

## 0.1 Reversible Transactions over a Single Ethernet Link

Adapted from ./AE-Specifications-ETH/@GPT-reversibility.tex

Traditional transaction protocols over Ethernet assume a forward-only time model. Failures are handled through retries, timeouts, or speculative execution. This results in complexity, energy waste, and difficulty in verification. Inspired by physics, we propose a model in which every Ethernet transmission is reversible: a transaction is not committed until its inverse is impossible.

### Vector-Based Framing

Rather than treating Ethernet frames as opaque byte sequences, we treat them as elements of a vector space over  $\mathbb{F}_2^n$ . This allows each transmission to be encoded as a linear transformation, and, crucially, inverted.

- A single frame becomes a vector  $v \in \mathbb{F}_2^n$ .
- A transaction is a linear operation  $T : v \mapsto Tv$ .
- Inverse operations  $T^{-1}$  can be issued on failure, restoring prior state.

### Two-Phase Semantics

Each transaction operates in two stages:

#### *Phase 1: Tentative Forward Transmission*

The initiator sends a vector transformation. The responder buffers the transformation but does not apply it semantically until Phase 2.

#### *Phase 2: Acknowledgment or Inversion*

If successful, a commit acknowledgment is returned. If failure or timeout, the initiator issues  $T^{-1}$ , instructing the responder to reverse the tentative state.

### Benefits of Reversibility

- **Verifiability:** All transactions are algebraically closed and auditable.
- **Composability:** Compound transactions can be expressed as chains of linear transformations.
- **Energy Efficiency:** Rollbacks avoid speculative execution and wasted computation.
- **No Silent Failures:** The absence of acknowledgment implies an outstanding transformation requiring cleanup.

### Implementation Implications

Æthernet links may cache a small rollback buffer per transaction and implement an algebraic acknowledgment loop. This loop behaves as a

“circular snake” — with the head and tail of a transaction circulating on the link until either:

- The transaction is committed (the tail is consumed).
- The transaction is explicitly reversed (the tail eats the head).

This model supports deterministic, lossless computing over unreliable mediums by ensuring all state transitions are recoverable — a local time-symmetric alternative to global distributed consensus.

### Mathematical Formulation

Let  $\mathbf{v} \in \mathbb{F}_2^n$  represent the contents of an Ethernet frame, modeled as an  $n$ -dimensional vector over the binary field. Each transaction is a linear transformation:

$$T : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n, \quad \mathbf{v} \mapsto T\mathbf{v}$$

Here,  $T$  is an  $n \times n$  binary matrix with full rank (i.e.,  $\det(T) \neq 0$  in  $\mathbb{F}_2$ ), ensuring invertibility:

$$T^{-1}T\mathbf{v} = \mathbf{v}$$

Each Ethernet transaction must store the pair  $(T, \mathbf{v})$  temporarily, enabling the rollback operation:

$$\mathbf{v}' = T\mathbf{v} \quad \text{and} \quad \text{on failure: } T^{-1}\mathbf{v}' = \mathbf{v}$$

This structure creates a *reversible pipeline*, where only acknowledged transformations become final state.

### Transaction Algebra and Composition

Multiple transactions may be composed sequentially:

$$\mathbf{v}_2 = T_2(T_1\mathbf{v}_0) = (T_2T_1)\mathbf{v}_0$$

To rollback to any previous stage, it suffices to apply the inverse of the composed transformation:

$$\mathbf{v}_0 = (T_2T_1)^{-1}\mathbf{v}_2 = T_1^{-1}T_2^{-1}\mathbf{v}_2$$

This algebraic chaining enables deterministic state replay and rollback — suitable for verifying transaction histories without logs.

### Interpretation as Homology

A reversible link can be seen as enforcing *cycle closure*. That is, any sequence of transformations  $T_1, T_2, \dots, T_k$  across the link must eventually yield:

$$T_k \dots T_2 T_1 = I \quad (\text{identity})$$

unless a semantic commit is agreed upon and the cycle is broken. This makes every transaction loop homologically equivalent to zero until commitment. The state machine on each end must therefore enforce that:

$$\sum_{i=1}^k T_i \in \ker(\text{Commit})$$

until an acknowledgment collapses the pending cycle into permanent change.

### Reversibility as a Link Invariant

Just as error correction preserves information despite physical noise, reversible algebra ensures *semantic reversibility* across potentially unreliable hardware. This enables a protocol invariant:

$$\forall t, \quad \text{State}_{\text{link}}(t) \in \text{Image}(\mathcal{T}^\pm)$$

where  $\mathcal{T}^\pm$  is the group of invertible transformations. All observable state transitions fall within this reversible group action.

