UCLouvain

Multipath Extensions for QUIC

draft-deconinck-quic-multipath

Quentin De Coninck (UCLouvain, Belgium)

https://qdeconinck.github.io/

6 February 2020

QUIC Interim, Zurich

Why Using Multiple Paths?

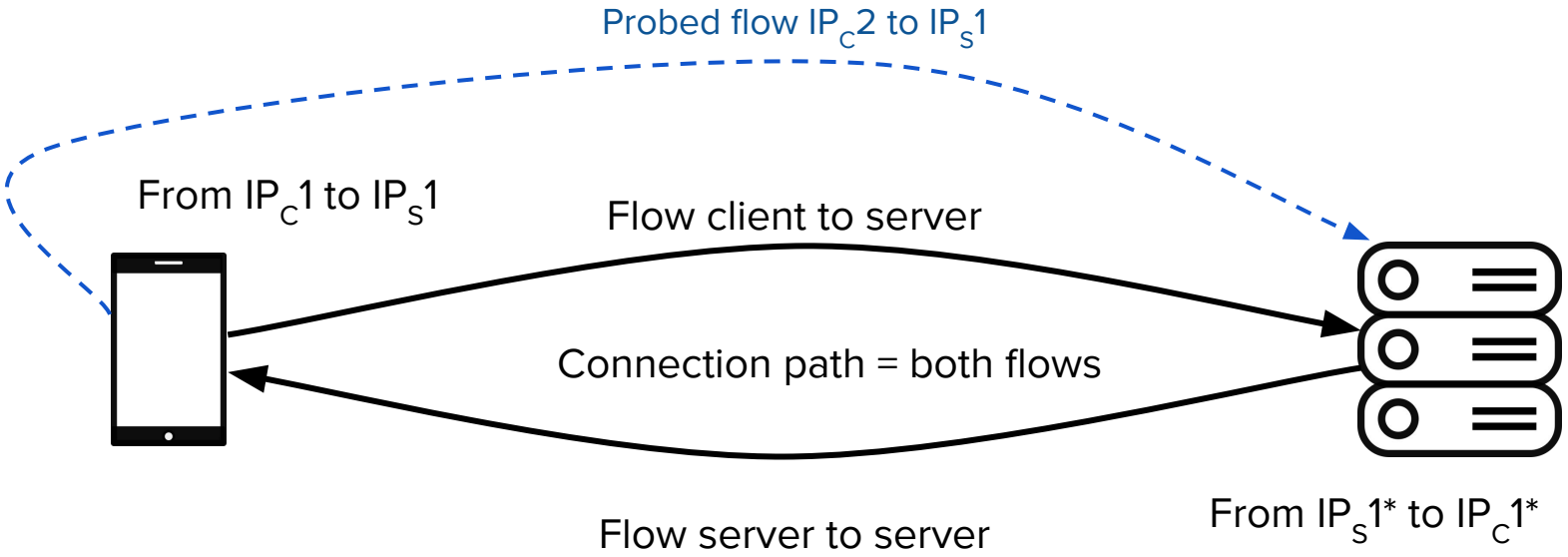
- Bandwidth aggregation
  - MPTCP: Hybrid networks, 4G/WiFi aggregation in Korea
- Network resiliency
  - E.g., duplicate some frames over multiple networks when paths experience connectivity issues
  - MPTCP: iOS deployment
- 3GPP discussions to integrate MPQUIC solution for 5G

