Issue #1764

QUIC Interim, New York, September 2018

First Principles

The server must never send more than 3x the number of bytes to an unvalidated address than it has (apparently) received from that address.

We can argue about 3 vs 4 vs ?, but there needs to be a limit

Status Quo

Server can only send 3x the bytes received from the client prior to address validation

ACK of Initial does NOT count as path validation

Receipt of Handshake counts as path validation

Server/Handshake deadlock

Server sends up to 3 packets in response to the Initial

Response includes an ACK of the client's Initial

Client has nothing to send until it receives the entire server flight

Deadlock If: Server's Handshake packets OR Client's ACK of Handshake packets are lost

3 packet limit pseudocode

```
int BytesToSend() {
  if (!PathValidated()) {
    return max(0, 3 * bytes_received - bytes_sent);
  }
  return max(congestion_window - bytes_in_flight);
}
```

Receipt of more bytes unlocks sending

Suggestion: Client keeps sending

If the entire ServerHello hasn't been received, ensure a **full-sized** reliable packet is outstanding, Initial or otherwise. Receipt provides extra ability to send.

If the entire ServerHello has been received, client should instead send something reliable in Handshake encryption.

- Prior to path validation, client is sending large packets
- Client drives the server to path validation w/Handshake

Old slides

Option 1: Keep retransmitting Client Initial

If the entire server flight hasn't been received, client keeps retransmitting the Initial on a timer

- Comes down to, if the client doesn't have 1RTT keys, it must retransmit the Initial
- and the server must retransmit all unacknowledged data when it receives an Initial

Cons: Initial is more data than typically necessary

Option 2: Client keeps something in flight

If the Initial is explicitly or implicitly ACKed, send something (ie: PING) that's retransmittable in Handshake

Con: New behavior

If ServerHello is multi-packet, client can't send Handshake

Option 3: Retry Token

The Server MUST provide an address validation token with the Initial Packet that contains Server Hello.

The client MUST echo this token in the Initial Packet that ACKs the server hello, which is sufficient to verify the path.

The server can then retransmit Handshake packets as necessary. Uses existing protocol mechanisms.

Con: An optimization, but does not solve the problem

Option 4: Require coalescing Initial & Handshake

If a server must coalesce Initial with Handshake AND a client must coalesce the ACKs, then the server is guaranteed to receive a Handshake ACK.

Con: an optimization that does not solve the problem

Option 5: No Initial ACKs

IF ACKs are never sent in Initial, then the issue is only if a client's ACK is lost and part of the server flight is missing.

=> Require client ACK be reliable(PING/PATH_CHALLENGE)

Con: Doesn't solve HRR and other cases with large Initial flights (e.g. with PADDING)

Handshake is ‘special’

Timer-based retransmissions could exceed the amplification factor

=> Special cases are necessary

Unfortunately, the handshake is very latency sensitive