

Feedback from using QUIC's 0-RTT-BDP extension over SATCOM public access

N. Kuhn - CNES

E. Stephan - Orange

G. Fairhurst - University of Aberdeen

T. Jones - University of Aberdeen

C. Huitema - Private Octopus Inc.

This talk concerns a transport (QUIC) and a path characteristic (satellite)

- Paths can be very different characteristics
 - Higher delay $\gg 10$ mS to \sim secs of Path RTT (~ 650 ms for GEO)
 - High capacity: Large Bandwidth Delay Product
 - Capacity available on demand (not **always** available)
 - Asymmetry improves overall efficiency
- Impacts
 - Delay: Startup; Flow Control Procedure
 - BDP: Flow Control Buffers; cwnd
 - Capacity: Not safe to assume always high capacity; but mostly true
 - Asymmetry: Watch-out for ACKs, etc
- Other paths might also have similar needs.

Context

- Extension to transport parameters
 - Shared during the 0-RTT phase (RTT, BDP, etc)
 - Allows resumption using the additional transport and connection properties discovered from previous connections
- Use cases:
 - Optimizing client requests
 - Safe jump in cwnd/flow control size
 - Sharing transport information across multiple connections

Similar idea proposed for H2 and TLS1.3 in :

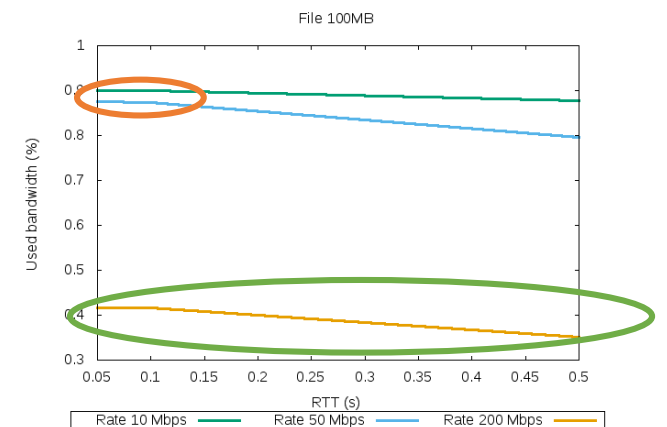
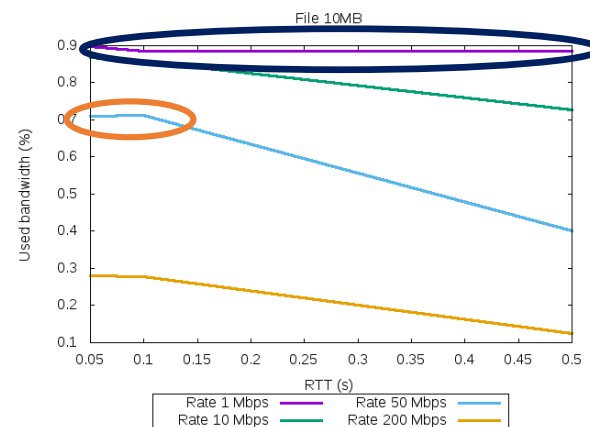
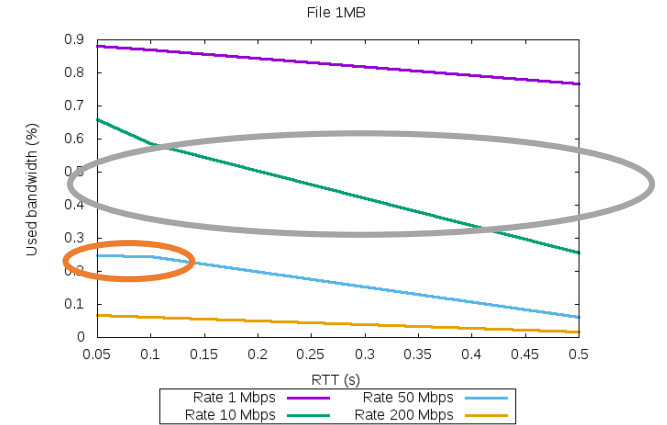
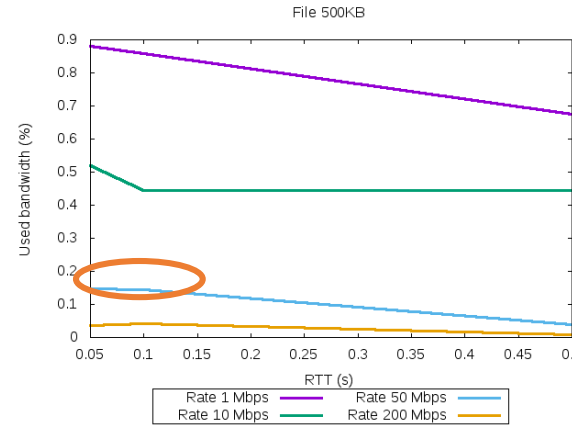
"Optimizations for Using TLS Early Data in HTTP/2 ; draft-thomson-httpbis-h2-0rtt-00"

