

Deadline-aware Transport Protocol

draft-shi-quic-dtp-00

Hang Shi, Yong Cui, Zhiwen Liu

Tsinghua University



清華大學
Tsinghua University

Internet is becoming real time

- More and more applications has requirements for their data to arrive before a certain time i.e., deadline

Application	Deadline	Push/pull	Block	Priority
Video Conferencing	e.g. 400 ms	Push	Frame	audio > video
Cloud VR gaming	e.g. 25 ms	Pull	Foreground object, Background scene	killing > moving



Existing transport layer lacks support

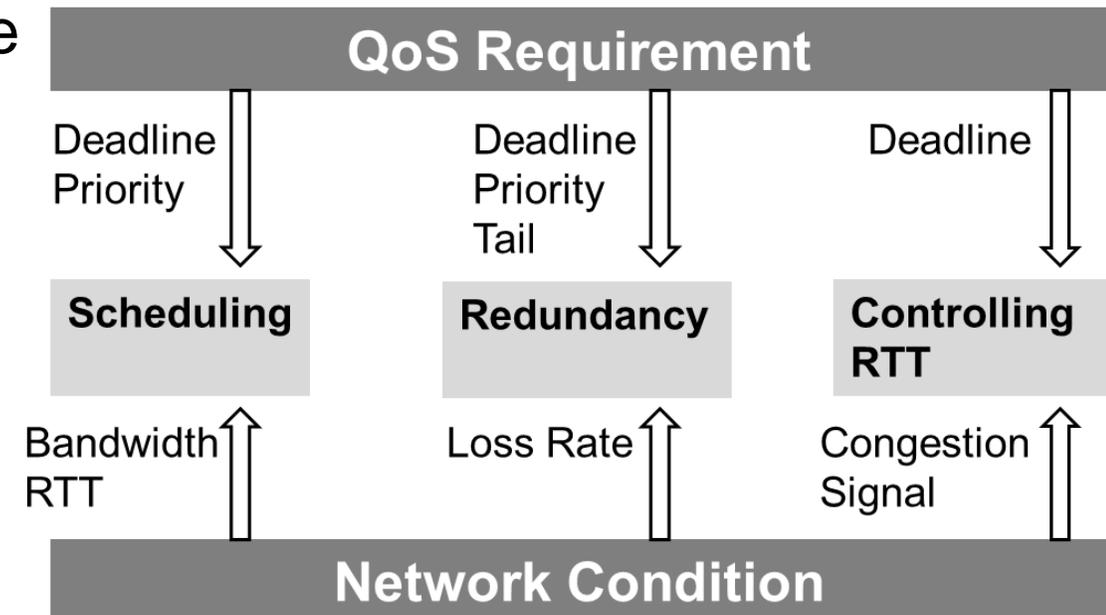
- deadline related protocols
 - SCTP, Timelined-TCP(IFIP 02), TCP Hollywood(IFIP 16)
 - hard to deploy
 - No optimization for deadline
 - No support for pull based transmission. How to build HTTP layer on top of deadline transport protocol?
- Apps are forced to build their own wheels(Salsify NSDI18, Redundant Data Bundling in games): Complex.
- We need a transport protocol to provide deliver-before-deadline service

Our solution: DTP

- Built on top of **QUIC**. Easy deployment
- Support both push and pull based transmission.
- Block instead of stream: Video frame, file chunk, html objects
- Application marks the block with dynamic deadline and priority

Architecture

- Scheduling block transmission order based on deadline, priority and network conditions
 - drop low priority and stale block
- Applying redundancy for tail packets to avoid retransmission delay when necessary
- Reducing queuing delay
 - congestion control: $RTT < \text{Deadline}$

