Mysticeti-Style Total Ordering on the LP-208 DAG

Zach Kelling

Lux Industries

2026-06-03

Abstract

The LP-208 header DAG saturates throughput but is not yet a chain: there are M concurrent heads at round R, not one. We turn the DAG into a total-ordered sequence of blocks for execution via leader-anchored waves, adapted from Mysten Labs' Mysticeti [1] for Quasar's cert tiers and the ZAP buffer's MVCC unwind semantics. Every two DAG rounds an elected leader emerges (VRF or round-robin over the validator set). The leader's vertex anchors a wave; the wave commits all vertices in the leader's causal history; total order within the wave is causal sort with lexicographic tiebreak. One QuasarCert per wave, not per header. 2-RTT to wave-commit under $< 33\%$ byzantine; 3-RTT under $< 50\%$. End-to-end LAN finality at PQ-strict: $\sim$120 ms under the projected GPU-accelerated cert (LP-203 GB10 sm_120 pairing rates measured 2026-06-04: 49 197 pairings/sec at 19 μs/pair); $\sim$711 ms on the measured CPU-only cert (`/tmp/quasar-results-spark.txt` matrix bench Spark N=64, PQ-strict 595 ms per-cert wall-clock). This paper expands on LP-209 (`lps/LP-209-mysticeti-ordering.md`) with the wave-anchoring algorithm, the causal-commit invariants, the throughput and latency models, the explicit adaptations from the original Mysticeti construction, and the composition with LP-205, LP-210, LP-211, and LP-217.

Contents

1 Introduction

This paper expands on LP-209 (`lps/LP-209-mysticeti-ordering.md`) with detail on the wave-anchoring algorithm, the causal-commit invariants, the throughput and latency models, the explicit adaptations from the original Mysticeti [1] construction, and the composition with LP-205, LP-210, LP-211, and LP-217.

1.1 Why a wave is not a block

LP-208's DAG vertex is a header – author NodeID, round number, K parent references, payload commitment. A wave is a *commit unit*: the set of all DAG vertices in the causal closure of the anchor (the leader's vertex). One wave can commit thousands of headers in one step.

The block boundary moves up a tier. A "block" in LP-209's vocabulary is a wave's committed vertex set, totally ordered by the algorithm in Section 2. LP-210 executes the wave's txs in parallel under Block-STM; LP-217 cert covers the wave as one unit; LP-211 (cross-shard) coordinates waves across multiple chain-VMs.

1.2 Adaptations from the original Mysticeti

The original Mysticeti construction is a committee-based DAG ordering with uncertified DAGs ([1]). Three adaptations for this LP:

1. **Cert tiers replace single-tier finality.** Mysticeti produces one finality signature per wave commit. LP-209 instead composes the wave commit with the LP-217 cert ladder (PQ-off/PQ-fast/PQ-strict/PQ-heavy), so the cert leg composition per wave is the operator's chosen mode.
2. **ZAP buffer + ZapDB MVCC for unwind.** Mysticeti describes the ordering at the protocol layer; LP-209 anchors the ordering output in the LP-202 atomic-unwind contract so a wave that fails commits at a lower mode (LP-202 tier degradation) rather than blocking finality entirely.
3. **VRF leader election with PQ fallback.** Mysticeti describes leader election generically. LP-209 specifies BLS-VRF for the classical case (LP-075) and Pulsar-threshold-derived randomness (LP-176) under PQ-heavy where BLS is dropped.

The core mechanism – leader-anchored waves, causal sort, lexicographic tiebreak, 2-RTT fast-path quorum – is the original Mysticeti construction. The composition with Quasar is the adaptation.

