Project Report – EXO (FDI-2025)

EXO deployment on Mac Studio network with MLX RDMA over Thunderbolt 5 mesh + Ethernet

1. Project overview

This project delivered a working EXO cluster across four Mac Studio machines, using MLX RDMA for high-bandwidth/low-latency transfers and a full mesh of USB-C Thunderbolt 5 cables (each node connected to every other node). The goal was to validate that EXO can run multi-node inference reliably and to benchmark functional behavior across multiple large models and execution modes (Pipeline vs Tensor) while keeping the network topology deterministic and debuggable.

Key outcomes:

- Stable 4-node EXO runtime achieved.
- Final design uses **Ethernet for control-plane connectivity** (discovery, RPC, health checks) and **Thunderbolt 5 mesh for the RDMA data-plane** (fast peer-to-peer transfers).
- Successfully tested inference on:
  - **gpt-oss-120b-MXFP4-Q8**
  - **gpt-oss-20b-MXFP4-Q8**
  - **Llama 3.1 8B (4-bit)**
- Validated both modes where supported:
  - **Pipeline + MLX RDMA** (most stable)
  - **Tensor + MLX RDMA** (model- and backend-dependent; intermittent issues remain due to current EXO/transport bugs)

2. Target architecture and topology

2.1 Hardware layout

- 4× Mac Studio hosts (Node A, B, C, D)
- Thunderbolt 5 USB-C cables connected in **full mesh**:
  - A↔B, A↔C, A↔D
  - B↔C, B↔D
  - C↔D
- 1× Ethernet switch with one Ethernet link per node

2.2 Network planes (final design)

Control plane (Ethernet):

- Purpose: peer discovery, membership, orchestration, EXO Models download
- Benefits: stable IP addressing, predictable routing, possibility to download AI Models

Data plane (Thunderbolt 5 mesh with MLX RDMA):

- Purpose: bulk tensor/activation transfers between peers via EXO's MLX RDMA backend
- Benefits: high bandwidth, low latency, consistent peer-to-peer performance once links are correctly enumerated

This split-plane design was the turning point: it eliminated the "works only sometimes" behavior observed when attempting to run everything on Thunderbolt-only networking.

3. Installation and configuration approach

3.1 Base system preparation

On each Mac Studio:

- Confirmed consistent OS configuration (same major version, compatible security policies).

- Verified developer tooling available (shell environment, required runtime dependencies, and EXO prerequisites).
- Ensured hostnames were unique and stable (important for logs and sanity checks).

3.2 EXO installation

EXO was installed identically on all four nodes to avoid drift:

- Same EXO build/version across nodes
- Same model cache directory layout
- Same permissions and execution method (service wrapper or user session, depending on local policy)

3.3 RDMA / MLX transport enablement

- Enabled EXO's **MLX RDMA** transport and validated that EXO recognized RDMA-capable paths between peers.
- Verified that the Thunderbolt mesh links were enumerated consistently (interfaces up, links stable).

3.4 Addressing and routing model

Ethernet:

- Static or DHCP-reserved IPs (recommended: static/reserved to keep cluster membership stable)
- One flat subnet (e.g., 10.10.10.0/24) for predictable peer reachability

Thunderbolt mesh:

- Ensured each Thunderbolt link came up deterministically and that interfaces did not remain stuck on link-local-only addressing for EXO's needs.
- Where necessary, assigned explicit IPs per Thunderbolt interface/link to prevent ambiguous routing decisions, but ended up running on Self-assigned IPs

4. Issues encountered and how they were resolved

4.1 Thunderbolt-only connectivity: peer discovery failures

Symptom:

When nodes were connected only via Thunderbolt, EXO intermittently failed to form a stable cluster. Some peers were reachable by basic connectivity checks, but EXO would not reliably discover or maintain connections.

Observed behavior patterns:

- Interfaces coming up with **link-local IPv4 (169.254.x.x)** addressing, leading to inconsistent routing and peer identification.
- Discovery and connection establishment would succeed for some peers but not others, depending on which link came up first.
- Control traffic (RPC/discovery) competing with data-plane transfers on the same Thunderbolt path made failures harder to reproduce and debug.

Resolution:

Introduced a dedicated Ethernet control plane. With Ethernet handling discovery and coordination, EXO membership stabilized immediately. Thunderbolt was then used primarily for the RDMA data-plane.

4.2 Thunderbolt Bridge experimentation: improved reachability, inconsistent stability

Symptom:

Using a Thunderbolt Bridge improved basic IP reachability and simplified the interface list, but EXO behavior remained inconsistent under load. Some runs were successful; others failed during cluster formation or during heavier transfers.

Likely root causes:

- Bridge behavior can mask per-link characteristics that RDMA/transport layers rely on.
- Subtle changes in interface naming, routing preference, or address assignment can break assumptions inside cluster discovery and peer mapping.

Resolution:

Abandoned the bridge and returned to a deterministic model:

- Ethernet for discovery/control
- Thunderbolt mesh for RDMA transfers

4.3 Tensor + MLX RDMA instability: model/backend-dependent bugs

Symptom:

Tensor mode sometimes triggered transport-level or execution-path issues when RDMA was enabled, depending on model and sharding pattern.

Resolution/workaround:

- Standardized on **Pipeline + MLX RDMA** for consistent results.
- Tested Tensor mode only for the models where it appeared to work, documenting failures as known limitations rather than blocking the cluster rollout.

5. Validation and testing

5.1 Cluster readiness checklist

Before model testing, each run required:

- All four nodes visible in EXO membership view and RDMA paths detected and mapped correctly

5.2 Model test matrix and results (functional)

Model	Quantization	Pipeline + MLX RDMA	Tensor + MLX RDMA	Notes
gpt-oss-120b-MXFP4-Q8	Q8	Stable	intermittent	Tensor path shows backend sensitivity / known bugs
gpt-oss-20b-MXFP4-Q8	Q8	Stable	Mixed	Worked on some runs; failures reproducible under specific sharding
Llama 3.1 8B (4-bit)	4-bit	Stable	Stable	Small enough to be robust; good sanity baseline

5.3 Functional QA (prompt-based)

Across models, we verified:

- General Q&A correctness (multi-turn prompts)
- Consistency under repeated runs
- No obvious hallucination spikes attributable to distributed execution (qualitative)
- Stability under concurrent prompts (where feasible)