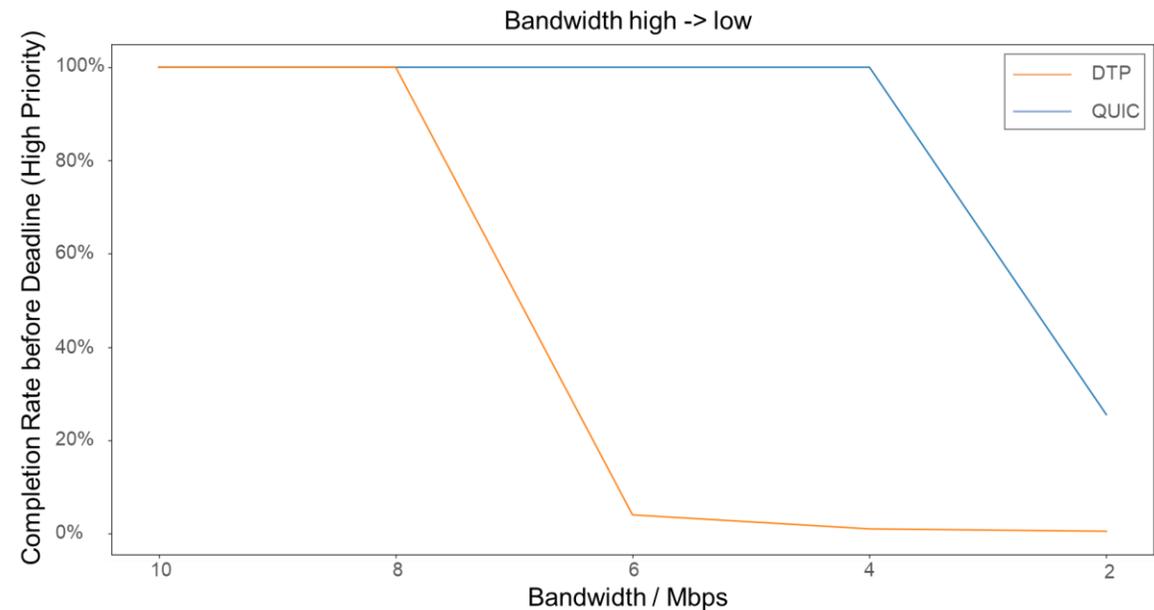
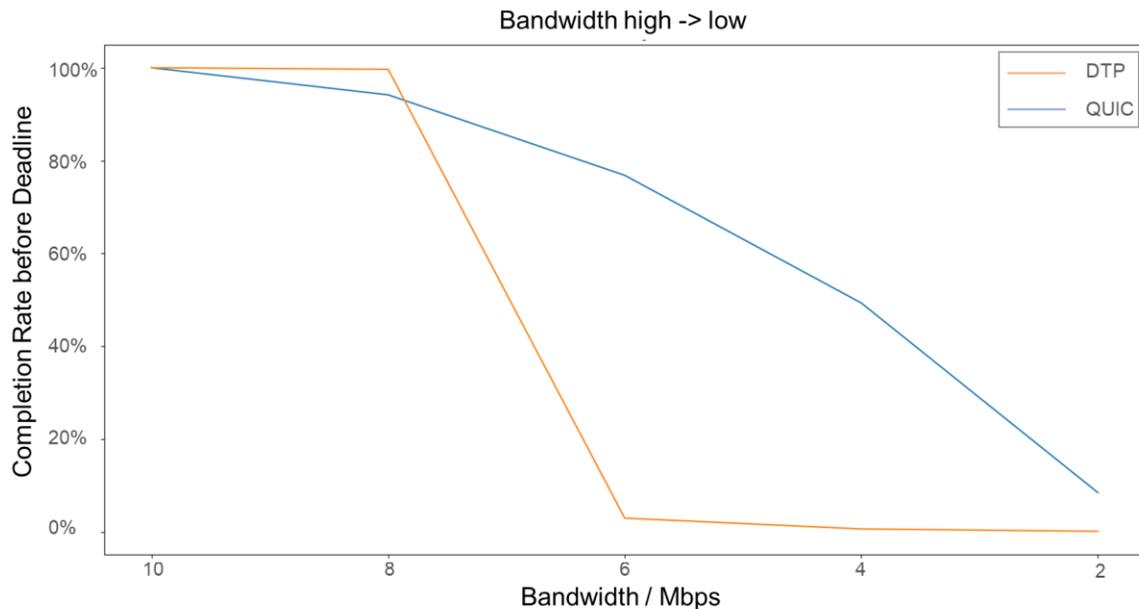
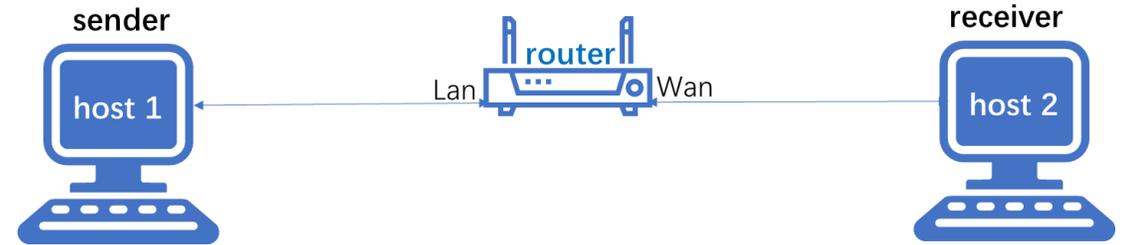


Extension to QUIC and HTTP/3

- 1 block = 1 HTTP3 stream = 1 QUIC stream(light weight)
- Add deadline and timestamp to STREAM frame
- When block is dropped by sender or receiver, close the stream and notify the other end by sending RESET_STREAM
- When QUIC stream is closed, notify HTTP 3 layer. HTTP 3 layer send HTTP CANCEL or HTTP REJECT to notify the other end.
- HTTP GET carry deadline and priority, server response call send with those metadatas.

Evaluation

- Implementation based on top of QUIC
 - Programming language is Rust
- Experimental design
 - Using TC to control bandwidth
 - Block sending rate is 6MB/s



Thank You!

- Looking for cooperators to develop applications on DTP
- If you are interested, please catch me during the week or email me at liu-zw16@mails.tsinghua.edu.cn
- Learn more about our work <https://star-tsinghua.github.io/DTP/>



清華大學
Tsinghua University