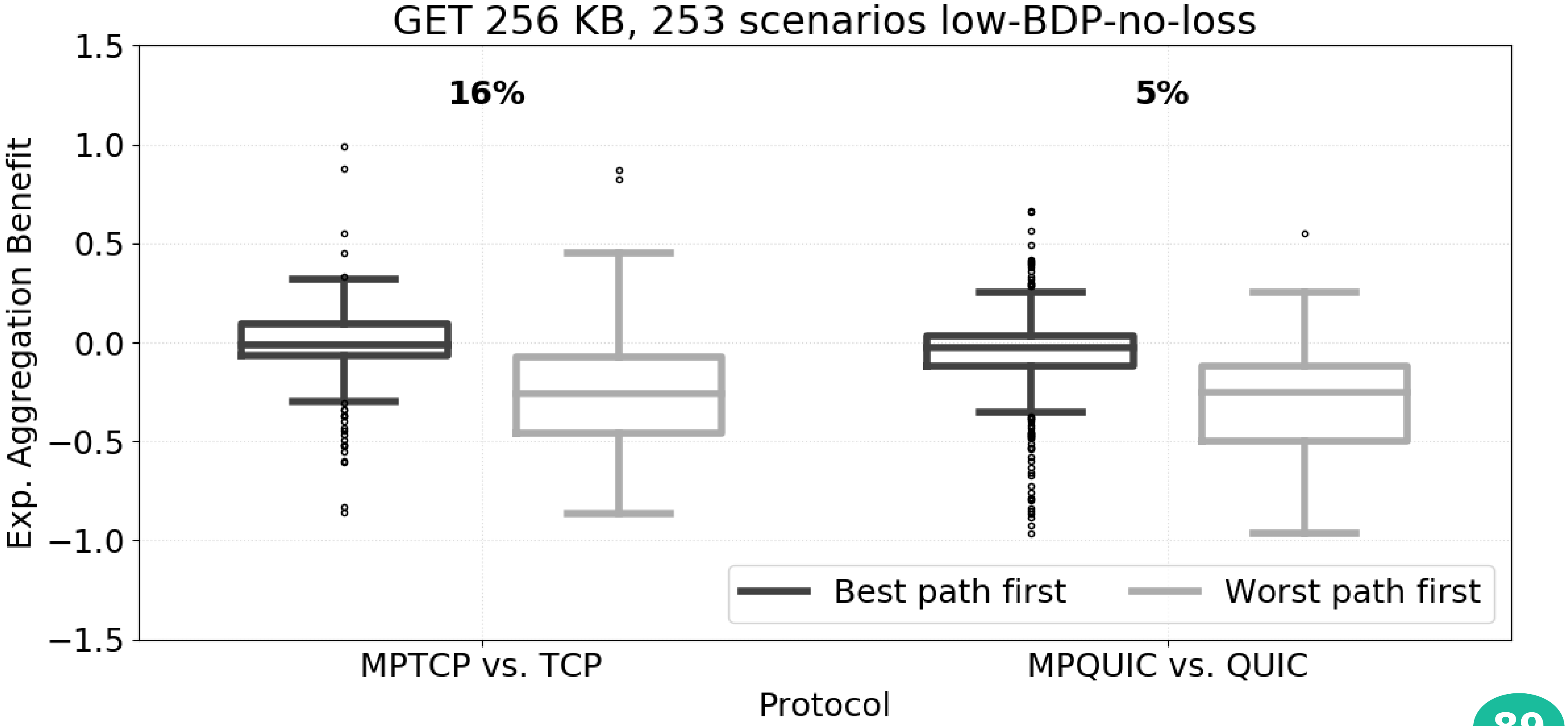


Comparison QUIC vs. TCP



Shorter QUIC handshake

Multipath Not Really Useful...



What about network handover?