Design Goals

Enable **simultaneous** usage of multiple network paths while

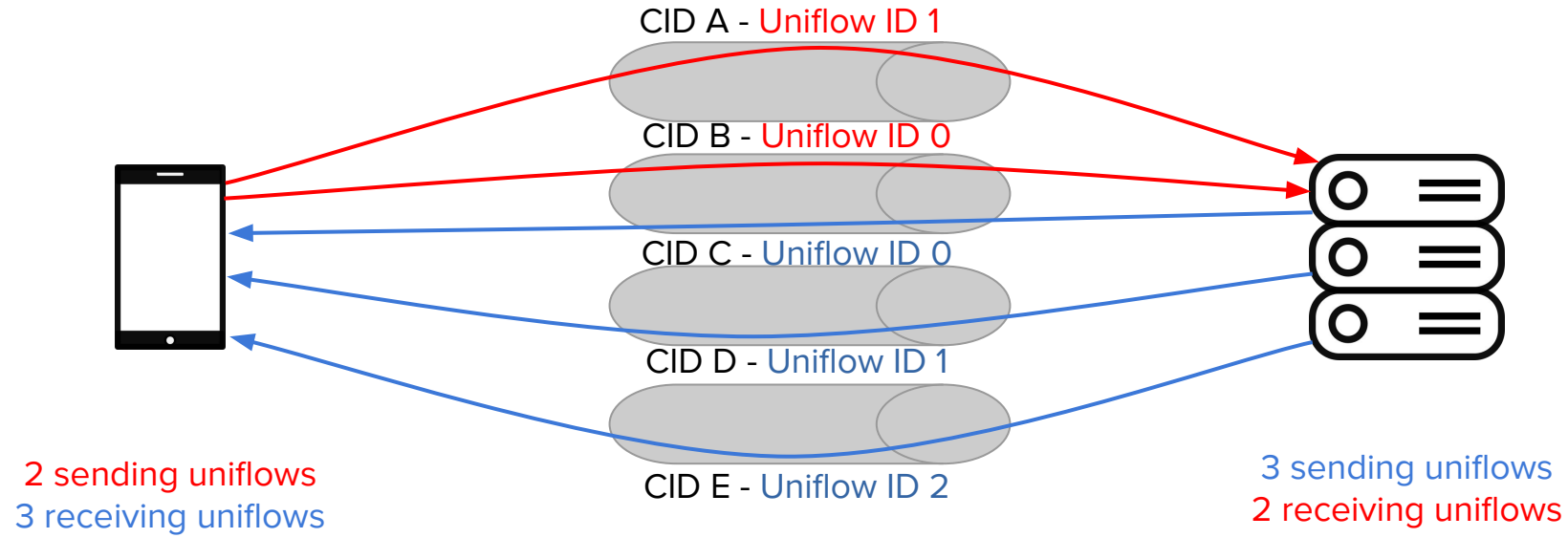
- Keeping control on path management
- Not introducing new privacy concerns
- Validating new addresses before use
- Handling network asymmetry

(Current) Notions of Paths in QUIC



Flow tuples on client and server are not necessarily the same (e.g., NAT)

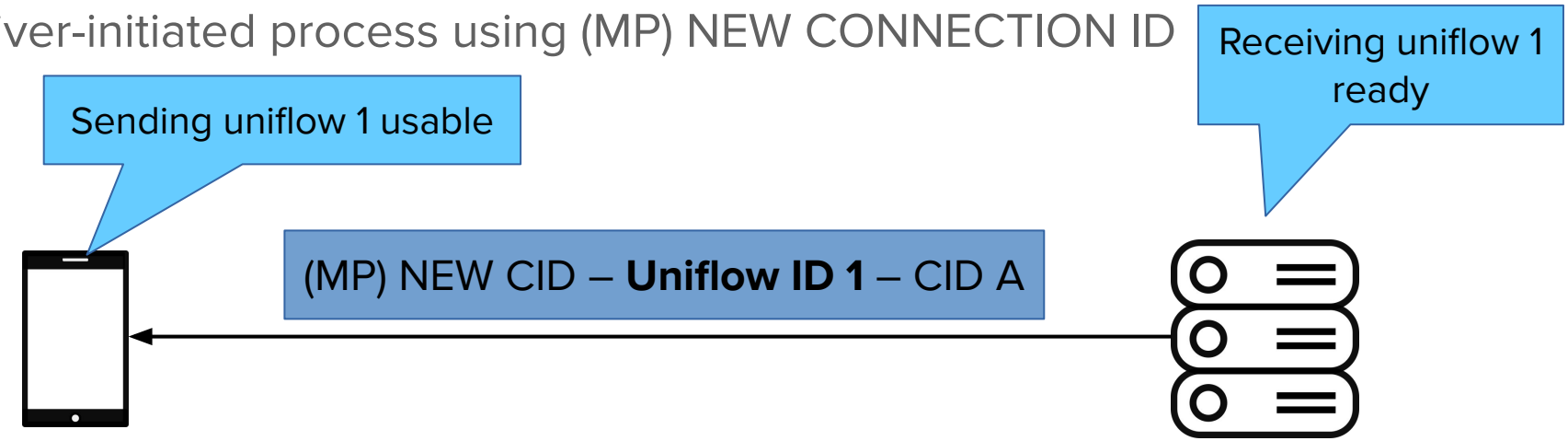
Idea of “Unidirectional Flows” (a.k.a. Uniflows)