On the need for a quick convergence to data-rate

- Tests based on open platform
 - <https://forge.net4sat.org/kuhnn/openbach-example-simple>
- QUIC :
 - PICOQUIC implementation
 - BBR (PICOQUIC implementation of BBR)
- Variable RTT (100 ms -> 500 ms)
- File size : 500 KB, 1MB, 10 MB, 100MB
- Bottleneck (Forward/Return)
 - 1 Mbps / 100 kbps
 - 10 Mbps / 2 Mbps
 - 50 Mbps / 25 Mbps
 - 200 Mbps / 100 Mbps

On the need for a quick convergence to data-rate

- With a 10 MB file and 1 Mbps, the link is used for all RTT
- For shorter files (1MB), increasing the RTT severely impacts link utilization
- When the data rate is high (250 Mbps), even a 100 MB transfer does not utilize the link
- Increasing the file size increases the link utilization

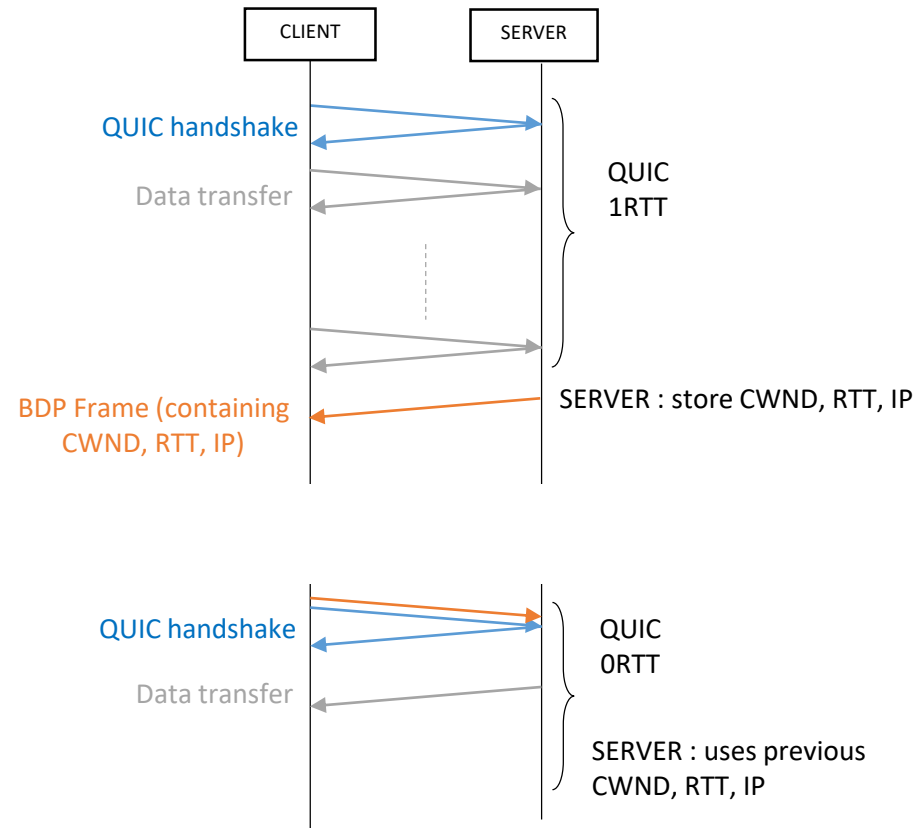


0-RTT-BDP extension : Description

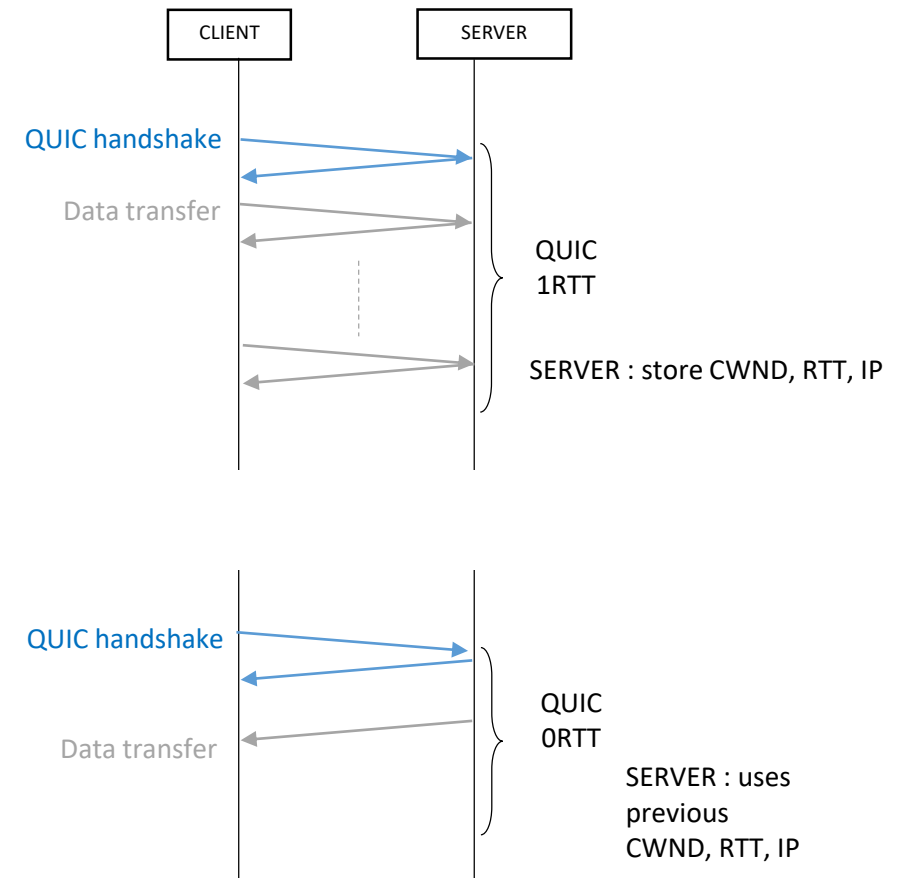
- *if time permits* presentation of various implementation approaches in QUIC WG meeting
 - <https://datatracker.ietf.org/doc/html/draft-kuhn-quic-0rtt-bdp>
 - 1. During a previous session,
 - current RTT (current_rtt), CWND (current_cwnd) and client's current IP (current_client_ip)
 - stored as saved_rtt, 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.

