Exchanging Congestion Control Data in QUIC

draft-yuan-quic-congestion-data-03 | Bytedance × Akamai

Reporter: Jinghao Yuan

Why Expose CC Data?

One signal, Multiple uses

Careful Resume[1]
Adaptive Bitrate
Diagnostics & Path Visibility

Existing solutions

CMSD/CMCD[2]
QUIC BDP Signalling[3]

Why inside QUIC

Extract at source / Encapsulate once / Feedback for origin / Uses for variant applications

TP and Network Statistics

support_congestion_data(i)

CONSUME (0x01): Use the peer's congestion data on this connection.

CACHE (0x02): Store the peer's congestion data and return it later.

CONSIDER (0x04): Accept my previously exported data when it is returned.

Notes: From the TP sender's perspective

Network Statistics

Basic Unit of the new frames

Structured as TLV

```
{
  Type (i),
  Length (i),
  Value (..)
}
```

Type	Name
0xc8	Timestamp
0xca	Path Tuple
0xcb	Slow Start Status
0xcc	Network Type
0xcd	Maximum Congestion Window
0xce	Maximum In-Flight Data
0xcf	Smoothed RTT
0xd0	Minimum RTT
0xd1	RTT Variance
0xd2	Latest Bandwidth
0xd3	Maximum Bandwidth
0xd4	Throughput
0xd5	Send Rate
0xd6	Receive Rate
0xd7	Input Rate
0xd8	Loss Rate
0xd9	Buffer Length

Table 3

Two Frames, Two Roles

CONGESTION_DATA

{

Type (i) = TBD1,

Protected Count (i),

Protected Network Statistics (..) ...,

[Integrity Tag (1..)],

Unprotected Count (i),

Unprotected Network Statistics (..) ...,

}

Notes: Export current state and cacheable state

CONGESTION_DATA_RECALL

{

Type (i) = TBD2,

Protected Count (i),

Protected Network Statistics (..) ...,

Integrity Tag (1..),

}

Notes: Return protected state unchanged

Transmission Properties

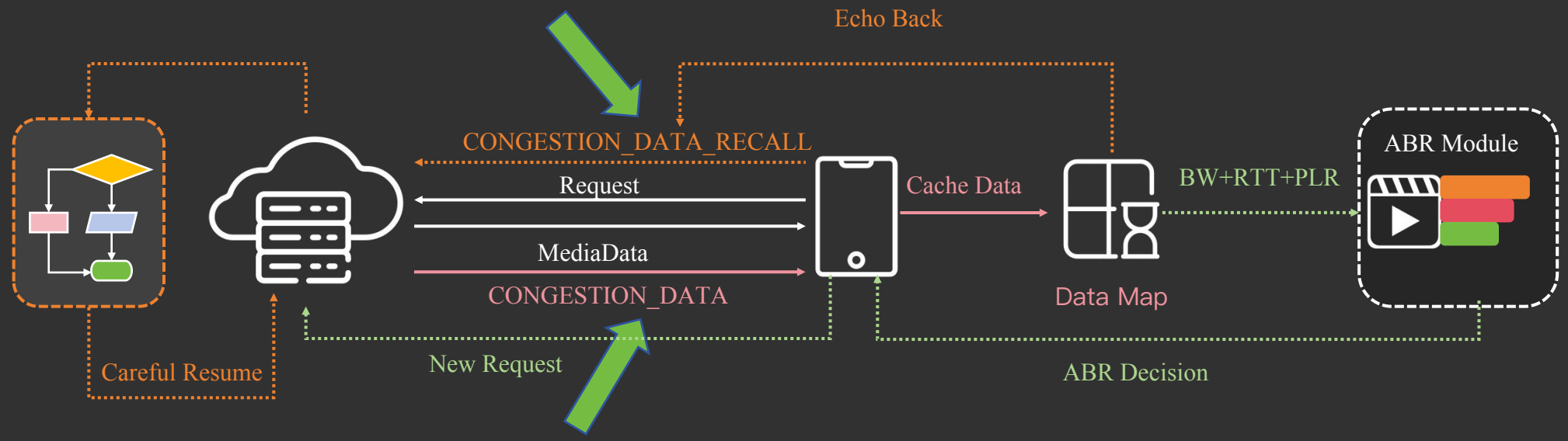
Common Properties

- 0-RTT or 1-RTT packets only
- ACK-eliciting
- Congestion controlled
- Not retransmitted when lost
- Not associated with a QUIC stream
- No ordered delivery

