Multipath QUIC: Design and Evaluation

Quentin De Coninck, Olivier Bonaventure
quentin.deconinck@uclouvain.be
multipath-quic.org
QUIC = Quick UDP Internet Connection

- TCP/TLS1.3 atop UDP
- Stream multiplexing → HTTP/2 use case
- 0-RTT establishment (most of the time)
QUIC Packet

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

Does not depend on 4-tuple

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

Cleartext Public Header
QUIC Packet

Does not depend on 4-tuple

Monotonically Increasing

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

Cleartext Public Header
QUIC Packet

- **Flag**
  - Does not depend on 4-tuple

- **Connection ID**
  - Monotonically Increasing

- **Packet Number**
  - Monotonically Increasing

- **Encrypted Payload...**
  - Contains control/data frames

**Cleartext Public Header**
QUIC Data Transfer

STREAM(id=5, off=0):"Some data in my long frame"
QUIC Data Transfer

STREAM(id=5,off=0): "Some data in my long frame"
QUIC Data Transfer

STREAM(id=5, off=0): "Some data in my long frame"

ACK(25) MAX_DATA(for stream=5): 1024
QUIC Data Transfer

H1

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

Control Frames

H2
QUIC Data Transfer

H1

| 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 |

| F | CID | PN=26 | STREAM(id=5,off=26):"." STREAM(id=7,off=0):"Y" ACK(19) |

H2
QUIC Data Transfer

1. **F CID PN=25**: 
   - **STREAM(id=5,off=0):"Some data in my long frame"**

2. **F CID PN=19**: 
   - **ACK(25)**
   - **MAX_DATA(for stream=5): 1024**

3. **F CID PN=26**: 
   - **STREAM(id=5,off=26):"."**
   - **STREAM(id=7,off=0):"Y"**
   - **ACK(19)**

**Multiplexing**
QUIC Data Transfer

STREAM(id=5,off=0): "Some data in my long frame"

ACK(25) MAX_DATA(for stream=5): 1024

STREAM(id=5,off=26): "." STREAM(id=7,off=0): "Y" ACK(19)

ACK(26)
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

<table>
<thead>
<tr>
<th>Flags</th>
<th>Connection ID</th>
<th>Path ID</th>
<th>Packet Number</th>
<th>Encrypted Payload...</th>
</tr>
</thead>
</table>

Explicit path identification
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 → **No path handshake**

Per-path numbering space
Multipath QUIC Data Transfer

Path 1: WiFi

Path 2: LTE
Multipath QUIC Data Transfer

Path 1: WiFi

Path 2: LTE

Server via WiFi

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

Phone

Server via LTE
Multipath QUIC Data Transfer

Path 1: WiFi

CID F PN=1 STR(id=5)

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

Path 2: LTE

CID F PN=1 STR(id=7,off=1024)
Multipath QUIC Data Transfer

Server via WiFi

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

CID F PN=1

STR(id=7,off=0)

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

Path 1: WiFi

Phone

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

Server via LTE

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

Path 2: LTE
Multipath QUIC Data Transfer

Path 1: WiFi

- F CID 1 PN=1 STR(id=5)
- F CID 1 PN=1 STR(id=7,off=0)
- F CID 1 PN=2 ACK(pid=1,1)
- F CID 1 PN=1 ACK(pid=2,1)

Path 2: LTE

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

Multiple paths acked on a single path
Multipath Mechanisms

- Path management

![Diagram showing IP1, IP2, IP3, and IP4 connections]
Multipath Mechanisms

- Path management

![Diagram showing IP1, IP2, IP3, and IP4 connections]

- IP1 connected to IP2
- IP3 connected to IP4
- IP2 connected to IP4
- IP1 connected to IP3
Multipath Mechanisms

- Path management
- Packet scheduling
Multipath Mechanisms

- Path management
- Packet scheduling

20 ms RTT
10 ms RTT
Multipath Mechanisms

- Path management
- Packet scheduling
Multipath Mechanisms

- Path management
- Packet scheduling
Multipath Mechanisms

- Path management
- Packet scheduling
- Congestion control
  - Opportunistic Linked Increase Algorithm
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 Bandwidth Aggregation

• Download of 20 MB file
  - Over a single stream
  - Collect the transfer time
Evaluating Bandwidth Aggregation

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

Diagram:
- 20ms RTT, 20 Mbps
- 40ms RTT, 15 Mbps
Evaluating Bandwidth 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 Bandwidth 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
- **But what about other topologies?**
Evaluating Bandwidth Aggregation

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

<table>
<thead>
<tr>
<th>Factor</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity [Mbps]</td>
<td>0.1</td>
<td>100</td>
</tr>
<tr>
<td>Round-Trip-Time [ms]</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>Queuing Delay [ms]</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Random Loss [%]</td>
<td>0</td>
<td>2.5</td>
</tr>
</tbody>
</table>
Large File Download - No Loss

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

CDF

Time TCP / QUIC

TCP better

QUIC better

Time Ratio

10^{-1} \rightarrow 10^{1}
Large File Download - No Loss

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

Time TCP / QUIC

Single-path

TCP better

QUIC better

CDF

Time Ratio

10^{-1} - 10^0 - 10^1
Large File Download - No Loss

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

MPQUIC better in 85% of cases
Large File Download - No Loss

GET 20 MB, 506 simulations low-BDP-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 queuing 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

GET 20MB, 506 simulations, low-BDP-losses

CDF

Time TCP / QUIC
Time MPTCP / MPQUIC
Large File Download - Losses

GET 20MB, 506 simulations, low-BDP-losses

QUIC copes better with losses
Additional Results (see paper)

- **QUIC benefits more of Multipath than TCP**
- **Bandwidth aggregation in high BDP**
  - MPQUIC still better performs than MPTCP
- **Short file transfers**
  - (MP)QUIC better thanks to its low latency handshake
- **Network handover**
  - MPQUIC can be very efficient
  - New frame to communicate path state
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**
What’s Next?

- **Perform tests in actual networks**
  - Does (MP)QUIC work in **your** networks?
  - Does MPQUIC provides better performances?
  - Application running on iOS11
  - Feel free to provide feedback :-)

---

**QUICTester**
Thanks!

multipath-quic.org