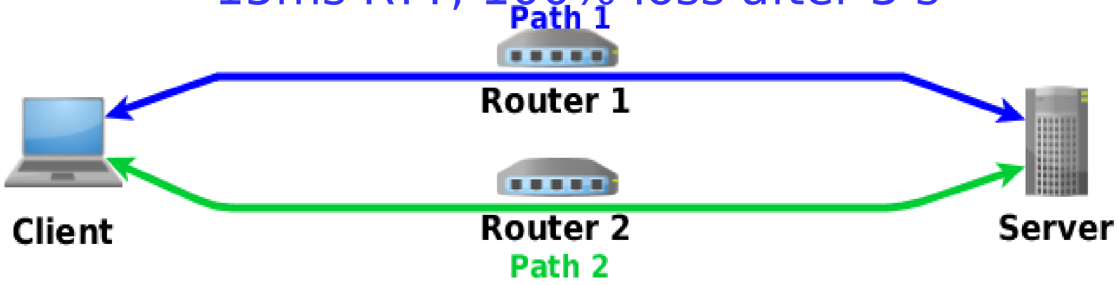
Network Handover Support

- **Apple MPTCP deployment mainly for handover**
  - Main use case for Siri

Network Handover Support

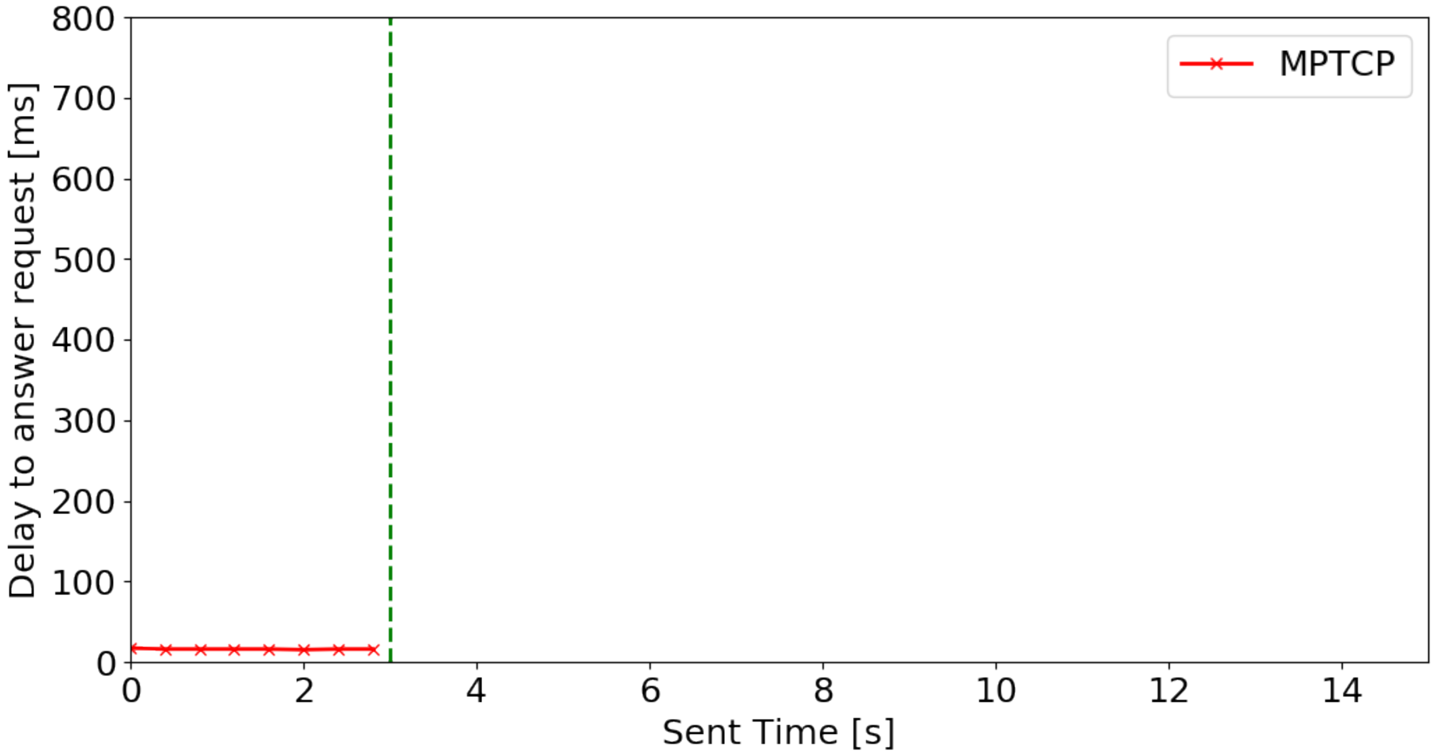
- **Apple MPTCP deployment mainly for handover**
  - Main use case for Siri
- **Request/Response traffic**
  - 750 bytes request/responses
  - Measure delay seen by client