Frame-Specific Properties

- TP: `CONSUME` or`CACHE` vs `CONSIDER`
- Timing: Any time after keys are available vs As early as possible
- Frequency: Multiple frames allowed vs Once per connection
- On loss: MAY regenerate vs ingore
- Ordering: Use Timestamp to identify newer state vs Not applicable
- Purpose: Report current state vs Echo back

Application Use Cases

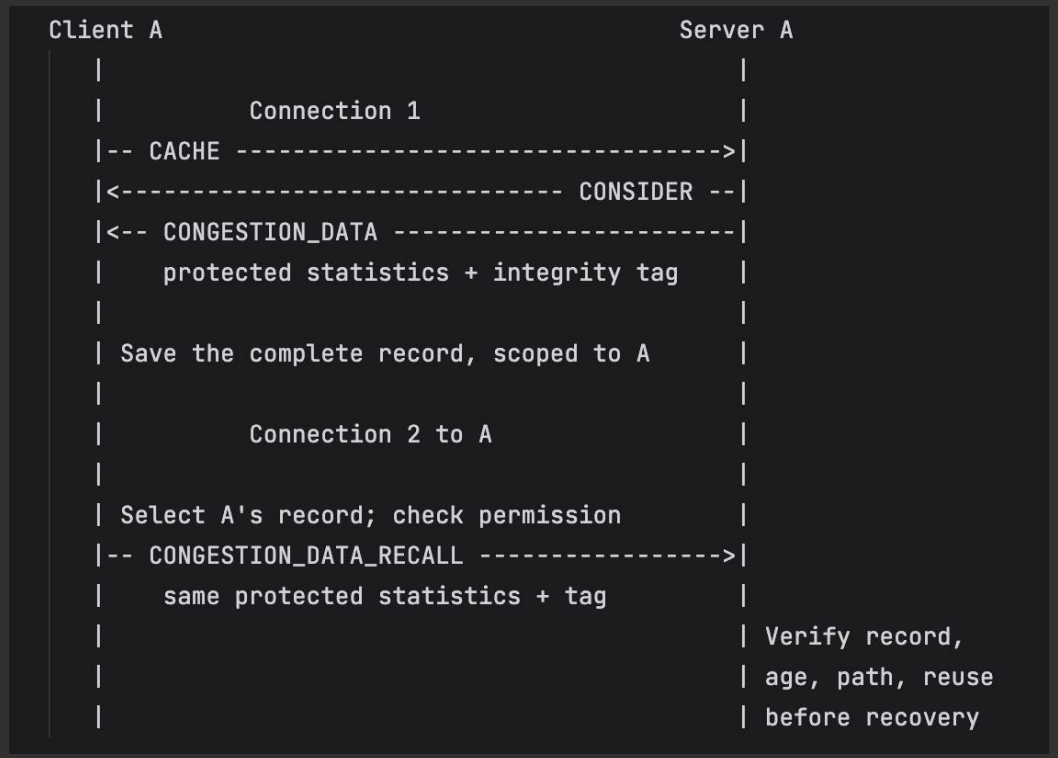


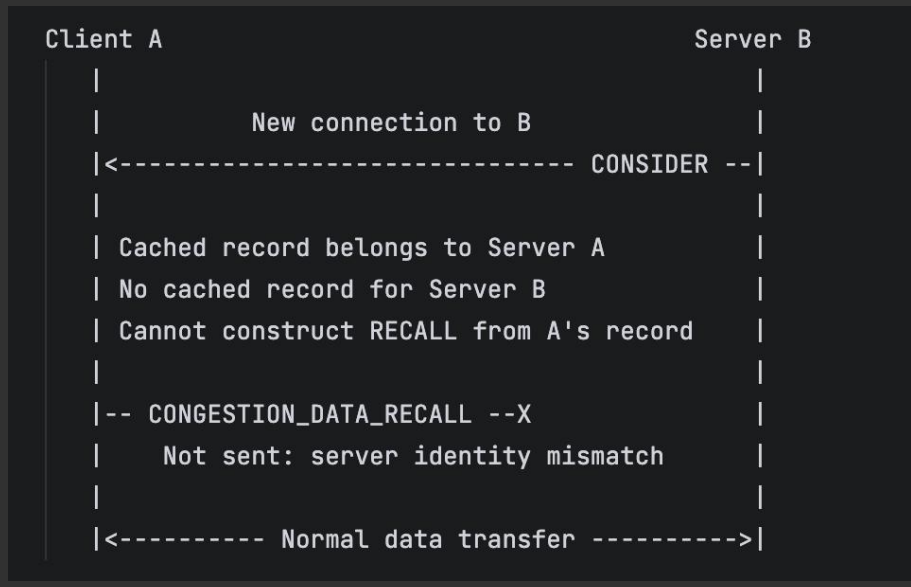
In TikTok implementation, we named CONGESTION_DATA frame as CCTK (Congestion Control ToKen) Frame

As for the recall direction, we use transport parameter as the channel.

