

# Deadline-aware Streams in QUIC

draft-tjohn-quic-multipath-dmtp-01

QUIC Meeting @ IETF-123 Madrid

Tony John  
NetSys Lab, OVGU Magdeburg

24.07.2025

[draft-tjohn-quic-multipath-dmtp-01](#)

# Brief History

## From Research to Internet Draft:

- June 2023: [DMTP: Deadline-aware Multipath Transport Protocol](#) (Paper at IFIP Networking)
  - Meet deadlines using multipath over SCION
- Oct 2024: Proposed as QUIC-MULTIPATH extension (Issue [#453](#) in multipath repo)
- Mar 2025: [draft-00](#) published, presented DMTP at [PANRG](#)
- June 2025: [draft-01](#) published after WG feedback:
  - Removed custom DMTP\_ACK → reuse timestamp extensions
  - Use of multipath optional - works now with single-path QUIC
- June 2025: MoQ discussions on partial reliability → decided on stream reset

# Why Deadline-aware Streams

## Modern real-time applications need:

- Strict latency
- High reliability
- Predictable performance

## Targeted use cases:

Teleoperation

Telesurgery

Live Streaming

Online Gaming

## Current QUIC streams:

- No awareness of time constraints
- Bandwidth wasted on expired data
- Head-of-line blocking with stale data
- Retransmits all lost packets

## Examples:

- Video frame arrives after playback time
- Game state update arrives too late
- Teleoperation command misses control deadline

# How DMTP fits with QUIC/QUIC-MULTIPATH

## With standard QUIC:

- No protocol changes required, but recommendations:
  - Sender-side deadline-aware scheduling
  - Leverages existing stream priority mechanism

## With QUIC-MULTIPATH:

- Minimal protocol extensions:
  - Path selection based on deadline
  - ACK/retransmit on fast path

## Path Aware Network (PAN) mapping:

- Map PAN paths to PATH IDs in QUIC-MULTIPATH

# Deadline-aware Streams over QUIC

## Three core mechanisms:

### 1. Per-chunk deadline tracking:

- Timestamp when data added to stream queue
- $\text{Deadline} = \text{enqueue\_time} + \text{deadline\_ms}$

### 2. Deadline-aware scheduling:

- Map deadline urgency to stream priority

### 3. Stream lifecycle management:

- Track expired vs delivered bytes
- Reset stream if thresholds exceeded

### Stream deadline:

- Relative to when data is queued
- Not relative to stream creation

### Expiry condition:

- $\text{now} > (\text{deadline} - \text{one-way delay})$
- One-way delay from  $\text{smoothed\_rtt}/2$  or  $\text{ACK\_EXTENDED}$

# Deadline-aware Streams over QUIC-MULTIPATH

## Protocol extensions:

### Transport parameter:

- enable\_deadline\_aware\_streams
  - Negotiated during handshake
  - Enables deadline-aware behavior

### DEADLINE\_CONTROL frame:

- Stream ID (i): Target stream
- Deadline (i): Relative deadline in milliseconds

## Benefits:

### Path selection opportunities:

- Schedule urgent data on lowest RTT path
- Send ACKs on fast-enough return path
- Retransmit on fast-enough path

### Enhanced strategies:

- Acknowledge probes on same path
- Dynamic path switching based on deadline urgency
- FEC repair symbols on diverse paths

# Support For Path Aware Networks (PAN)

## Path Aware Networks (e.g. SCION):

- Explicit path selection on endhosts
- Single address pair → Multiple distinct end-to-end paths
- See [draft-zaeschke-scion-quic-multipath-00](#)
  - Presented at [PANRG](#) meeting

## Added benefits for deadline-aware streams:

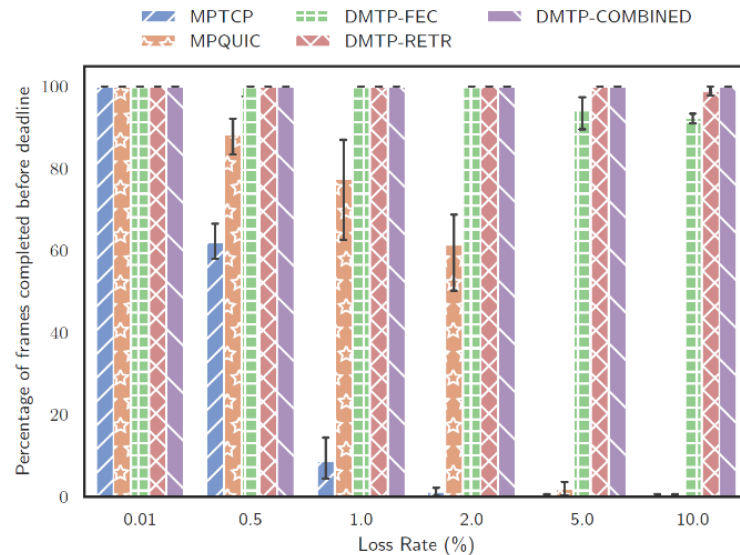
- Wider and diverse paths
- Not limited to multi homing
- Select disjoint paths / identify shared bottlenecks
- Geofencing

## Mapping to QUIC-MULTIPATH:

- Each PAN path → Unique Path ID
- Deadline-aware scheduling – Select best path(s)

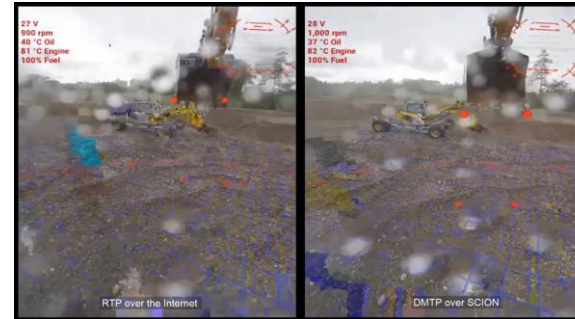
# Prototype Implementation

## SCION Based DMTP Implementation - Evaluation



Comparison with QUIC-MP and MPTCP

## Field test on remote excavator control: [Video](#)



## QUIC-MULTIPATH implementation based on picoquic

- Code: [github.com/netsys-lab/picoquic](https://github.com/netsys-lab/picoquic)
- Evaluation using picoquic's simulation framework:
  - Average: +18% deadline compliance
  - Best case: +88% deadline compliance (Asymmetric paths)



## Next Steps

- **Seeking feedback on:**
  - Interest in QUIC WG
  - Integration with MoQ
    - Object aligned deadlines and partial reliability
    - Path aware relays
- **Draft-02: ongoing**
  - Lessons learned from implementation

## References

- Email: [tony.john@ovgu.de](mailto:tony.john@ovgu.de)
- Implementation: <https://github.com/netsys-lab/picoquic>
- PANRG [slides](#)