15ms RTT, 100% loss after 3 s

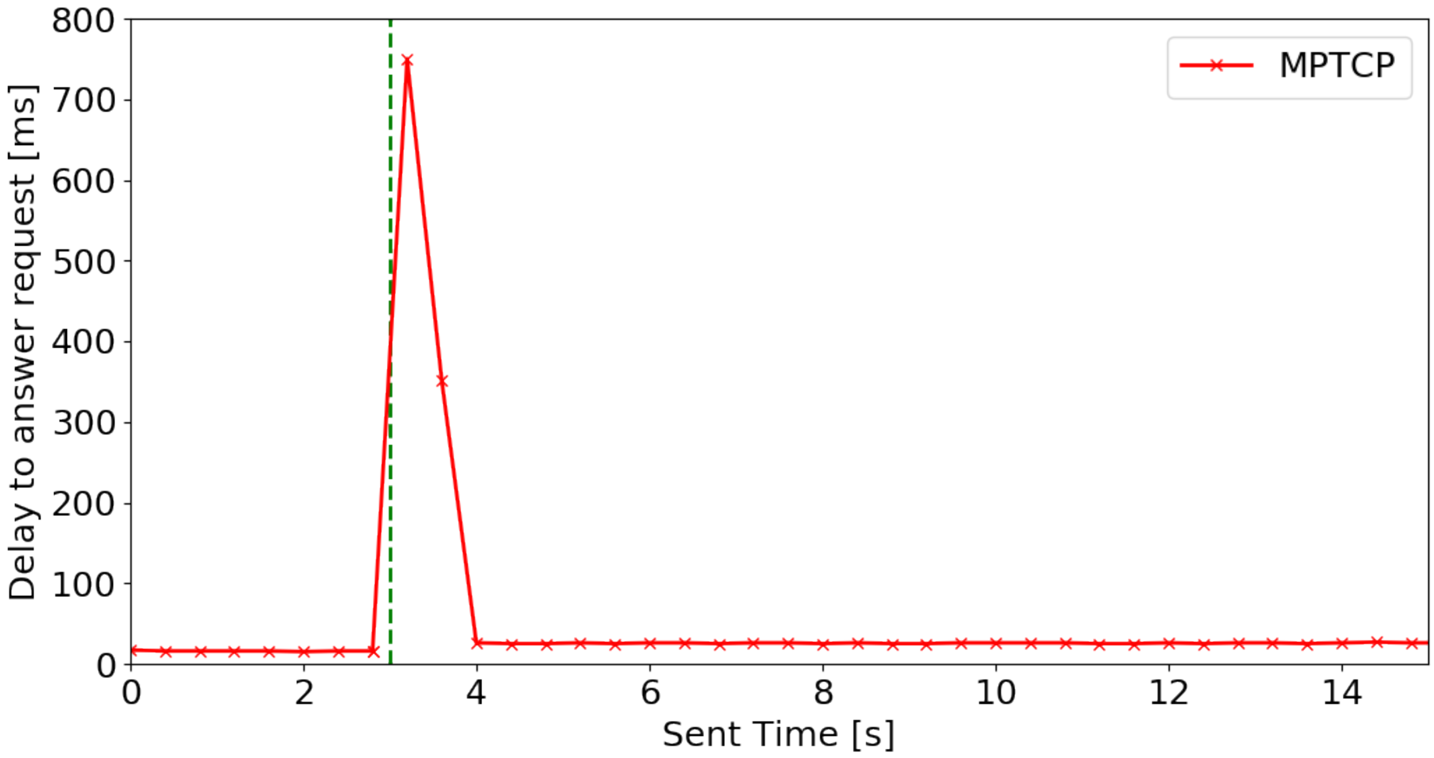


25ms RTT

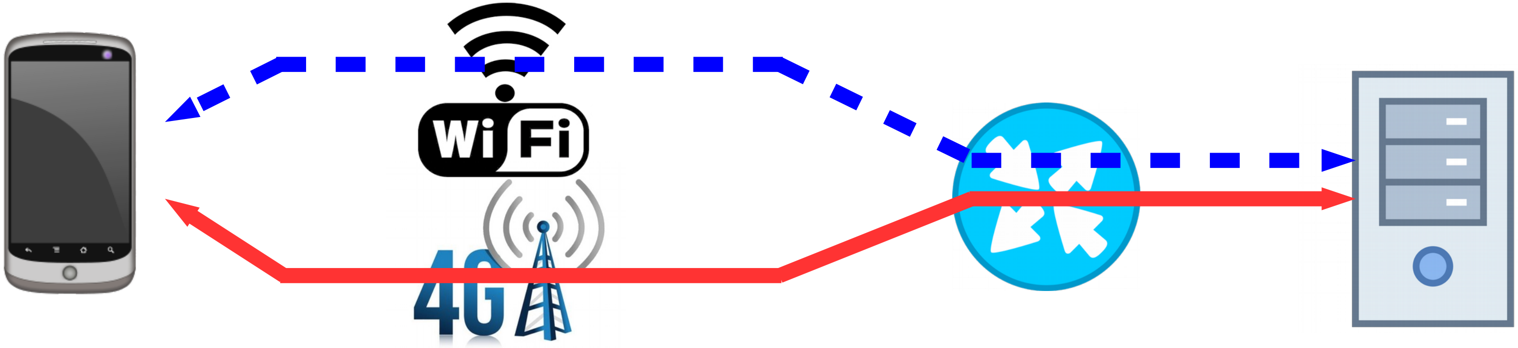
Multipath TCP Handover



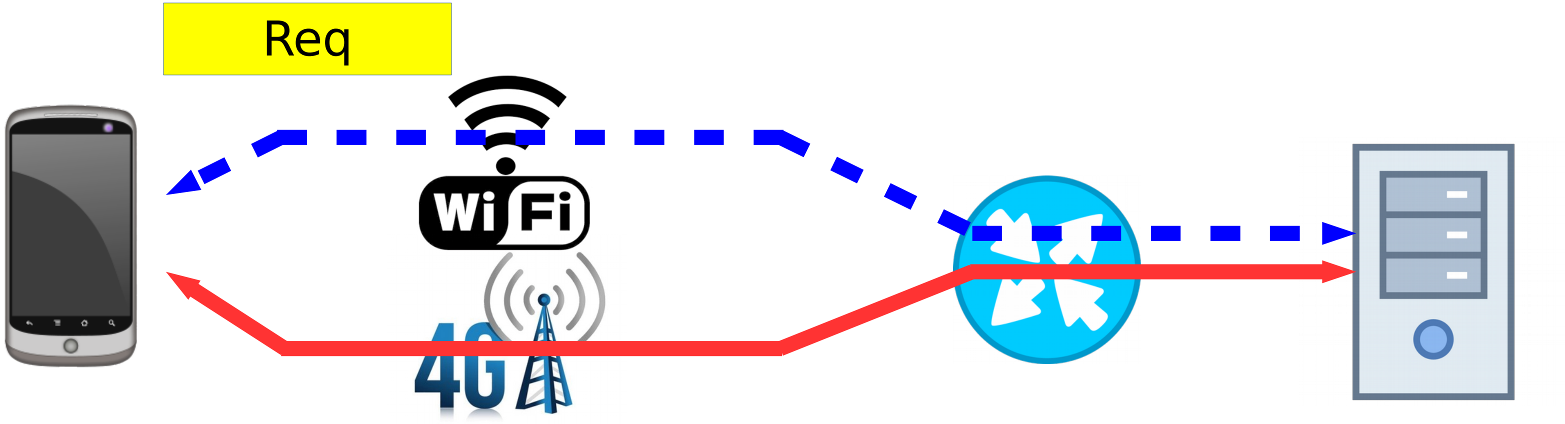
Multipath TCP Handover



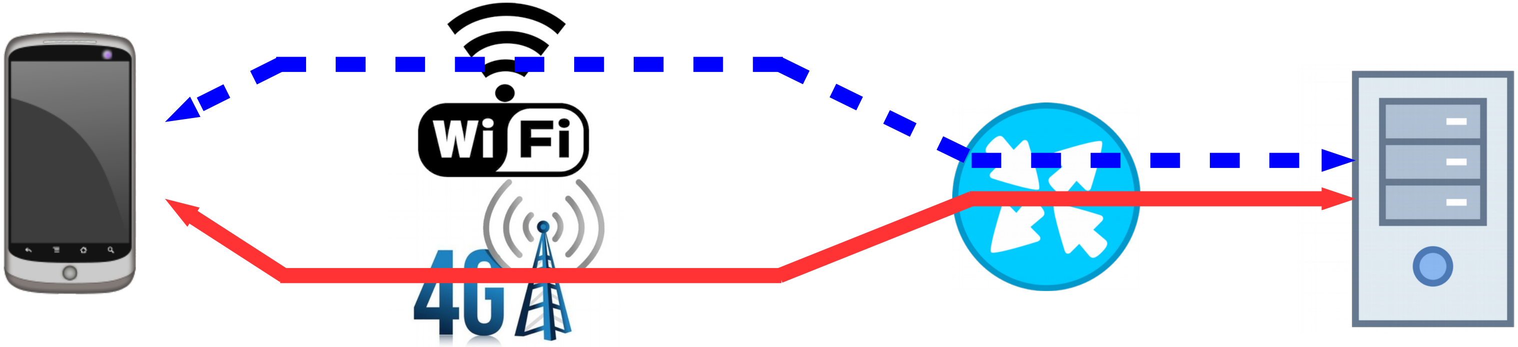
What Happened During MPTCP Handover?



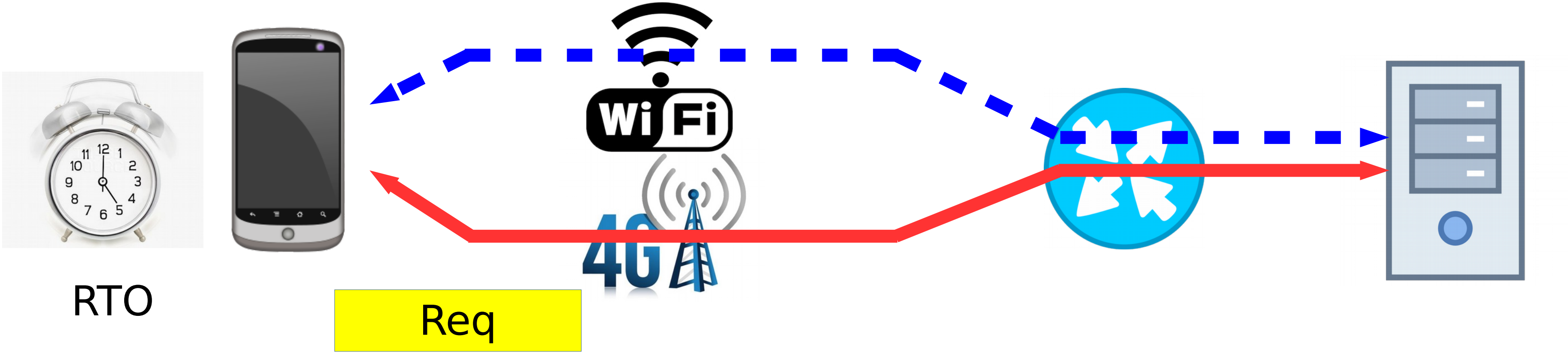
What Happened During MPTCP Handover?



