

# Ethan Thornberg

Aliso Viejo, CA · (909) 936-5891 · [ethornbe@uci.edu](mailto:ethornbe@uci.edu) · [linkedin.com/in/ethan-thornberg](https://www.linkedin.com/in/ethan-thornberg) · [ethanthornberg.dev](https://ethanthornberg.dev) · [github.com/EthanT89](https://github.com/EthanT89)

Systems engineer who ships transport-layer networking in production, with full-stack range to back it up. Writes C and C++ close to the socket: custom UDP protocols, QUIC, network state machines, and the platform abstractions that keep sessions alive when the network underneath them changes.

## EXPERIENCE

**Roblox** — Software Engineer Intern, Engine, Systems & Network Transport

San Mateo, CA · 06/2026 – 09/2026

- Shipped Connectivity, an engine-wide connection-migration subsystem, to production on iOS, Android, Windows, macOS, and Linux for 123M daily active users, ending game drops on network changes that previously forced restarts.
- Built per-platform network monitors and a path-change state machine (debouncing, deduplication, generation counters) rejecting ~92% of raw signals as noise, plus a captive-portal-aware probe confirming reachability in ~180 ms.
- Migrated HTTP/3 in place via RFC 9000 and added spare-token re-identification to RbxTransport (UDP); de-risked rollout via a 15% iOS observation phase, adopted beyond charter by telemetry and voice chat, and selected for RDC 2026.

**First American** — Full Stack Software Engineering Intern

Santa Ana, CA · 06/2025 – 06/2026

- Built full-stack features for the DMS2 Admin Workspace: React/TypeScript frontend, C#/.NET APIs with Entity Framework and SQL Server, deployed via GitHub Actions and Helm to Kubernetes.
- Retained part-time through a 12-month term; led GitHub Copilot adoption across 100+ engineers, running mentorship sessions and reporting measured productivity gains to engineering leadership.

## PROJECTS

**Real-Time Networked Systems** — 180-Day Deep Work Journey

C · 12/2025 – Present

- Built a UDP multiplayer game server in C on Berkeley sockets: a custom binary protocol (fixed-size headers, sequence numbers, ack bitfields), a 60 Hz authoritative tick loop, and delta-compressed state broadcast cutting per-client bandwidth ~70%, tested with 12+ concurrent clients.
- Wrote a network simulation proxy (configurable 0-1000ms latency, 0-100% loss, reordering) that exposed architectural limits in naive state sync and drove a rewrite toward client prediction and lag compensation.
- Added spatial-grid collision detection, cutting broad-phase cost ~85%, and A\* bot pathfinding to keep per-client update cost bounded as entity count grew.

**Software Renderer**

C · 2026

- Built a rasterizer from scratch with no graphics API: full model-view-projection pipeline, clipping, Phong lighting, perspective-correct UV interpolation, and shadow mapping, rendering at 60 FPS at 1080p.

**Distributed Task Queue**

C · 2026

- Wrote a multi-worker task queue in C with a custom wire protocol, worker heartbeating, and at-least-once failure requeue, scaling throughput ~7x across 8 workers to drive a parallel file-processing pipeline.

## SKILLS

**Systems & Networking** — Berkeley sockets, UDP/TCP, QUIC/HTTP3, custom binary protocols, connection migration, packet capture and traffic shaping, concurrency, memory management, performance profiling

**Infrastructure** — Linux, Docker, Kubernetes, Helm, GitHub Actions, AWS

**Languages** — C, C++, C#, Python, JavaScript/TypeScript, SQL

**Databases** — PostgreSQL, Redis, SQL Server, MongoDB, Elasticsearch

## EDUCATION

**University of California, Irvine**

Expected Graduation: Winter 2028

B.S. Computer Science and Engineering — Regents Scholar · GPA 3.95

**Relevant Coursework:** Operating Systems, Computer Architecture, Design & Analysis of Algorithms, Data Structures & Algorithms, Digital Systems Design, Programming in C++, Discrete Math, Linear Algebra