0-RTT-BDP extension : Description

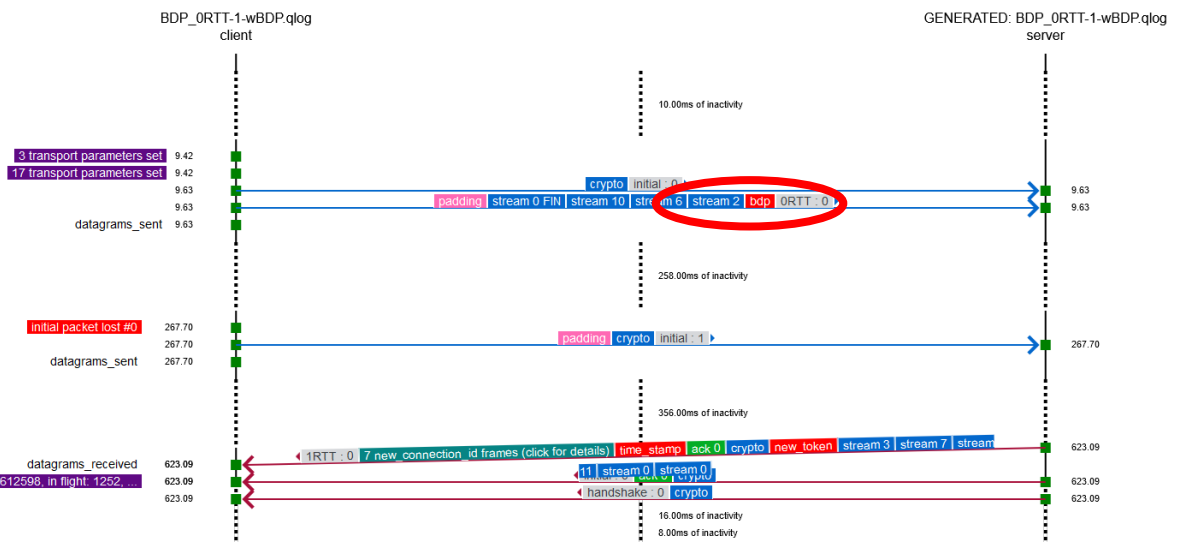