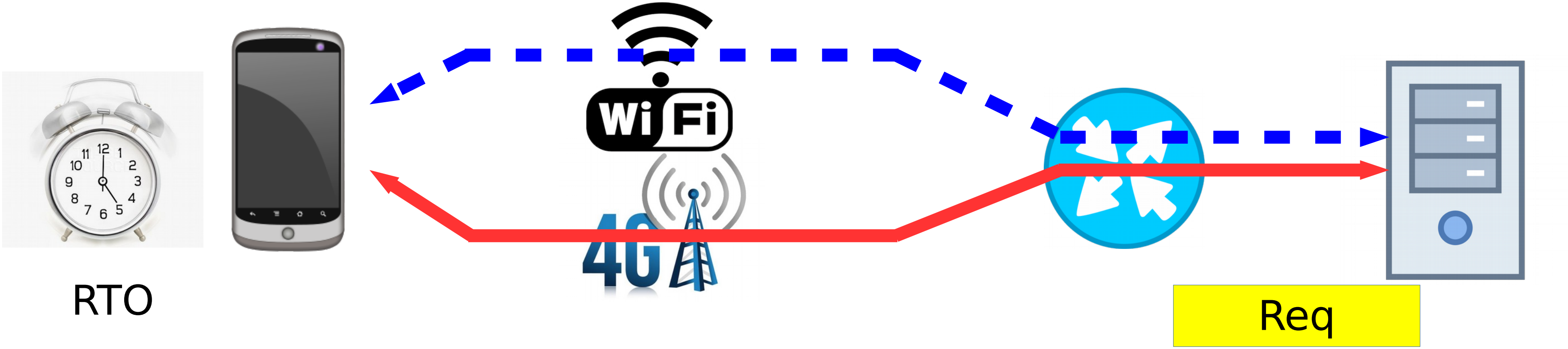
What Happened During MPTCP Handover?



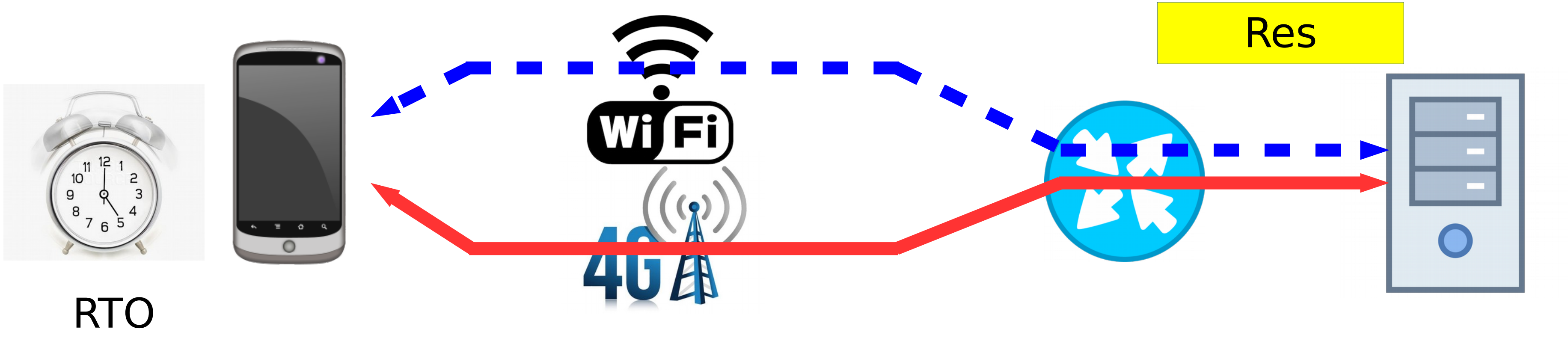
What Happened During MPTCP Handover?



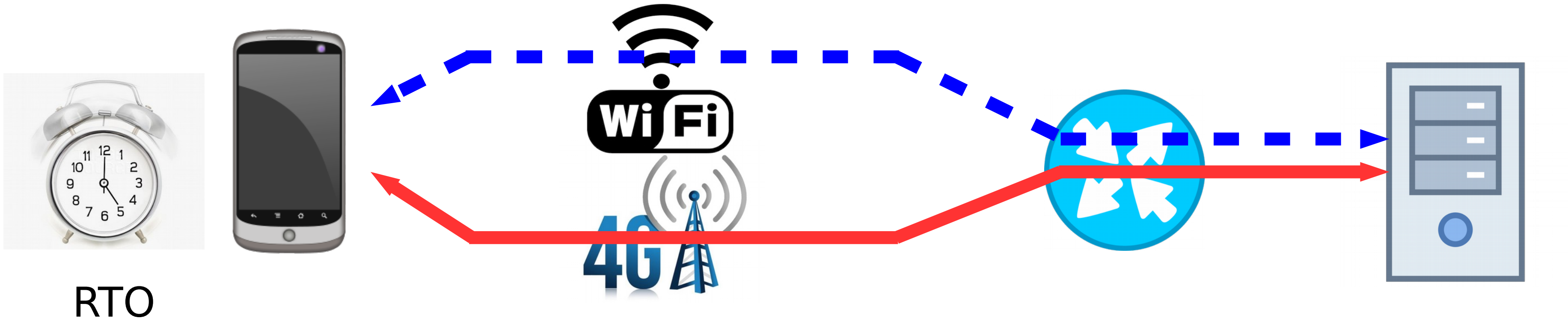
What Happened During MPTCP Handover?