1.3 What this LP solves

LP-208 fixes throughput by inverting the mempool from min-bounded to sum-bounded. But the LP-208 DAG is not yet a chain – the consumer needs a total order over the DAG to feed an execution engine. LP-209 supplies that total order with:

- 2-RTT to wave-commit under $< 33\%$ byzantine (fast path)
- 3-RTT under $< 50\%$ byzantine (recovery path)
- One QuasarCert per wave (not per header), sized to the operator's LP-217 mode
- $\sim$40 000 tx/wave commit at the LAN protocol floor (40-validator chain, K=50 ms round cadence, 500 tx/header)

2 Algorithm

2.1 Round structure

Rounds in the DAG advance at a fixed cadence – every header references K parents from the prior round (LP-208 round structure). Waves are anchored at even rounds $(2, 4, 6, \ldots)$; leader election runs every 2 rounds.

2.2 Leader election

At even round $2k$, the leader is selected via VRF over the round digest:

$$\text{leader}(2k) = \arg\max_{v \in \text{validators}} \text{VRF}(v.\text{staking\_key}, \text{round\_digest}(2k)) \tag{1}$$

where $\text{round\_digest}(r) = \text{sha256}(\text{chain\_id} \| r \| k_{\text{prev\_round\_root}})$.

The VRF is BLS-based (LP-075 BLS leg) for the classical case; under PQ-heavy the VRF falls back to a Pulsar threshold-derived randomness beacon (LP-176). Round-robin is the deterministic fallback when no validator publishes a VRF proof within the round timeout: $\text{leader}(2k) = \text{validators}[(2k/2) \bmod N]$.

2.3 Wave anchoring

At the end of round $2k{+}1$, every validator examines the DAG and decides: *did the leader's round-$2k$ vertex receive $\geq 2f{+}1$ references in the round-$2k{+}1$ layer?*

- **Yes** $\rightarrow$ the leader vertex is the wave anchor. The wave commits.
- **No** $\rightarrow$ the wave skips. The leader of round $2k{+}2$ anchors a larger wave covering rounds $2k..2k{+}2$.

The $\geq 2f{+}1$ references condition is the Mysticeti fast-path quorum under $< 33\%$ byzantine. Under $< 50\%$ byzantine (recovery path), 3-RTT vote counting on round $2k{+}2$ votes for round-$2k{+}1$ votes for round-$2k$ anchors the wave at the cost of one extra round.

2.4 Causal commit set

Once the anchor is decided, the wave's commit set is the causal closure of the anchor in the DAG:

$$W(\text{anchor}) = \big\{v \in \text{DAG} : v \in \text{causalAncestors}(\text{anchor})\big\} \cup \{\text{anchor}\} \setminus \bigcup_{j<k} W(\text{prior\_anchor}_j) \tag{2}$$

Subtract prior anchors' commit sets – a vertex is committed exactly once, by the earliest wave whose anchor causally dominates it.

2.5 Total order within wave

The wave's vertices are totally ordered by:

1. **Round number ascending** (vertices in lower rounds order first).
2. Within a round, **lexicographic on `(NodeID, headerHash)` ascending**.

This is the canonical sort. Every validator independently computes the same order from the same DAG – no coordination needed. The order is deterministic and content-addressable.

2.6 Cert ceremony

Once the wave is ordered, the LP-217 cert ceremony runs on the wave's ordered byte payload (concatenation of header bytes in order). One QuasarCert per wave, signed by $\geq 2f{+}1$ validators per LP-182. The cert's leg composition is per the chain's configured `quasar.cert_mode` (LP-217); the cert's `blockID` field is sha256(wave_ordered_bytes).

2.7 Stack composition

```
LP-208 DAG mempool produces headers (continuous)
     |
     v
LP-209 wave commit (every 2 rounds; leader-anchored)
     |
     v
LP-210 Block-STM execution (parallel; per-tx MVCC)
     |
     v
LP-217 cert ceremony (one cert per wave)
     |
     v
ZapDB MVCC commit (LP-202 atomic unwind)
```

LP-209 never re-orders within a header – the tx order inside one header is the header author's choice, sealed at LP-208 publish time. LP-209 only orders *across* headers, by the rule above.

3 Throughput and Latency

Reference numbers from a 40-validator LAN at K=4 parents per header, T=500 txs per header, round cadence 50 ms.