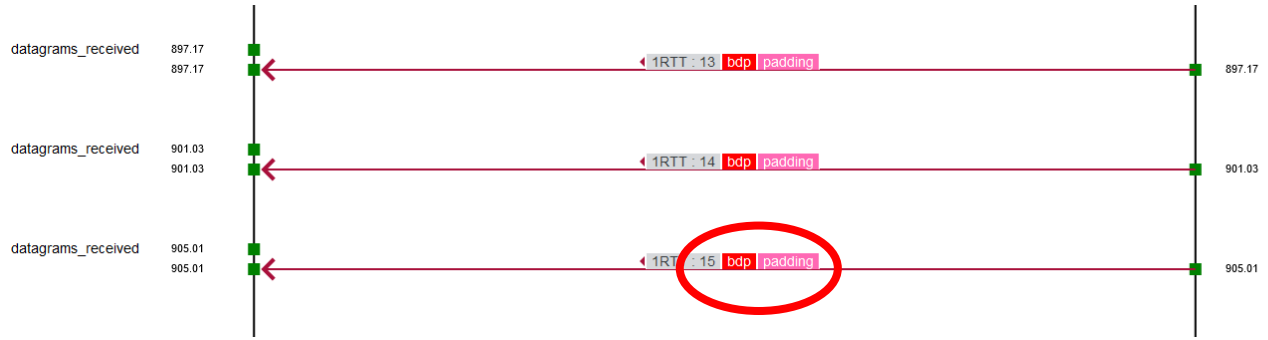
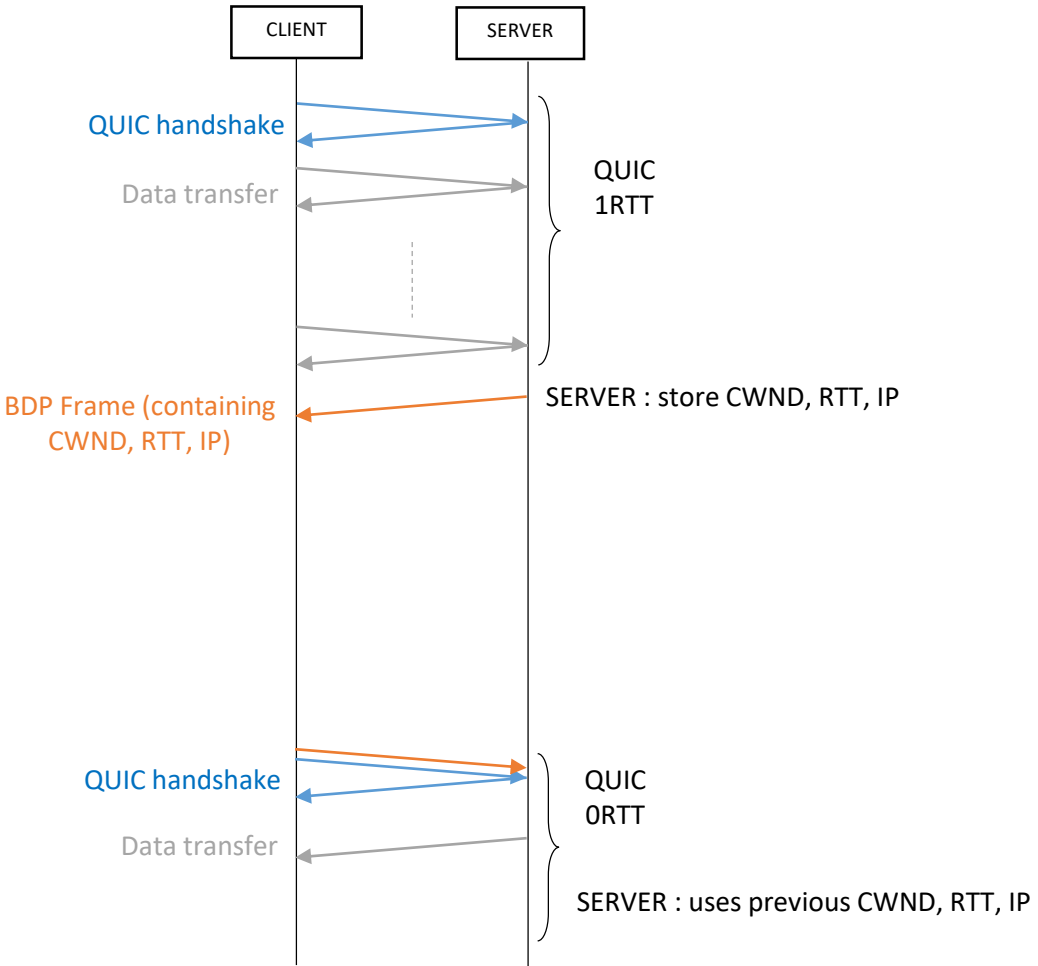
Using BDP_FRAME extension



