Updates to 0-RTT-BDP

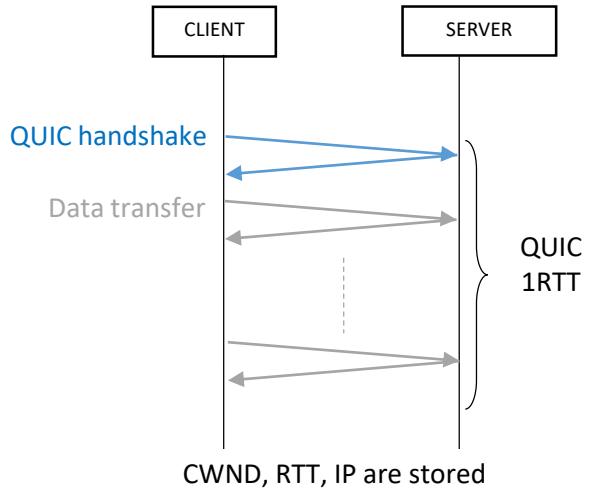
N. Kuhn - CNES
E. Stephan - Orange
G. Fairhurst - University of Aberdeen
T. Jones - University of Aberdeen
C. Huitema - Private Octopus Inc.

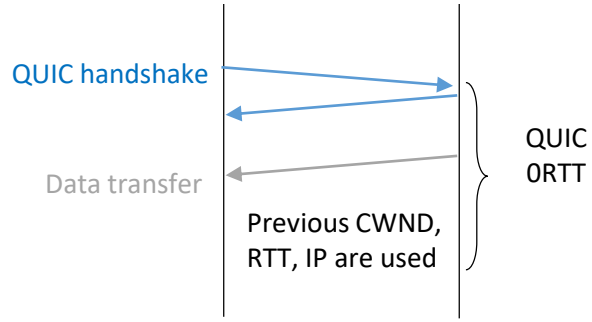
0RTTBDP

Core idea

1. During a previous session, current RTT (current_rtt), CWND (current_cwnd) and client's current IP (current_client_ip) are stored as saved_rtt, saved_cwnd and saved_client_ip;

2. When resuming a session, the server might set the current_rtt and the current_cwnd to the saved_rtt and saved_cwnd of a previous connection.





Rationale behind the safety guidelines

Previously measured saved_rtt and saved_cwnd should not be used as-is to avoid potential congestion collapse:

- Rationale #1: An Internet method needs to be robust to network conditions that can differ between sessions.

- Rationale #2: Information sent by a malicious client would not be relevant since it might try to convince servers to use a CWND higher than required. This could increase congestion.

Solutions and associated trade-offs

Rationale	Solution	Advantage	Drawback
#1 : Variable network	#1 : set_current_* to saved_*	Ingress optimization	Risks of adding congestion
	#2 : implement safety check	Reduce risks of adding congestion	Negative impact on ingress optimization
#2 : Malicious client	#1 : Local storage	Enforced security	Client can not decide to reject Malicious server coulf fill client's buffer Limited use-cases
	#2 : NEW_TOKEN	Save resource at server Opaque token protected	Malicious client may change token even if protected Malicious server could fill client's buffer Server may not trust client
	#3 : BDP extension	Extended use-cases Save resource at server Client can read and decide to reject BDP extension protected	Malicious client may change BDP even it protected Server may not trust client

Implementation recommendations

[RFC9000]: "Generally, implementations are advised to be cautious when using previous values on a new path."

This draft:

- proposes a discussion on how using previous values can be achieved in a interoperable manner
- how it can be done safely
- Integrates some implementation recommendations for BBR, NewReno and CUBIC.

Comment on no_save_metrics

If the server does not want to store the metrics from previous connections:

- Is an equivalent of the tcp_no_metrics_save for QUIC necessary ?
- Does this need to be negociable for a client to refuse the exploitation of previous sessions parameters ?

Performances

- Deployment results of the 0RTTBDP on satellite systems at IETF111

https://datatracker.ietf.org/meeting/111/materials/slides-111-maprg-feedback-from-using-quics-0-rtt-bdp-extension-over-satcom-public-access-00.pdf

  - 0-RTT vs 1-RTT
    - up to 33% gain for 500 kB
    - up to 45 % gain for 1 MB
  - 0-RTT-BDP vs 1-RTT
    - up to 67% gain for 500 kB
    - up to 62% gain for 1 MB
- On-going – results published soon : assessing fairness impact

Next steps

Status

draft-kuhn-quic-0rtt-bdp includes 3 methods

- 2 methods are implemented in picoquic
  - BDP frame - https://github.com/private-octopus/picoquic/pull/1209
  - local storage of CWND, RTT parameters - https://github.com/private-octopus/picoquic/pull/1204

- **Next**
  - Looking for other implementers
  - Integration in QUIC interop matrix