- Each uniflow has its (set of) Connection ID(s)

Proposing Uniflows

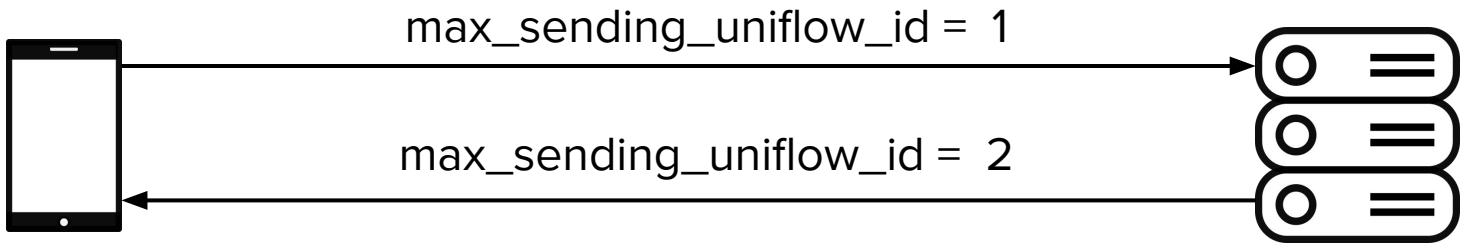
Receiver-initiated process using (MP) NEW CONNECTION ID



- Host determines the number of sending uniflows of its peer
  - = number of different Uniflow IDs sent in (MP) NEW CID frames
- Can only use Destination CID from different Uniflow IDs simultaneously
- Path migration of a given Uniflow still possible

Negotiating the Multipath Extensions

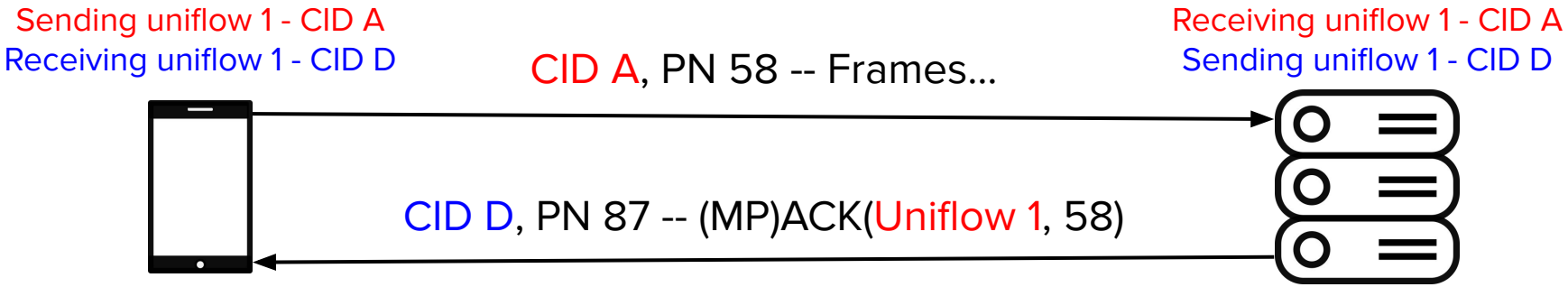
Use a transport parameter



- Put a limit on the number of desired sending uniflows
  - Limit number of peer's receiving uniflows
- Initial path has (implicit) uniflow ID 0
- If one does not advertise the TP, extension is disabled