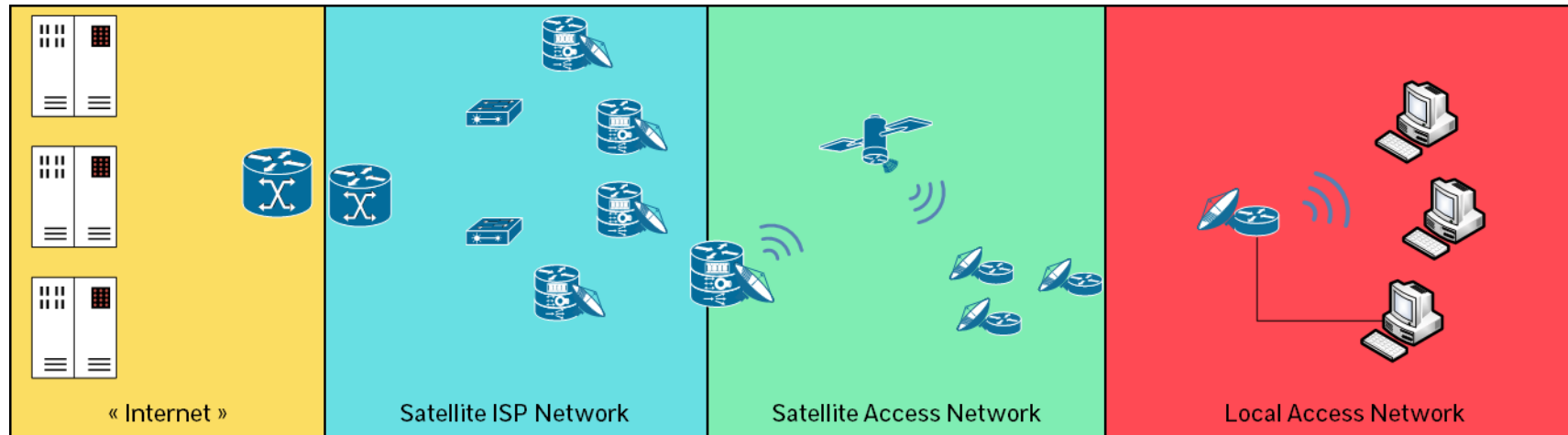
Local storage at the server



0-RTT-BDP extension : : QLOG example of QUIC 0RTTBDP connection with BDP_FRAME



0-RTT-BDP extension : Real satellite access Performance



PICOQUIC CLIENT

- Default and 0-RTT :
commit :
6b917d69bf4ac69d4ab43c425
54bb702ed844561
- ORTTBDP :
<https://github.com/private-octopus/picoquic/pull/1209>

Satellite : GEO - KaSAT
Offer : PRO25

PICOQUIC SERVER

- Default and 0-RTT :
commit :
6b917d69bf4ac69d4ab43c425
54bb702ed844561
- ORTTBDP :
<https://github.com/private-octopus/picoquic/pull/1209>

Upload 500 kB or 1MB



50 runs

0-RTT-BDP extension : Real satellite access Performance

	Without 0-RTT		With 0-RTT (and local storage at the server)		With 0-RTT-BDP	
	<p>CLIENT SERVER</p> <p>QUIC handshake</p> <p>Data transfer</p> <p>QUIC 1RTT</p>		<p>CLIENT SERVER</p> <p>QUIC 1RTT</p> <p>SERVER : store CWND, RTT, IP</p> <p>SERVER : « safely » uses previous CWND, RTT, IP</p> <p>QUIC 0RTT</p> <p>Data transfer</p>		<p>CLIENT SERVER</p> <p>QUIC handshake</p> <p>Data transfer</p> <p>QUIC 1RTT</p> <p>SERVER : store CWND, RTT, IP</p> <p>BDP Frame (containing CWND, RTT, IP)</p> <p>QUIC 0RTT</p> <p>SERVER : directly set new parameters to previous CWND, RTT, IP</p>	
	500 kB	1 MB	500 kB	1 MB	500 kB	1 MB
Min transfer time	3,12 s	3,87 s	2,43 s	3,19 s	1,87 s	2,78 s
Average transfer time	11,34 s	17,15 s	7,59 s	10,24 s	4,24 s	6,47 s
Max transfer time	47,82 s	61,43 s	33,69 s	33,92 s	15,55 s	23,88 s

0-RTT-BDP extension : Real satellite access Performance

	Without 0-RTT		With 0-RTT (and local storage at the server)		With 0-RTT-BDP	
	500 kB	1 MB	500 kB	1 MB	500 kB	1 MB
Min transfer time	3,12 s	3,87 s	2,43 s	3,19 s	1,87 s	2,78 s
Average transfer time	11,34 s	17,15 s	7,59 s	10,24 s	4,24 s	6,47 s
Max transfer time	47,82 s	61,43 s	33,69 s	33,92 s	15,55 s	23,88 s

Gain for a full 500kB or 1MB transfer (including HTTP GET)

0-RTT vs 1-RTT

up to 33% gain for 500 kB

gain due to the arrival of HTTP request one RTT in advance
(and large satellite RTT !)

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

Discussion

- Why tests are only run in upload direction ?
 - Because traffic was blocked (server ? ISP?) on the download direction
- Why tests are not run with terrestrial access ?
 - Because my terrestrial ISP was blocking traffic in both directions
- Is it safe to deploy the ORTTBDP draft ?
 - We could use it without « breaking » the Internet
 - QoS mechanisms are robust

Next steps

Status

draft-kuhn-quic-Ortt-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