### Acknowledgment as Projection

Acknowledgment is not merely a signal but a *projection operator*  $\mathcal{P}$  that collapses the reversible superposition of possible futures into a committed state:

$$\mathcal{P}(T\mathbf{v}) = \mathbf{v}_{\text{committed}}$$

Until this projection is received, the state must be maintained in an entangled (reversible) buffer space — a form of local decoherence management.

### Extending Reversibility to Graphs

While the reversible transaction model began with point-to-point (1D) links, it naturally generalizes to multi-node graphs.

*Spanning Tree for Determinism* To maintain deterministic reversibility in a graph of  $n$  nodes, define a spanning tree  $T_s$  rooted at a node  $r$ . Each transaction along edge  $(i, j)$  becomes a matrix  $T_{ij}$  with:

$$T_{ij}^{-1}T_{ij} = I$$

The system maintains a log of traversed edges to support backward computation if recovery is triggered.

*Multi-path and DAG Tensions* Acyclic paths support clean reversibility. In contrast, fully connected graphs ( $K_n$ ) or cyclic DAGs may complicate rollback semantics unless time-bounded checkpoints are defined. Reversibility can be ensured by constraint:

$$\text{Path}(r \rightarrow t) \in \text{Tree}_{\text{epoch}}$$

Where  $\text{Tree}_{\text{epoch}}$  rotates to provide redundancy without losing invertibility guarantees.

### Valency Constraints and Physical Limits

Most physical Ethernet or SerDes ports limit a node's degree to 8 or fewer. As such, highly connected graphs ( $K_n$  for  $n > 9$ ) are infeasible. Instead, the protocol supports:

- **Hypercube Topologies:** Efficient for uniform, fault-tolerant routing under degree constraints.
- **Torus and Mesh Graphs:** Enable deterministic reversibility and spatial locality.
- **Tree-Stacked Subgraphs:** Spanning trees with redundant shadows for failover support.

### Dynamic Laplacian Resilience (DLR)

We define a resilience metric for reversible Ethernet graphs under link failures:

$$\text{DLR}(G) = \frac{1}{k} \sum_{i=1}^k \lambda_2(G_i)$$

where  $G_i$  is the graph after  $i$  failures and  $\lambda_2$  is the second-smallest eigenvalue of the Laplacian matrix (algebraic connectivity). High  $\lambda_2$  implies robust path redundancy and invertibility.

### Snake-Based Circulation Model

In the reversible framing model, the "packet" and its acknowledgment together form a single logical object — a snake of bits that must eventually return to its source or be rolled back. Let:

- $\text{Head}(T\mathbf{v})$  be the transmitted transformation.
- $\text{Tail}(T^{-1})$  be the reversible closure.

Completion occurs when both ends of the transformation exist and collapse:

$$\text{Commit}(T\mathbf{v}) \iff \text{Tail returns within timeout and matches } T^{-1}$$

This model avoids deadlocks and waste. Only transformations that complete the loop can semantically alter state.

### Time Symmetry in Protocol Design

All operations are temporally bidirectional unless explicitly committed. This mirrors time symmetry in physics and allows for:

- **Exact Rollbacks:** Even at multi-hop or multi-frame granularity.
- **Auditable Transactions:** Linear algebra ensures closure and reconstructibility.
- **Semantic Isolation:** No side effects are propagated until loop closure.

### Implications for Reversible Hardware

The reversibility model has strong implications for hardware design:

- **Frame Buffers:** Temporarily store  $(T, \mathbf{v})$  until acknowledgment or rollback.
- **Commit Logic:** Frame commits must be locally verifiable by algebraic identity.
- **Link Loops:** The head and tail of a transaction can circulate on the same physical link, forming a tight loop.

In short links (e.g., chiplets), the "snake" can be fully contained in the round-trip delay of the cable. This permits sub-microsecond full transaction turnaround without global arbitration.

### Conclusion: Reversibility as a First-Class Network Property

Reversible computation has long been a topic in theoretical computer science, but with  $\mathcal{A}$  Ethernet, it becomes a foundational property of the link layer. The implications are:

1. **All state transitions are reversible until observed.**
2. **Transactions are modeled as group operations with inverse semantics.**

### 3. Deterministic rollback replaces speculative retry.

This model positions Æthernet as the first Ethernet protocol that enforces reversibility as a *physical layer invariant*, enabling full-stack semantic safety — from API to transceiver.