Acknowledging Packets Sent on Uniflows

- One packet number space per uniflow
- (MP)ACK frame indicates the **receiving Uniflow ID** it acknowledges

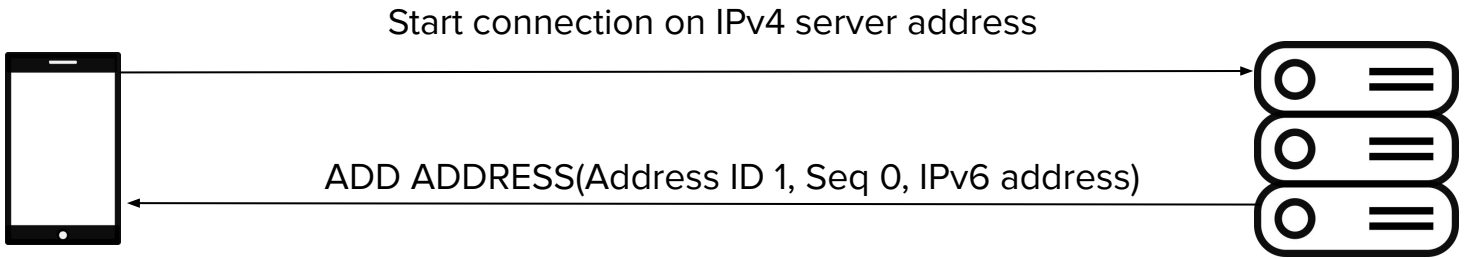


Cons of using a single packet number space:

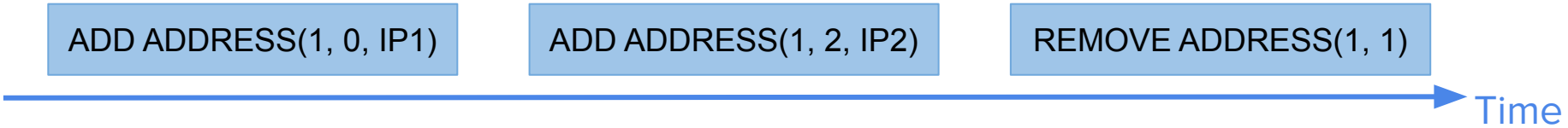
- May complexify ACK processing, more ACK blocks if path reordering
- What if a single ACK acknowledges two packets from different uniflows?
  - Uniflow latency estimation?

Advertising Host Addresses

Relies on ADD ADDRESS and REMOVE ADDRESS frames



- Similar to preferred_address here
- Enable each host to advertise address changes (and possibly address priority)
- Address ID events ordered by a sequence number to handle reordering



- Hosts can indicate current Address ID / Uniflow ID mappings in dedicated PATHS frame

Estimating the Latency

(MP)ACK frames can be sent on any (sending) uniflow

- Estimate RTT between tuple (sending uniflow ID, receive uniflow ID)
- Would be interesting to estimate One-Way-Delay of uniflows
  - draft-huitema-quic-1wd

Multipath Algorithms

- Path management (link uniflow to a 4-tuple)
  - Client can link a uniflow to pair of local address / remote validated address
  - Server should first wait for client packet to learn client addresses
    - Then link uniflows to pair of local address / seen and validated client address
- Packet scheduling
  - Only applies on sending uniflows
  - Can be lowest-latency oriented, priority based,...
  - Frames like MAX DATA should be duplicated on all sending uniflows
    - To limit head-of-line blocking

Is This Complex to Implement?

- quicly -- ~350 lines of diff
  - Very basic multipath support with round-robin
- PQUIC multipath plugin (based on picoquic) -- 2500 lines of C code
  - Nearly complete implementation, RTT + round-robin schedulers
  - See https://pquic.org
- Connection migration already implements a lot
  - Path validation,...

Next Steps

Multipath ID is ongoing work, feel free to contribute

- https://github.com/qdeconinck/draft-deconinck-multipath-quic

Also preparing a new version of the MultipathTester iOS app

- Include support for IETF (multipath) QUIC
- https://apps.apple.com/us/app/multipathtester/id1351286809
- See also https://multipath-quic.org