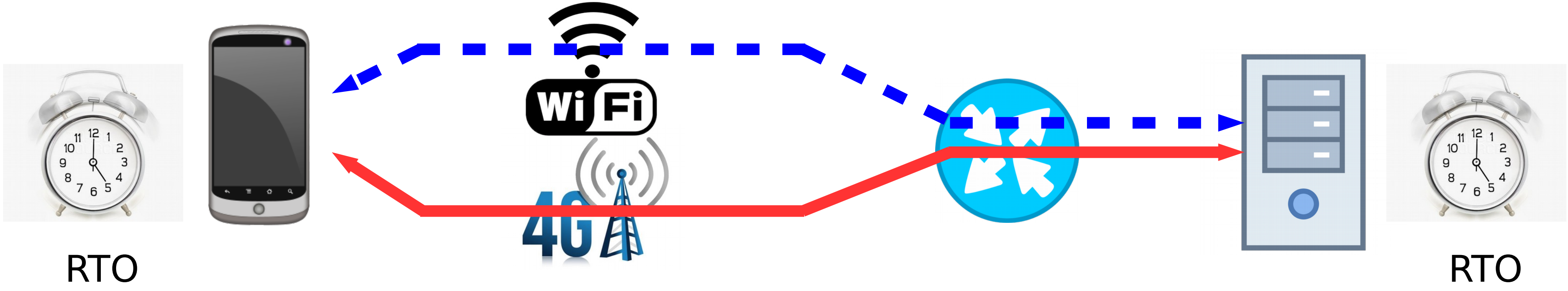
What Happened During MPTCP Handover?



What Happened During MPTCP Handover?

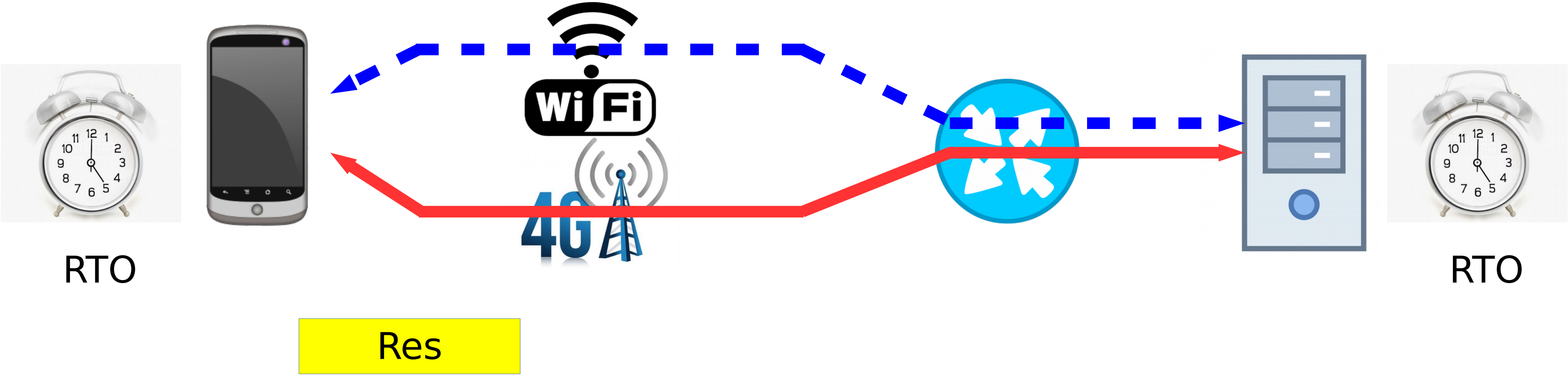


What Happened During MPTCP Handover?