3.1 Throughput

Quantity	**Value**	**Source**
Headers per wave (typical)	$\geq 40 \times 2 = 80$	every validator publishes 2 headers per wave
Txs per wave	$80 \times 500 = 40{,}000$	header count × tx count
Wave cadence	50 ms ×2 = 100 ms	2 rounds per wave
Waves/sec at cadence floor	10	1/100 ms
Txs/sec at cadence floor (1 wave)	400 ktx/s	40k × 10
Txs/sec at 3-deep pipelining (LP-205)	1.2 Mtx/s	3× pipeline depth

Table 1: Throughput at the LAN protocol floor. **Projected, not measured** – assumes the projected GPU-accelerated cert path of LP-203 holds the wave cadence below 100 ms. The 2026-06-04 quasar-matrix-bench measured CPU-only cert wall-clocks (Spark N=64: PQ-fast 585 ms, PQ-strict 595 ms, PQ-heavy 1326 ms, `/tmp/quasar-results-spark.txt`) exceed the 100 ms cadence in every PQ row, so this floor is achievable only with GPU offload (LP-203 wrappers, pending).

The 1.2 Mtx/s figure is the projected protocol-floor throughput at LAN scale under GPU-accelerated cert. On the measured CPU-only cert path, the wave cadence is cert-bottlenecked at ∼585 ms (PQ-fast Spark N=64) or higher, so aggregate at 3-deep pipelining drops to ∼205 ktx/s. At WAN scale (∼150 ms RTT, ∼50 ms round cadence held by overlap gossiping) the wave cadence rises to ∼300 ms; aggregate falls to ∼400 ktx/s (projected GPU path) or stays cert-bottlenecked at ∼205 ktx/s (measured CPU path) – still saturates the LP-210 per-validator execution budget.

3.2 Latency

Path to finality for a tx admitted at time $t = 0$. Two reference columns: the projected GPU-accelerated cert path and the 2026-06-04 matrix-bench measured CPU-only cert path.

Event	GPU proj.	CPU meas.	Path
Tx admitted to mempool	0 ms	0 ms	LP-208 tx receive
Header containing tx published	$\leq$ 25 ms	$\leq$ 25 ms	LP-208 round cadence, half-round budget
Header referenced by $\geq 2f+1$ in next round	$\leq$ 75 ms	$\leq$ 75 ms	+ 1 round (50 ms cadence)
Wave anchor decided (round $2k+1$)	$\leq$ 100 ms	$\leq$ 100 ms	+ 1 round to count references
Wave ordered + LP-210 exec begins	$\leq$ 105 ms	$\leq$ 105 ms	causal sort $O(\lvert W \rvert)$
LP-210 exec complete (40k tx batch, 16-core)	$\leq$ 115 ms	$\leq$ 115 ms	$\sim$10 ms for 40k txs
LP-217 cert at PQ-strict	$\leq$ 120 ms	$\leq$ 710 ms	GPU: + $\sim$5 ms; CPU: + $\sim$595 ms (matrix bench Spark N=64)
ZapDB commit	$\leq$ 121 ms	$\leq$ 711 ms	LP-202 atomic commit, $O(1)$ at MVCC layer

Table 2: End-to-end LAN latency to PQ-strict finality. GPU column is projected (LP-203 pairing-rate measurement extrapolated); CPU column is grounded in the 2026-06-04 quasar-matrix-bench cert wall-clock (`/tmp/quasar-results-spark.txt`, N=64, PQ-strict row).

Summary. End-to-end LAN finality at PQ-strict: $\sim$120 ms (**projected** GPU-accelerated path), $\sim$711 ms (**measured** CPU-only path, 2026-06-04 matrix bench Spark N=64). At PQ-heavy: projected GPU adds $\sim$80 ms for Magnetar $\rightarrow$ $\sim$200 ms; measured CPU adds 1326 ms (matrix bench Spark N=64 PQ-heavy row) $\rightarrow$ $\sim$1441 ms. Under $<$ 50% byzantine (3-RTT path), add one extra round ($\sim$50 ms).