Media → Congestion Data → Recall Logic → ABR Logic

Case A: Server-Side Congestion State Recovery





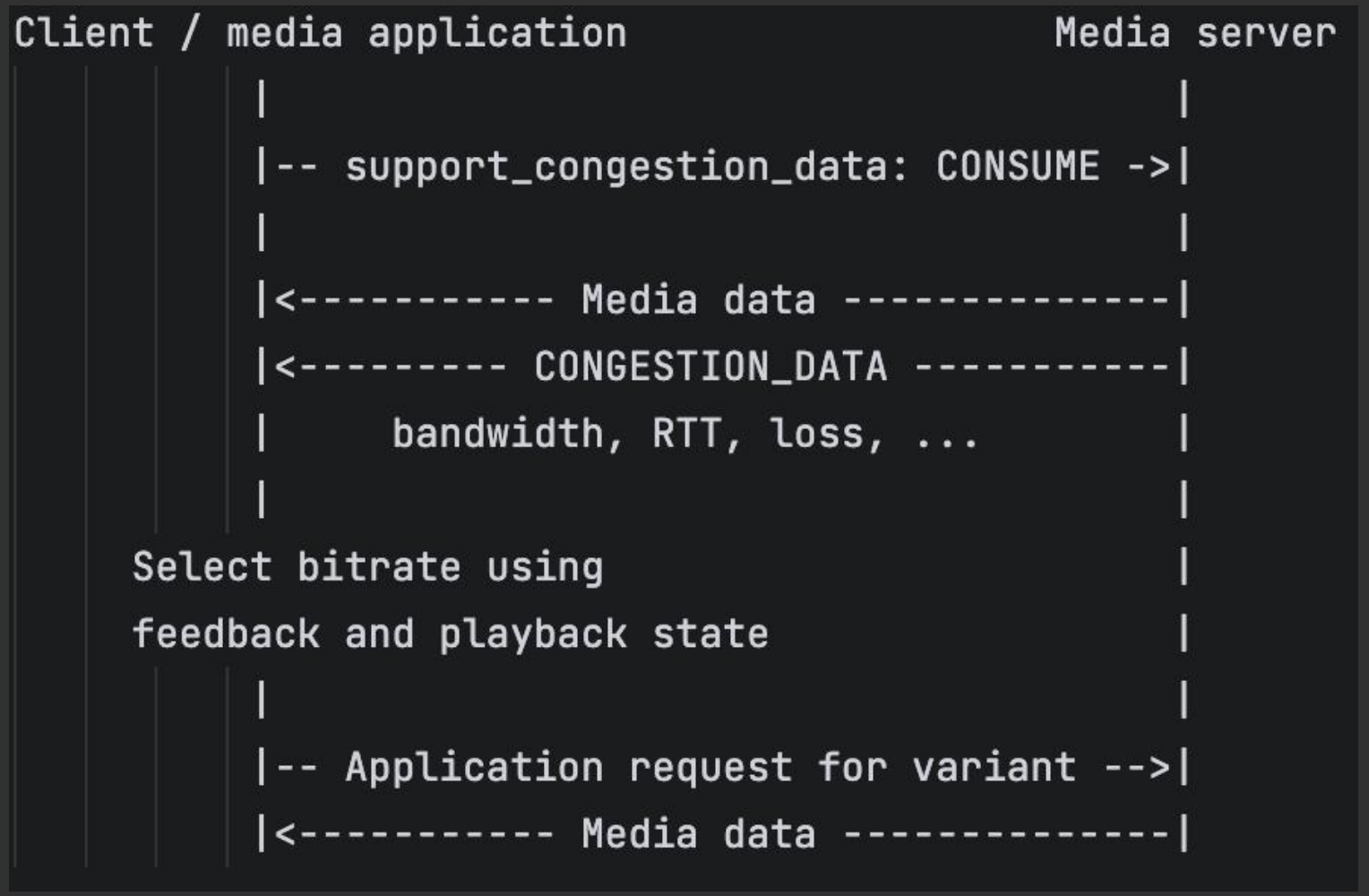
Data From TikTok Live

-0.19% First Frame Avg Cost	-2.20% Video Stall Rate	-2.75% Audio Stall Rate
-1.22% Stall Rate first 4 sec	-0.64% Video Stall Count	-4.98% Audio Stall Count

Each experiment group gathers data almost 1 Billion PV (Page View).

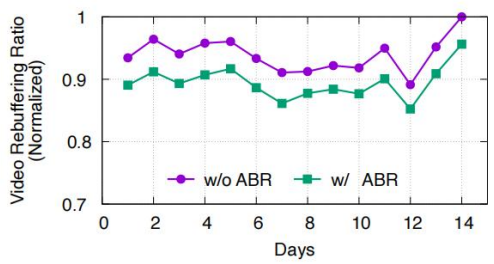