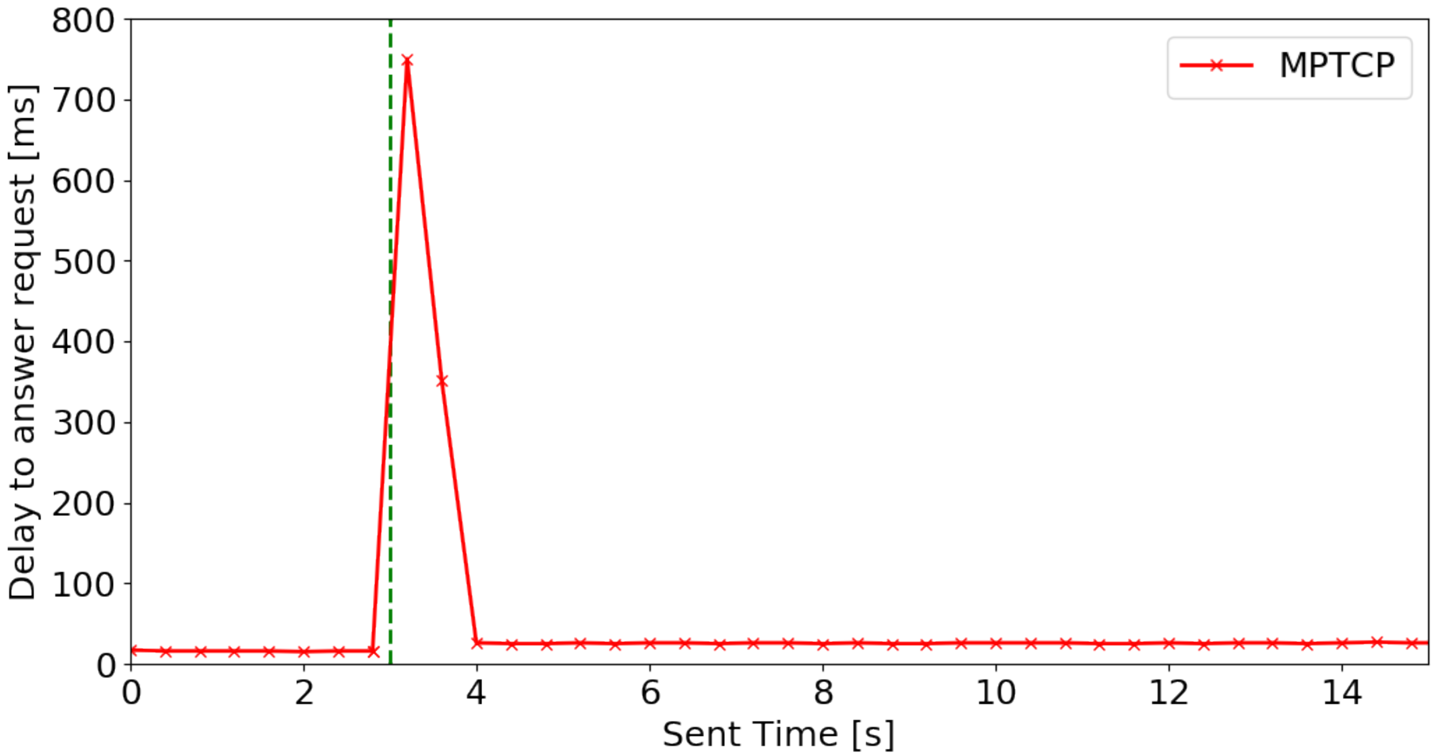


Res

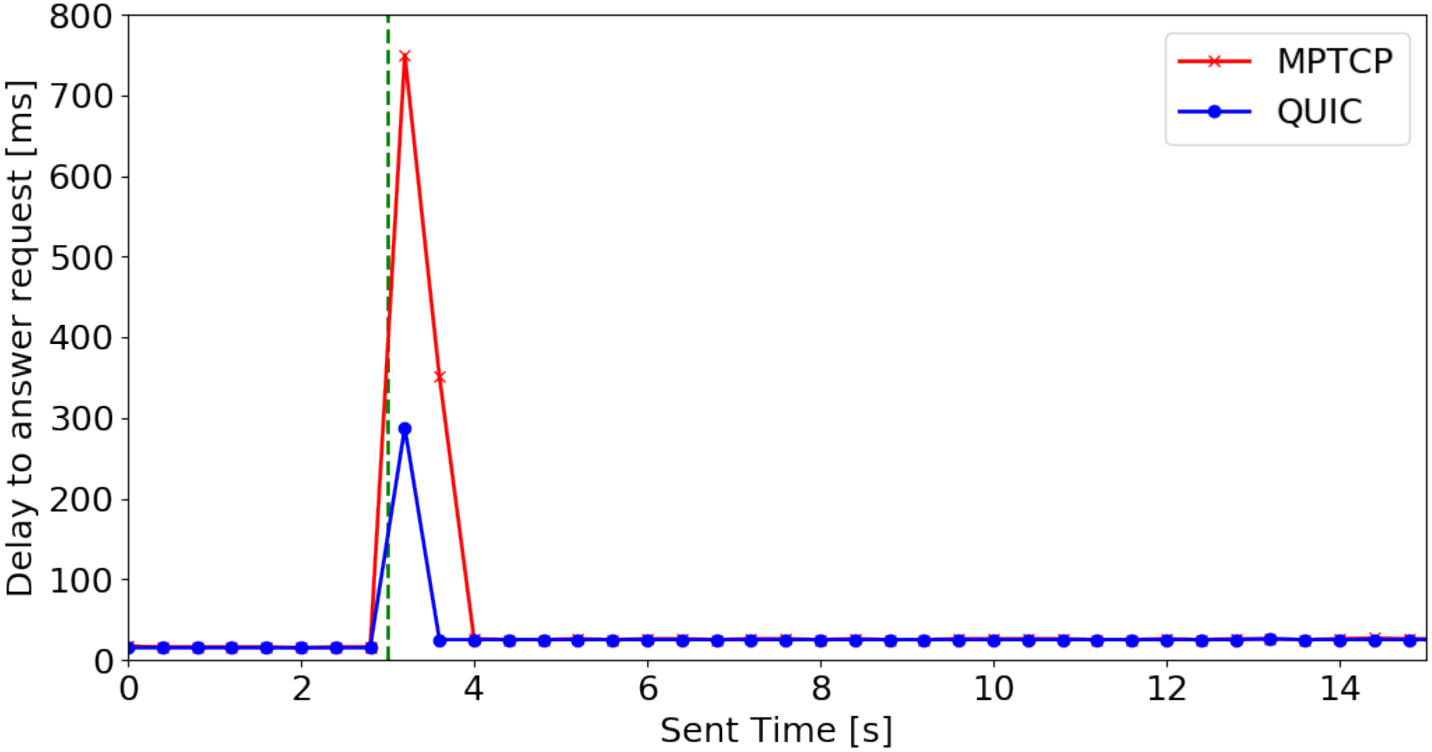
What Happened During MPTCP Handover?



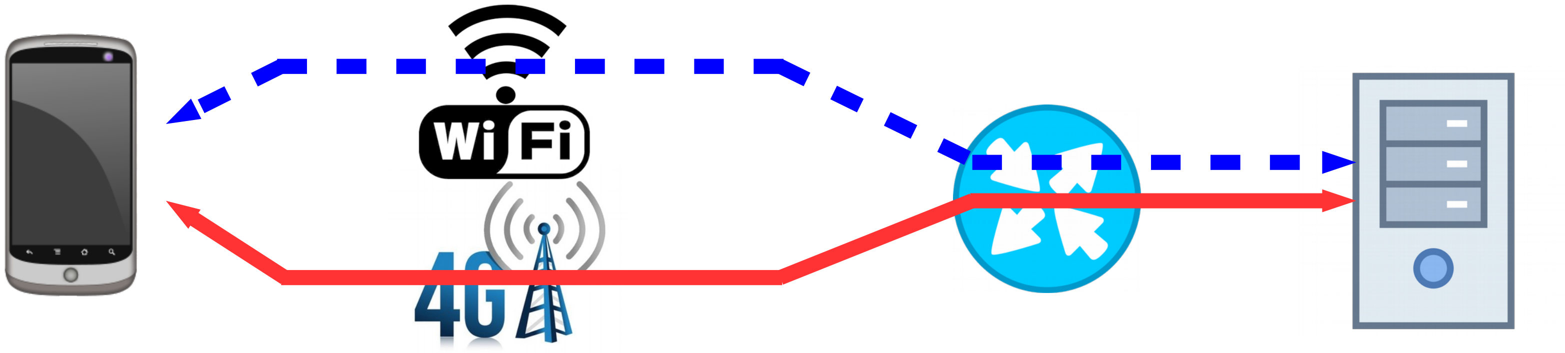
And What About Multipath QUIC?



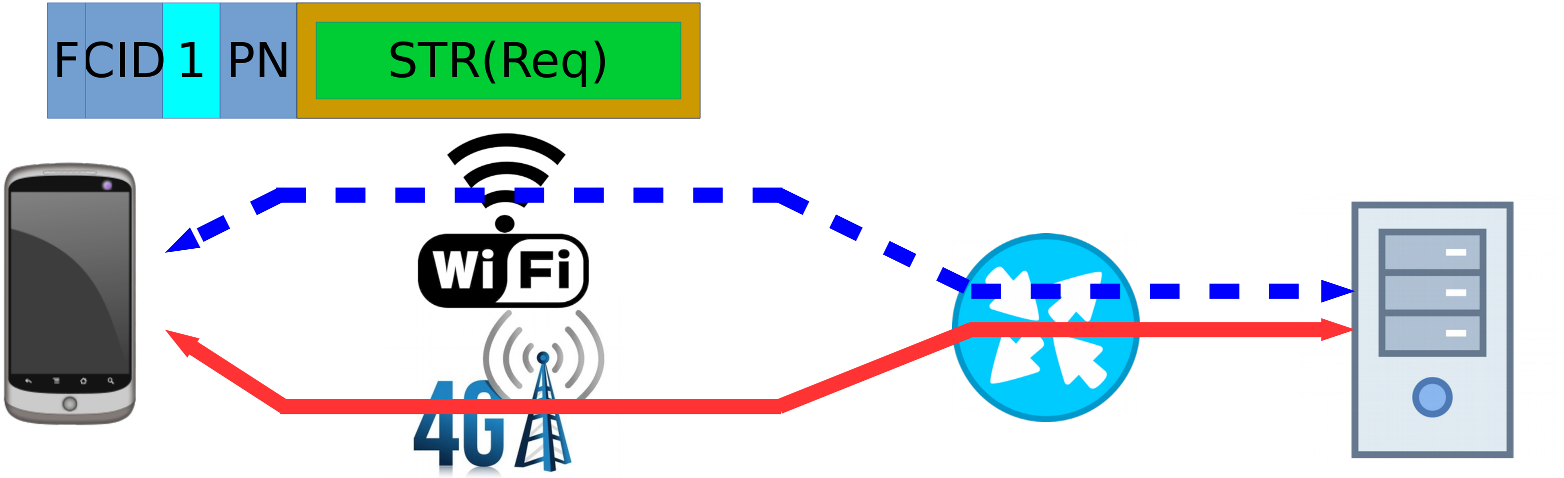
And What About Multipath QUIC?



