An Interface to the QUIC Transport Protocol

draft-pauly-quic-interface-00

Tommy Pauly, Eric Kinnear, Brian Trammell

QUIC
September 2018, NYC

Overview

Define a standard application interface for QUIC as a transport

Discuss two potential “API mappings” of transport objects to QUIC streams

Abstractions

Previously discussed[1]

- QUIC (base)
- Connections
- Streams
- Compression
- HTTP

[1] https://github.com/quicwg/wg-materials/blob/master/interim-18-01/abstractions.pdf

Abstractions

Previously discussed

- QUIC (base)
  - Packets, loss detection, cryptographic context
- Connections
- Streams
- Compression
- HTTP

Abstractions

Previously discussed

- QUIC (base)
- Connections
  - Connection ID, multiplexed, non-HoL-blocking streams, congestion control, loss recovery
- Streams
- Compression
- HTTP

Abstractions

Previously discussed

- QUIC (base)
- Connections
- Streams
  - Ordered bytestreams, stream-level flow control, data and non-data frames
- Compression
- HTTP

Abstractions

Previously discussed

- QUIC (base)
- Connections
- Streams
- Compression
  - Flow controlled headers, compression
- HTTP

Abstractions

Previously discussed

- QUIC (base)
- Connections
- Streams
- Compression
- HTTP
  - Maps requests to streams, HTTP semantics

Motivations

Why do we need to talk about interface?

- Common abstractions result in common programming models
- Transport API needs to support benefits
    - Non-HoL-blocking, 0-RTT
- Common surface yields (more) portable implementations
- Standard configuration of security properties