Share the same idea of **Careful Resume**, use the cached congestion control data from the same server.

Case B: Client-Side Adaptive Bitrate Streaming

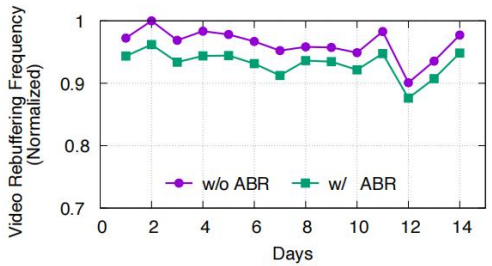


Data From TikTok Live

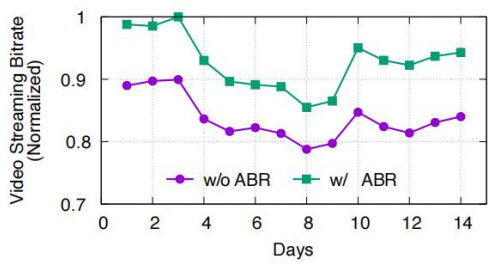
-4.7%	-3.3%	+10.8%
Stall Ratio	Stall Frequency	Streaming Bitrate



(a) Per-session video rebuffering ratio



(b) Per-session video rebuffer frequency



(c) Average streaming bitrate

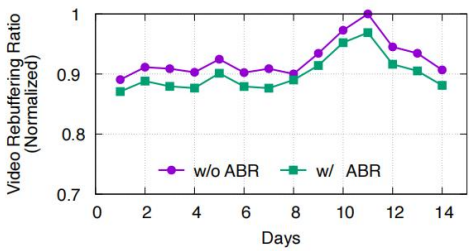
Figure 9: Overall QoE gains of ABR over HTTP-FLV