What Happened During Handover?

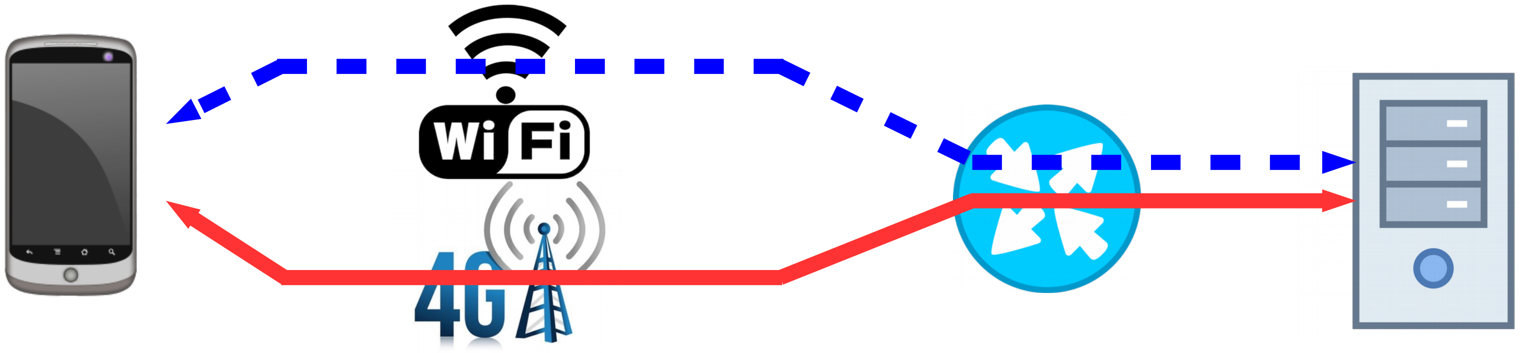


What Happened During Handover?

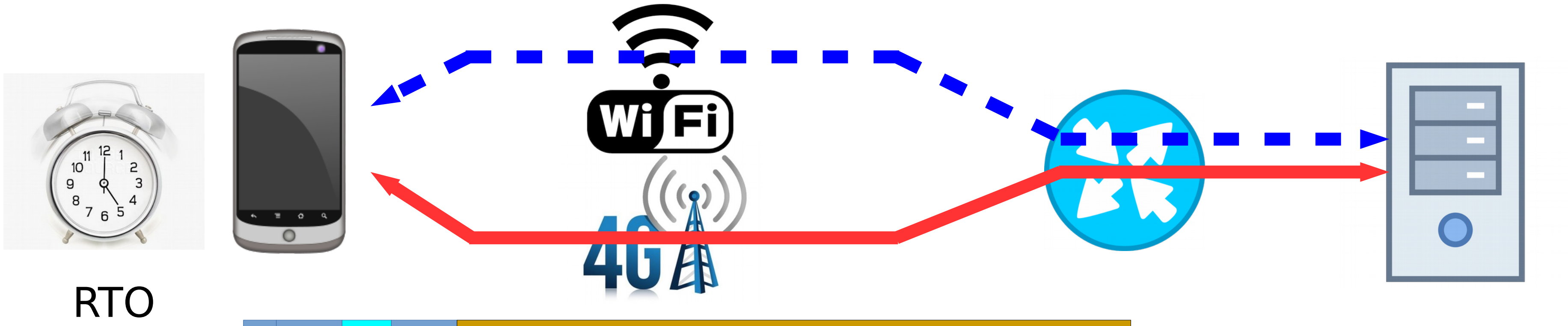


F	CID	1	PN	STR(Req)

What Happened During Handover?

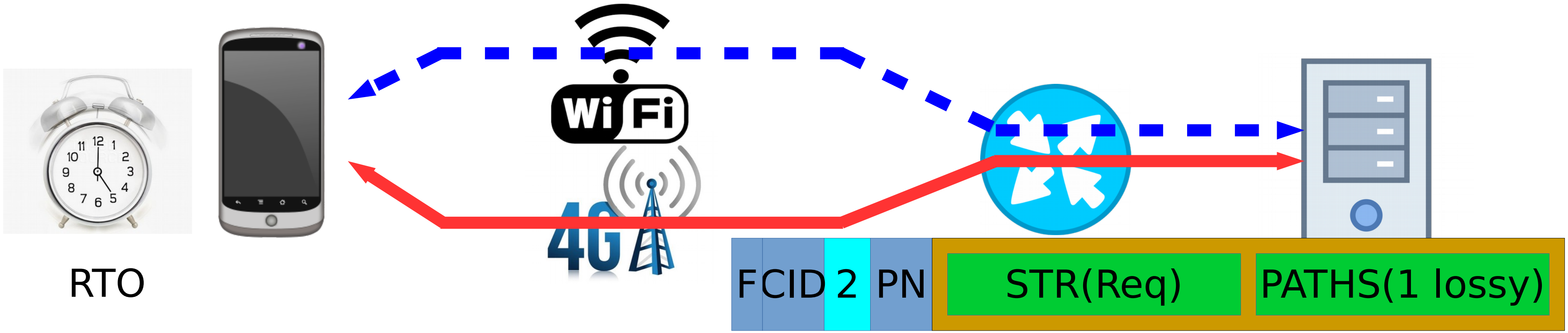


What Happened During Handover?