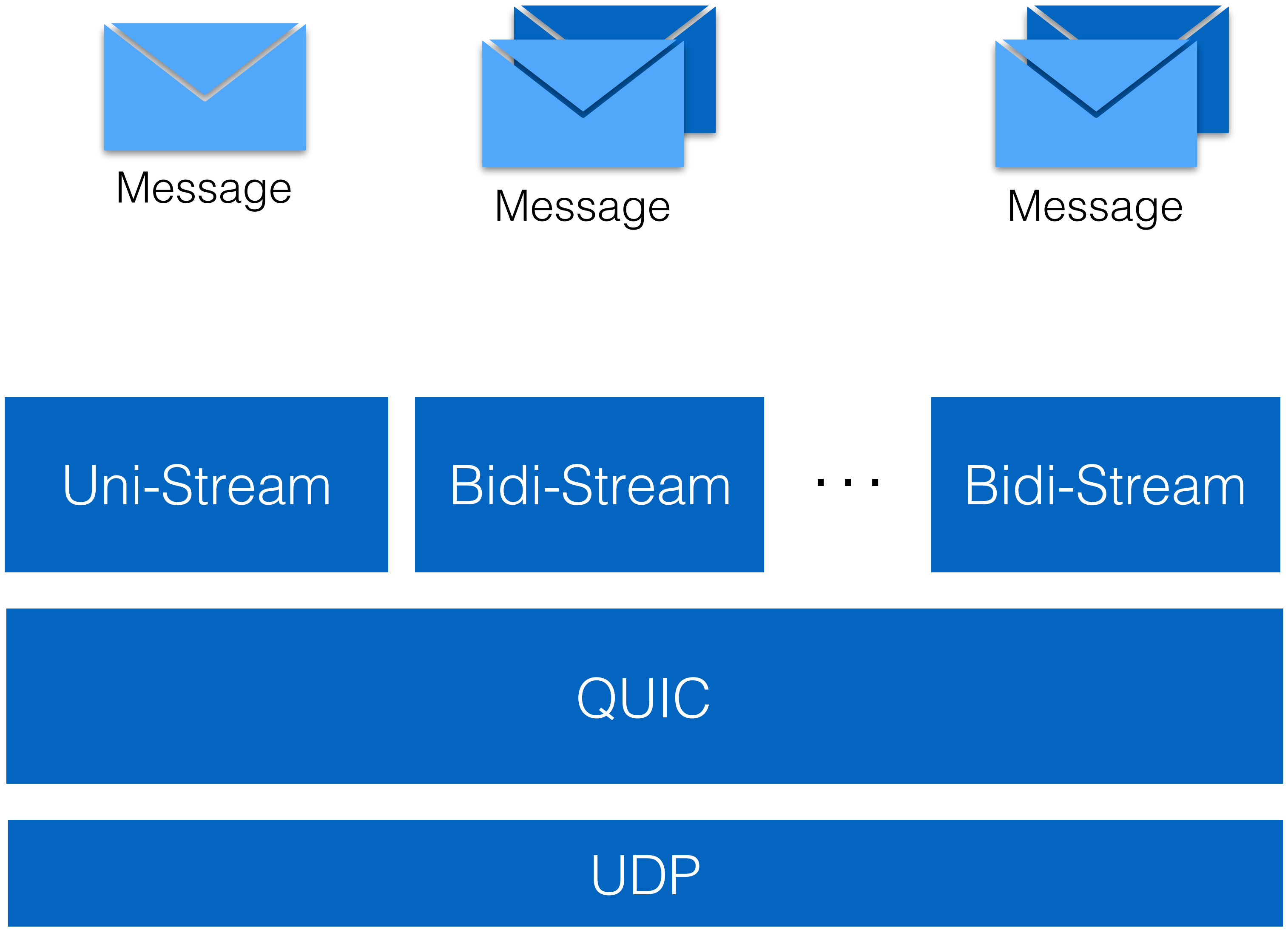
API Mappings

Transport Connection as QUIC Connection

Transport Connection as QUIC Stream

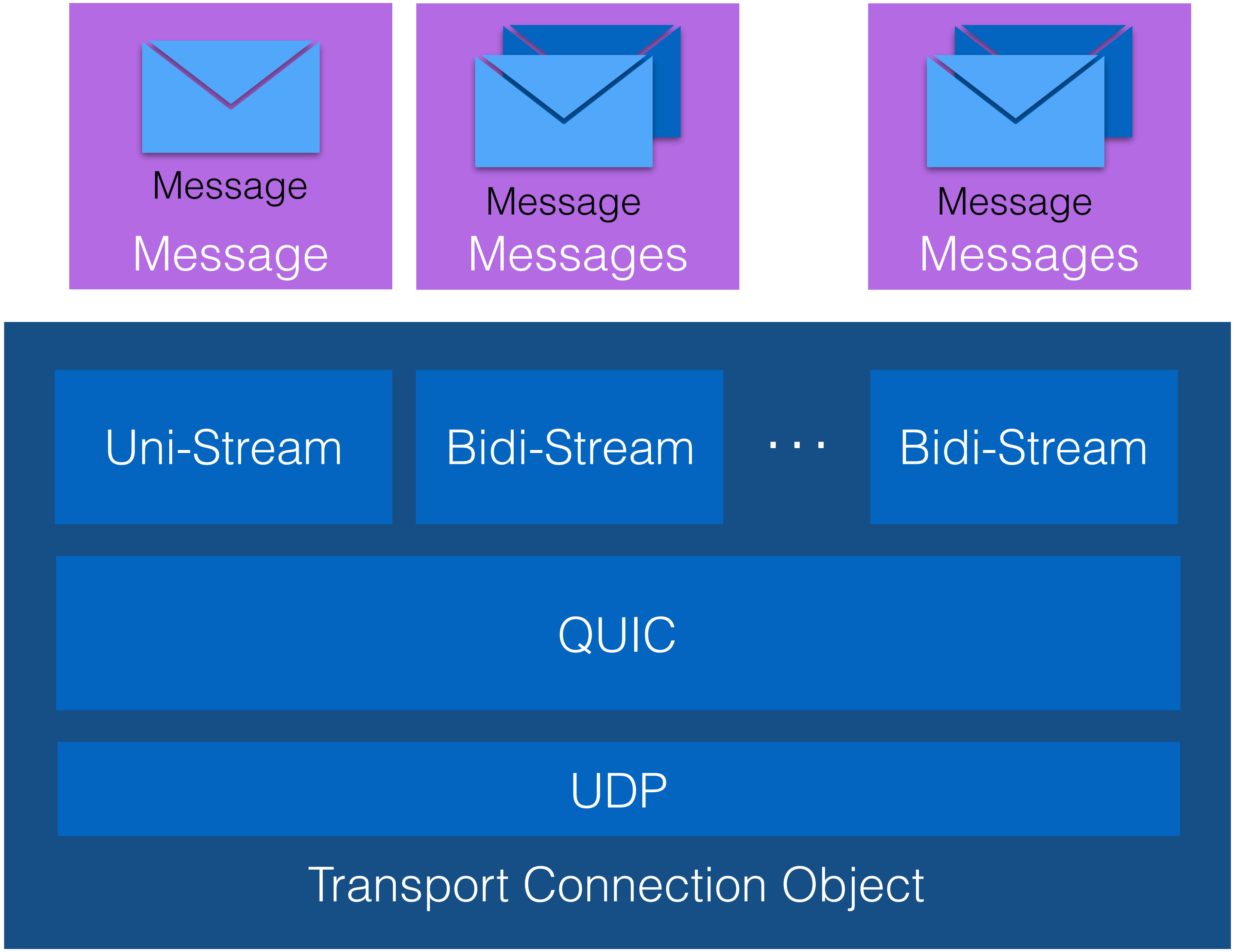
Transport Connection as QUIC Connection

Every Message object corresponds to a QUIC stream



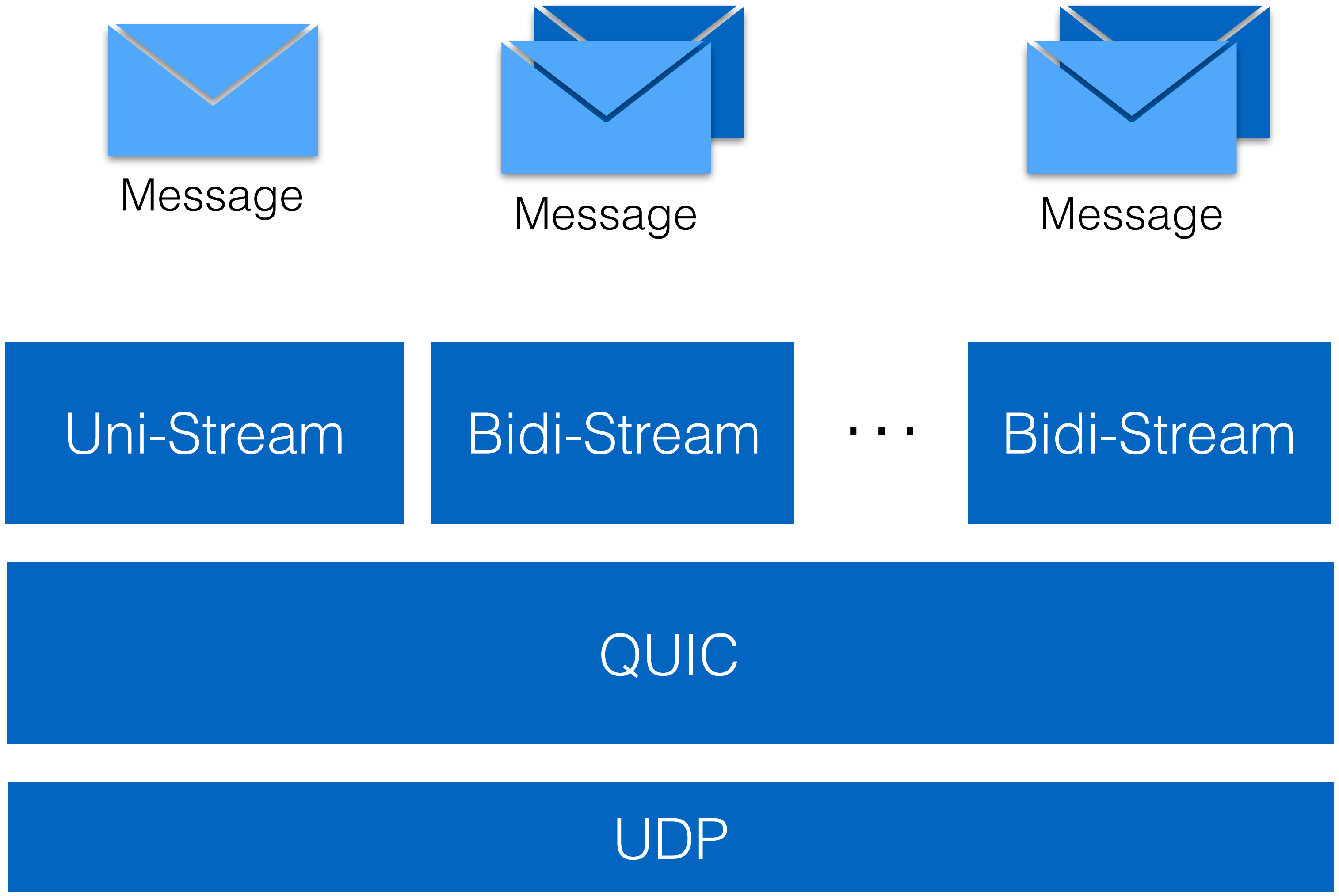