3.3 Honest reading of the projections

Measured cert-leg numbers. The earlier 2026-04 multi-validator harness (`/tmp/multi-validator/result-` recorded N=100 validators with Pulsar p50 = 297.5 ms, BLS p50 = 161.6 ms, Corona p50 = 2593 ms under the Hybrid profile on CPU. The 2026-06-04 quasar-matrix-bench on Spark at N=64 refines this with the full 7-cert-profile sweep: PQ-off 6.3 ms, PQ-fast 585 ms, PQ-strict 595 ms, PQ-heavy 1326 ms per-cert wall-clock. Both rounds are CPU-only; GPU was not exercised (mean SM%=11.9%).

Projected GPU-accelerated numbers. The LP-203 GB10 sm_120 BLS12-381 pairing measurement (49 197 pairings/sec, 19 μs/pair, 4 W) and the keccak256 batch (11.38 Mops/s) are measured and operational. With the Pulsar / Corona verify wrappers wired in (currently NotSupported), the projected PQ-fast GPU-accelerated cert wall-clock is $\sim$5 ms; PQ-strict $\sim$15 ms; PQ-heavy $\sim$80 ms. These remain projections until the wrappers ship.

Throughput-floor sensitivity. The Table 1 1.2 Mtx/s number assumes the projected GPU path. On the measured CPU-only path, throughput is cert-bottlenecked: PQ-fast yields 1/585 ms $\times$ 40k txs/wave $\times$ 3 (pipeline) $\approx$ 205 ktx/s, not 1.2 Mtx/s. The pipelining and DAG-throughput claims hold structurally; GPU offload (LP-203 wrappers) closes the 6$\times$ gap.

4 Failure Modes

4.1 Safety property

The Mysticeti safety property holds: at most one wave anchor is committed per even round across the entire honest quorum. Mysticeti [1] proves the safety theorem in the partial-synchrony model; LP-209 inherits the proof, with the only adaptation being that the cert ceremony following the ordering is the LP-182 QuasarCert rather than a single-leg BFT cert.

Failure	Trigger	Effect	Recovery
Leader missed (no header at round $2k$)	leader offline, partition, equivocation suppressed by reputation	wave skips; round $2k+1$ votes do not anchor	next leader at $2k+2$ anchors a 4-round wave covering $2k..2k+2$
Leader equivocates (two distinct headers at $2k$)	byzantine	DAG retains both; neither receives $\geq 2f+1$ references (Mysticeti slashing)	wave skips; next $2k+2$ leader anchors
$< 2f+1$ references on leader vertex	partition, slow validators	fast-path quorum not met	3-RTT recovery: round $2k+2$ votes for round $2k+1$ votes for round $2k$
Cert leg fails to aggregate (LP-217)	insufficient threshold sigs	cert tier downgrade per LP-202	relying parties at the mode wait or fall back; consensus liveness preserved
LP-210 execution fails on a wave tx	sig invalid, balance underflow	per-tx MVCC rollback; other txs commit	invalid tx dropped; commit proceeds
Partition splits validators	dual halves cannot reach $\geq 2f+1$	both halves stall at anchor decision; no double-commit (Mysticeti safety)	partition heals; quorum-side advances
Reorg at committed wave	impossible	n/a	Q-Chain finality (LP-079) + LP-217 cert PQ-strict+ is content-addressed; no alternative ordering produces same cert

Table 3: Failure modes and recovery.

4.2 Liveness under partition

Under network partition $> 1/3$ stake, the wave anchoring halts: neither half can produce $\geq 2f+1$ references on the leader vertex. When the partition heals, the half that resumes with $\geq 2f+1$ advances; the other half catches up via the LP-202 bootstrap pipeline. No double-commit is possible because each honest validator can be on only one side of the partition, and the $2f+1$ quorum cannot have members from both sides simultaneously in a valid validator-set partitioning.