F	CID	2	PN	STR(Req)	PATHS(1 lossy)

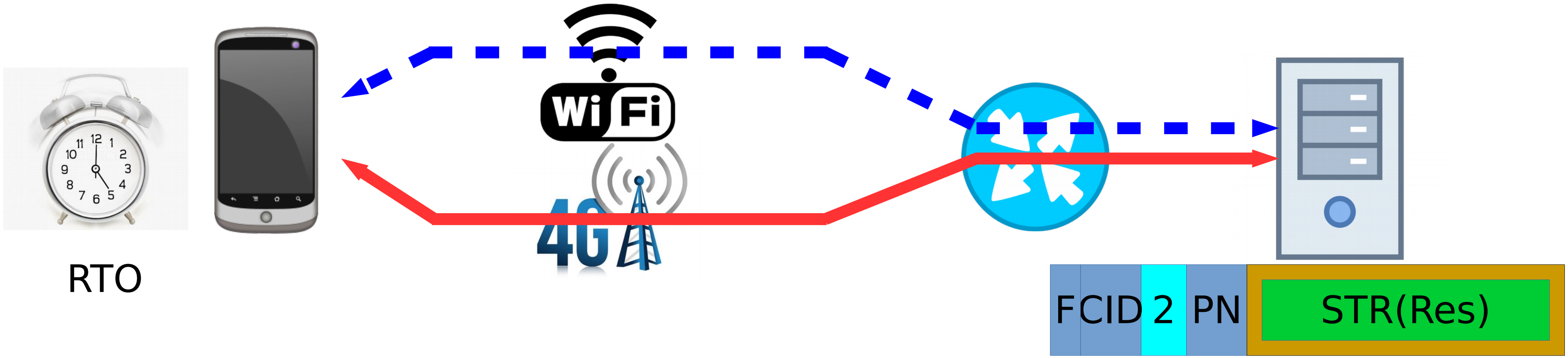
What Happened During Handover?



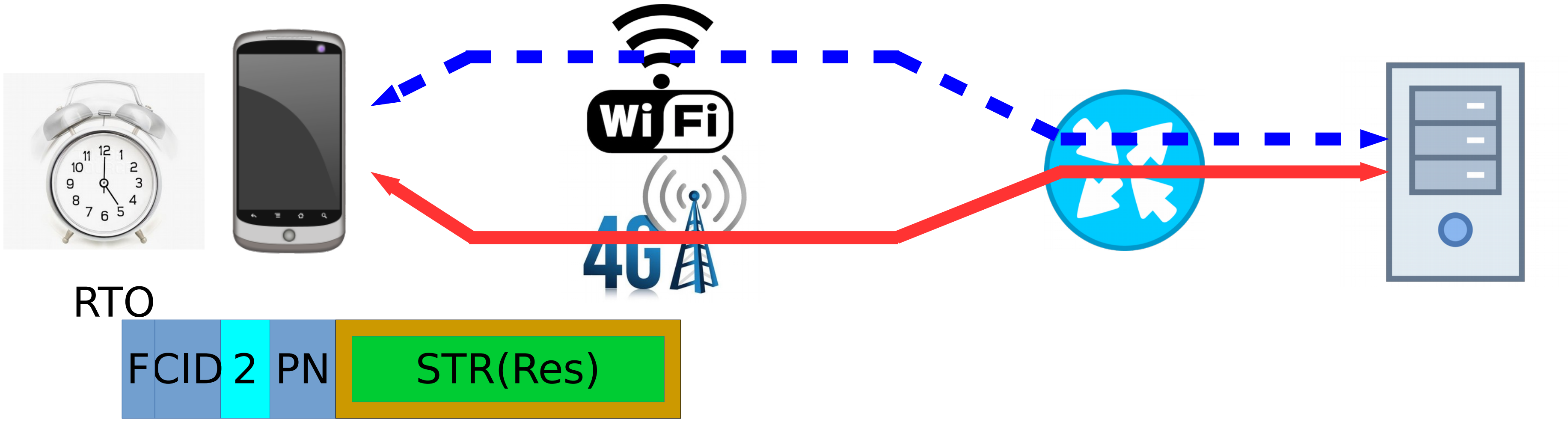
RTO

	FCID	2	PN	STR(Req)	PATHS(1 lossy)

What Happened During Handover?



What Happened During Handover?



RTO

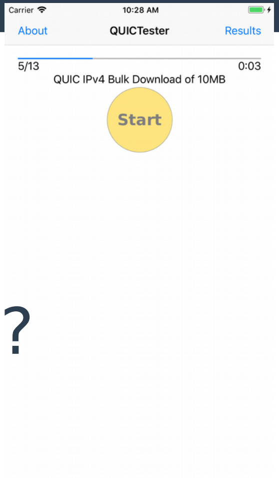
FCID	2	PN	STR(Res)

What about actual networks?

QUICTester Application

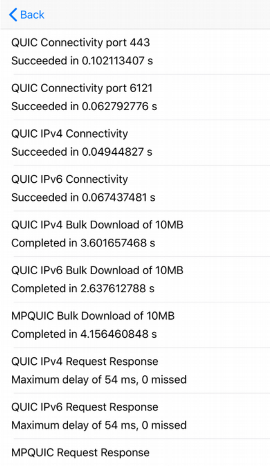
- **Perform tests in actual networks**
  - Does (MP)QUIC work in **your** networks?
  - Does MPQUIC provides better performances?
  - Application running on iOS11
    - https://itunes.apple.com/fr/app/quictester/id1322019644?mt=8
  - Feel free to provide feedback :-)

QUICTester



5/13 0:03

QUIC IPv4 Bulk Download of 10MB



QUIC Connectivity port 443
Succeeded in 0.102113407 s

QUIC Connectivity port 6121
Succeeded in 0.062792776 s

QUIC IPv4 Connectivity
Succeeded in 0.04944827 s

QUIC IPv6 Connectivity
Succeeded in 0.067437481 s

QUIC IPv4 Bulk Download of 10MB
Completed in 3.601657468 s

QUIC IPv6 Bulk Download of 10MB
Completed in 2.637612788 s

MPQUIC Bulk Download of 10MB
Completed in 4.156460848 s

QUIC IPv4 Request Response
Maximum delay of 54 ms, 0 missed

QUIC IPv6 Request Response
Maximum delay of 54 ms, 0 missed

MPQUIC Request Response

To sum up…

Conclusion

- **Multipath should be part of any transport protocol**
  - Most devices are multihomed
- **Designed and implemented Multipath QUIC**
  - Source code + artifacts + IETF draft available
  - See multipath-quic.org
- **Multipath more promising with QUIC than TCP**
  - Also opens potential new use cases

Thanks!

multipath-quic.org