Notes: CCTK is the only but comprehensive network input for the ABR algorithm

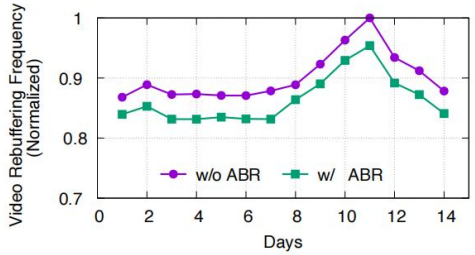
Data From TikTok Live

Good Conditions

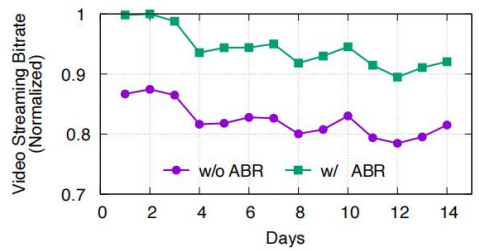
- Stall Ratio: - 2.6 %
- Stall Freq: -4.2 %
- Bitrate: +14.5 %



(a) Per-session video rebuffering ratio



(b) Per-session video rebuffer frequency

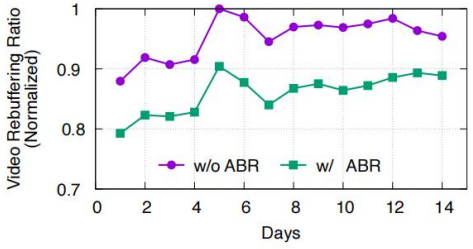


(c) Average streaming bitrate

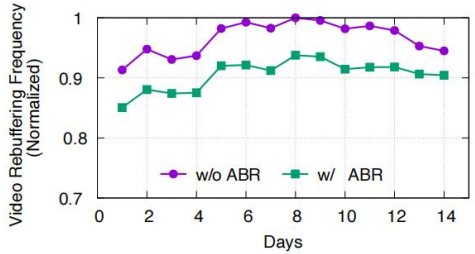
Figure 10: QoE gains of ABR over HTTP-FLV under "good" network conditions: >10 Mbps, < 50 ms RTT, < 0.2% loss

Poor Conditions

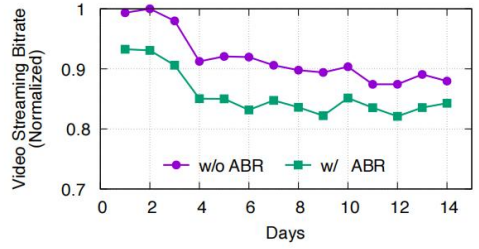
- Stall Ratio: -9.8 %
- Stall Freq: -6.3 %
- Bitrate: +6.6 %



(a) Per-session video rebuffering ratio



(b) Per-session video rebuffer frequency



(c) Average streaming bitrate

Figure 11: QoE gains of HTTP-FLV ABR in "weak" network conditions: [2, 5] Mbps, [50, 200] ms RTT, [0.5%, 5%] loss

Please refer to our paper[4] for more details.

Citation

1. [CAREFUL-RESUME] Kuhn, N., Stephan, E., Fairhurst, G., Secchi, R., and C.Huitema, "Careful Resume: Convergence of Congestion Control from Retained State", RFC 9959, DOI 10.17487/RFC9959, May 2026,<https://www.rfc-editor.org/rfc/rfc9959>

2. [CMCD] "Common Media Client Data (CTA-5004-B)", 2026, <https://cta-wave.github.io/Resources/common-media-client-data--cta-5004-b.html>.

3. [QUIC-BDP] Kuhn, N., Stephan, E., Fairhurst, G., Secchi, R., and C. Huitema, "Signalling CC Parameters for Careful Resume using QUIC", Internet-Draft draft-kuhn-quic-bdpframe-extension-05, March 2024, <https://datatracker.ietf.org/doc/draft-kuhn-quic-bdpframe-extension/05/>

4. [HTTP-FLV-ABR] Zhang, L., Meng, T., Lu, B., Yuan, J., Chen, L., Liu, H., Xiao, L., Wu, N., Sha, Z., Yan, C., Zhang, J., Kuang, J., and L. Song, "Adaptive Bitrate Live Streaming over HTTP-FLV: A Practical System Perspective", DOI 10.1145/3789240.3829169, August 2026, <https://doi.org/10.1145/3789240.3829169>. ACM SIGCOMM 2026 Conference.