5 Composition With Sibling LPs

5.1 With LP-205 pipelining

LP-205 specifies block production pipelining at depth 3. LP-209 extends this to the wave layer:

The pipelining-depth cap is the LP-217 cert mode's required-leg slowest arrival – at PQ-strict $\sim$5 ms is far below the 100 ms wave cadence, so pipeline depth saturates at the wave cadence itself. At PQ-heavy $\sim$50 ms is still within the 100 ms cadence, so 2–3 deep pipelining holds.

5.2 With LP-208 DAG mempool

LP-208 produces the substrate; LP-209 consumes it. The contract:

LP	Composition
LP-020	TEE extensions (validator TEE attestation for VRF leader-election integrity under PQ-heavy)
LP-075	BLS consensus leg (the VRF source under classical and PQ-fast/PQ-strict modes)
LP-079	Q-Chain finality blocks (where wave commits land)
LP-176	DRBG randomness beacon (Pulsar-threshold-derived VRF fallback under PQ-heavy)
LP-182	Consensus wire / QuasarCert (one cert per wave)
LP-202	ZAP Pipelining and Atomic Unwind (cert leg degradation contract; MVCC unwind on cert failure)
LP-203	GPU-native verify (source of the projected GB10 sm_120 cert wall-clocks: BLS pairing 19 μs/pair, keccak256 11.38 Mops/s)
LP-204	Network of Blockchains (per-L1 LP-209 instantiation)
LP-205	Pipelined block production (three waves in flight = 3$\times$ throughput)
LP-208	DAG mempool / header DAG (the substrate LP-209 orders)
LP-210	Block-STM optimistic parallel execution (per-wave per-tx parallelism)
LP-211	Cross-shard atomic Polaris cert (cross-VM wave composition)
LP-217	Cert Profile Modes (per-wave QuasarCert profile)

Table 4: Composition of LP-209 with sibling LPs.

5.3 With LP-211 cross-shard

When state is sharded across multiple chain-VMs (LP-204 Tier-2), each shard runs its own LP-208/209 stack. A cross-shard tx group is committed atomically by LP-211's cross-shard cert at the parent L1 – the per-shard wave commits provide each shard's prepare-ack; the parent L1's wave commits the cross-shard cert.

This means LP-209's "wave" is per-shard; LP-211's "cross-shard commit" is a parent-L1 wave that references prepare-acks from N shard waves. The parent L1's wave latency is the max of the per-shard wave latencies plus the LP-217 cert at the parent L1.

6 Activation Marker

```
activates:      2025-12-25T16:20:00-08:00
activates-unix: 1766708400
```

Applies to the wave-ordering layer from the genesis block of the new final Lux network onward.

7 Backwards Compatibility

None. The pre-Quasar Edition Lux network's serial block production is out of scope.

References

[1] Kushal Babel, Andrey Chursin, George Danezis, Anastasios Kichidis, Lefteris Kokoris-Kogias, Arun Koshy, Alberto Sonnino, and Mingwei Tian. Mysticeti: Reaching the limits

Wave	$t{=}T$	$t{=}T{+}100\,\mathrm{ms}$	$t{=}T{+}200\,\mathrm{ms}$
W_k	LP-210 exec	LP-217 cert + commit	settled
W_{k+1}	causal sort + ordering	LP-210 exec	LP-217 cert
W_{k+2}	round $2(k{+}1){+}1$ vote count	causal sort + ordering	LP-210 exec
W_{k+3}	rounds $2(k{+}2), 2(k{+}2){+}1$ in DAG	round $2(k{+}2){+}1$ vote count	causal sort

Table 5: Four waves in different stages at any wall-clock moment.

LP-208 produces	**LP-209 consumes**
ZAP-framed headers, schema `0xF0`	DAG vertex set
K parent references per header	causal-ancestry graph
Round number per header	wave-eligibility filter (round mod 2)
Author NodeID per header	leader-election membership check
Payload commitment (sha256 over txs)	wave-ordered bytes for cert
Equivocation detection (two headers same round, same author)	wave anchor disqualification

Table 6: LP-208 / LP-209 contract.

of latency with uncertified DAGs, 2024. Mysten Labs research; https://arxiv.org/abs/2310.14821.