Flow-state signaling and QUIC
(draft-trammell-plus-statefulness)

Brian Trammell
QUIC WG - IETF 97 Seoul - 15 Nov 2016
A Problem

- Lots and lots of state-keeping devices on path…
  - … that assume TCP semantics
  - … won’t work with non-TCP transports
- UDP-based transports need:
  - frequent keepalives
  - explicit directional rules or port mapping
  - other nasty hacks
  - or fall back to TCP.
- Common *wire image* for UDP-based, encrypted transport protocols like QUIC.
A Solution

Diagram:
- **zero**
- **uniflow**
- **closing**
- **biflow**

Transitions:
- \( t_1 \) from **zero** to **uniflow**
- \( t_2 \) from **biflow** to **closing**
- \( t_3 \) from **closing** to **zero**
- Associate from **biflow** to **uniflow**
- Forward packet from **uniflow**
- Forward packet from **biflow**
- Stop from **closing**
- Any packet from **uniflow**
Why should QUIC care?

- Requires two signals to drive:
  - Associate: “receiving endpoint thinks this traffic is good, OK to send more,” replaces SYN/ACK.
  - Stop: “receiving endpoint stopping connection or thinks this traffic is bad,” replaces FIN, RST.
How to add to QUIC?

• Associate *almost exists*: initial recipient sends back packet to initial sender with the same Connection ID, assume connection is OK.

• Works in the middle of a connection (e.g. on firewall restart), as long as you see ACKs / keepalives.

• Stop *needs a new signal*. If every QUIC connection ended with the equivalent of a Public Reset, this would be sufficient.