



P2P DYNAMIC PORT SWITCHING

A Moving Target Defense Approach for Peer-to-Peer Networks
Team: Eren Abdullah Beşik | Şahin Erşan | Berfin Duru Alkan
Advisor: Dr. Atila Bostan



Çankaya University, Department of Software Engineering

About the Project

This Python-based system applies Moving Target Defense (MTD) to P2P communication. Two peers periodically hop their communication ports using time-step synchronization and dynamically switch between TCP and UDP based on real-time network conditions.

The bootstrap process uses RSA-2048 + AES-256-GCM encryption. After key exchange, the discovery port is permanently disabled. An embedded Live Web UI (pure HTML/CSS/JS) provides real-time monitoring of port hops and network metrics.

Core Components

Port Manager: $\text{SHA-256}(\text{params} + \text{step} + \text{peer_id})$ — deterministic, collision-safe port derivation.

Timing Sync: $\text{step} = \text{floor}((\text{now} - \text{epoch}) / \text{interval})$ via monotonic clock — NTP-independent sync.

Protocol Decision: AUTO mode: HIGH packet loss $\rightarrow$ TCP (reliability); LOW latency $\rightarrow$ UDP (speed).

Web UI: Pure HTML/CSS/JS dashboard — live port timeline, network metrics, integrated chat.

System Architecture & Protocol Flow

Phase 1 — Secure Bootstrap: RSA key exchange on discovery port 55000. Session parameters (seed, interval, port range) transmitted via AES-256-GCM. Discovery port permanently closed after handshake.

Phase 2 — SHA-256 Port Hopping: Both peers independently compute the same ports per step:
 $\text{port} = \min + (\text{SHA-256}(\text{params} + \text{step} + \text{peer_id}) \bmod \text{range})$.
Each peer calculates both its own port and the remote peer's port — no extra messaging needed.

Phase 3 — Emergency Jump: If no DATA/ACK/HEARTBEAT is received for 3 seconds, both peers compute an emergency backup port: $\text{SHA-256}(\text{params} + \text{step} + \text{peer_id} + \text{"emergency"})$. Connection is automatically restored without any coordination overhead.

Security Layers: RSA-2048 bootstrap | UUID session seed | Monotonic timestamps (anti-replay) | Port scanning results auto-expire on each hop.

Results & Future Work

Achievements: Deterministic port hopping with zero coordination overhead. Recovery from active port blocking in under 4 seconds. Real-time metric-driven TCP/UDP switching. Fully functional live monitoring dashboard.

Future Work: Multi-peer mesh network (3+ nodes). TLS per-message encryption. Mobile client. WAN testing. Machine learning for adaptive interval optimization.

Problem & Motivation

Problem: Traditional P2P systems use fixed port numbers. Attackers can scan, identify, and block these ports — making all traffic vulnerable.

Inspiration: Military frequency hopping — the radio rapidly switches frequencies to prevent detection and jamming. We apply the same principle to P2P port numbers.

Solution: Every N seconds, ports are recomputed via SHA-256. Even if an attacker identifies the current port, it expires instantly at the next hop interval.