Transport Connection as QUIC Connection

Every Message object corresponds to a QUIC stream



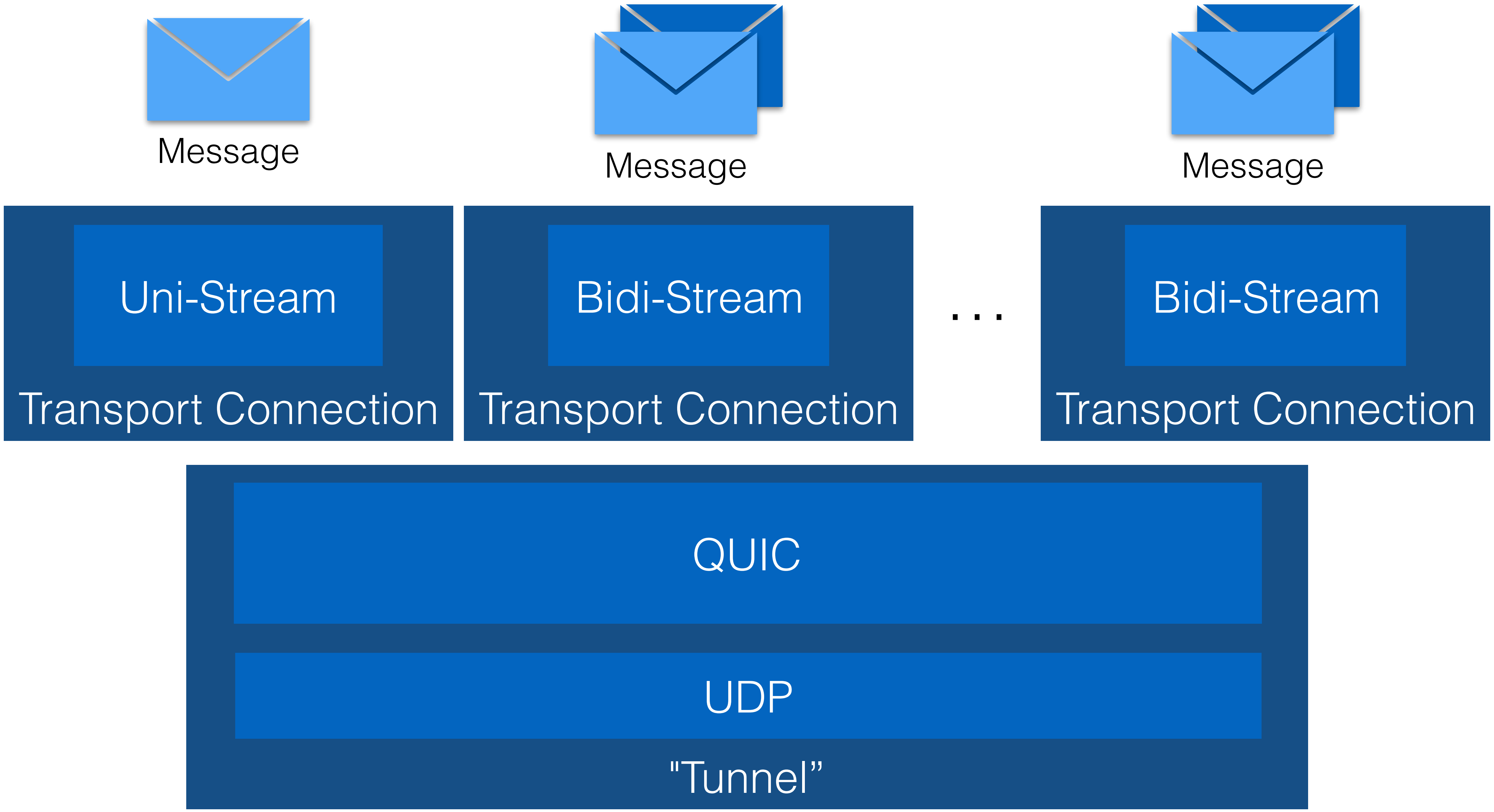
Transport Connection as QUIC Stream

A QUIC stream is a (bi/uni)directional ordered bytestream with flow control and loss recovery.



Transport Connection as QUIC Stream

A QUIC stream is a (bi/uni)directional ordered bytestream with flow control and loss recovery.



Thoughts?

Other interface directions?

Capabilities that must be supported by any option?

Next Steps

Start discussion on the list?

Work on sample application code?